**Version Summary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.1 | Feb 2018 | G Mulligan | Initial document – Copied from UCSD migration document (DB2) and updated here for UCSB |
|  |  |  | **Values in Step 7K need to be double-checked** |
| 1.2 | Feb 7, 2018 | LScott | Add step 7L - ‘Over The Cap’ and ‘Salary Cap’ to ERSEarnings table |
| 1.3 | Feb 7, 2018 | GMulliga | Executed Step 2 commands through 'DROP INDEX ERS\_AER\_UNQ' |
| 1.4 | Feb 8, 2018 | GMulliga | Added the following line to Step 7L:**update ersearnings set cap\_rate=0** Finished Steps 2 and 3. Next: resume at Step 4.  |
| 1.5 | Feb 9, 2018 | GMulliga | Previously Page 9 had dropped constraint ERS\_ORGDEPT\_FK1 (on ORG\_CD). But Path isn’t changing ORG\_CD so this constraint need not be dropped. Removed Page 9.However, since Path **is** changing DEPT\_ID, there is a separate statement later to drop DEPT\_ID constraint (ERS\_ORGDEPT\_FK2). Executed commands up to and including the following:ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERSORGDEPTPK |
| 1.6 | Feb 12, 2018 | GMulliga | Misc changes |
| 1.7 | Feb 12, 2018 | GMulliga | Added Step 7M, 7N. Also added code to expand the Employee Id in the ERSPOSSIBLEER table and add back the constraint in ERSPOSSIBLEER |
| 1.8 | Feb 15, 2018 | GMulliga | Ran the steps through 7E. |
| 1.9 | Feb 20, 2018 | GMulliga | Skipped Step 7F and continued. |
| 1.10 | Feb 21, 2018 | GMulliga | Ran steps through Step 8. Skipped Step 9. Resumption point is Step 10. |
| 1.11 | Mar 20, 2018 | GMulliga | Completed running most steps. Some steps were skipped after consultation with Larry. These may be revisited.  |
| 1.12 | April 5, 2018 | GMulliga | Streamlined to remove comments, screen shots, etc., so that it can be distributed to others within the PPS / ERS group.  |
| 1.13 | June 12, 2018 | GMulliga | Removed redundant insertions of 2/O and 2/R to the ERSPAYCAT table. There is also discussion that perhaps 2/O and 2/r only apply to Legacy and therefore should not be inserted at all here, but for now, the script will insert a single unique set of 2/O and 2/R.  |

|  |  |  |  |
| --- | --- | --- | --- |
| 1.14 | July 3, 2018 | GMulliga | Step 7I. Removed the insertion of two rows to the ERSPAYCAT table (per Kerry, these P(Par file) Legacy entries are not applicable to PPS since PPS's Pay Categories are all alpha, not numeric).2 R 2018-05-16 14:40:11 sys.admin 2018-05-16 14:40:11 sys.admin P2 O 2018-05-16 14:40:15 sys.admin 2018-05-16 14:40:15 sys.admin P |
| 1.15  | July 24, 2018 | GMulliga | Added more off-quarter ERSDOSCODEs: 9AS, 9CS, 9DC, AAC, ACS, ACT, ADC, APA, ASN, SSC |
| 1.16 | August 1, 2018 | GMulliga | Step 7I. Background: v1.14 removed the 2/R/P and 2/O/P rows from ERSPAYCAT, since DOS\_PAY\_CAT = '2' is UCPath, but INTERFACE = 'P' stands for 'PPS PAR File'. Therefore '2' and 'P' were incompatible. But the DOS\_PAY\_CAT = '2' rows are still needed for UCPath, except they should have INTERFACE = 'L', not 'P' (INTERFACE = 'L' stands for 'Labor Ledger' from UCPath). So v1.16 will add 2/R/L and 2/O/L.To do that, v1.16 first inserts 2/O and 2/R rows in Step 7I (without any INTERFACE value yet).Then in Step 7K, v1.16 also later changes how ERSPAYCAT.INTERFACE is initialized:  Update UCPath's DOS\_PAY\_CAT = '1' and '2' rows to have Interface='L' (Labor Ledger). Update PPS's DOS\_PAY\_CAT = 'N' rows to have interface = 'P' (PPS PAR File).Update ERSPayCat set interface = 'P' where DOS\_PAY\_CAT = 'N';Update ERSPayCat set interface = 'L' where DOS\_PAY\_CAT IN ('1', '2'); |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

This document provides step-by-step instructions to upgrade a DB2 ERS 10.12 B009 database to ERS 11.0 format. Due to dependencies, this document must be executed in the order presented. See related documents for IBM DB2 databases and Oracle databases.

**Recommendation:**  Execute these database SQL commands individually and verify that each statement completes successfully before proceeding to the next statement. This is a longer, but more thorough approach. If statements are executed in blocks, it is easy to miss an error which may have a cascading (bad) effect as more SQL statements are executed.

**Step 1: UCSB Database Upgrade Prerequisite**

Before starting this database upgrade process, make sure your ERS instance is running at release level

ERS 10.12 B009 before proceeding with the following steps.



If needed, the UCOP ERS program support staff can provide a custom ERS upgrade script (eliminating the need to perform all interim releases) to get your database to ERS 10.12 B009 format.

**To be used anywhere in this script:**

The following 'reorg' command is useful and may need to be run to put tables into an alterable state:

 CALL SYSPROC.ADMIN\_CMD ('REORG TABLE ERS\_SOMETABLE');

**Step 2: Employee Update from char(9) to varchar(11)**

 ALTER TABLE ERSUSER DROP CONSTRAINT ERSUser\_FK\_EMP;

 ALTER TABLE ERSPILOTPARTICIPANT DROP CONSTRAINT ERSPilotPart\_FK\_E;

 ALTER TABLE ERSPOSSIBLEER DROP CONSTRAINT ERS\_PER\_FK\_EMP;

 ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_FK\_EMP;

 ALTER TABLE ERSLATEPAY DROP CONSTRAINT ERSLatePay\_FK;

 ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERS\_ORGDEPT\_FK2

 ALTER TABLE ERSEXCLUDEUSERTOKEN DROP CONSTRAINT ERS\_EUTOKEN\_FK\_EMP;

ALTER TABLE ERSEARNINGS DROP CONSTRAINT ERSEARN\_FK\_EMP;

 ALTER TABLE ERSCSEMPL DROP CONSTRAINT ERSCSEMPL\_FK\_EMP;

 ALTER TABLE ERSCRITERIAEMPLOYEE DROP CONSTRAINT ERS\_CE\_FK\_EMP;

 ALTER TABLE ERSEMPL DROP CONSTRAINT ERSEmpl\_PK;

 ALTER TABLE ERSEMPL ADD PPS\_EMPLOYEE\_ID CHAR(9);

 UPDATE ERSEMPL set pps\_employee\_id = employee\_id;

 ALTER TABLE ERSEMPL ALTER COLUMN PPS\_EMPLOYEE\_ID SET DATA TYPE varchar(9);

 ALTER TABLE ERSEMPL ALTER COLUMN PPS\_EMPLOYEE\_ID SET NOT NULL;

 ALTER TABLE ERSEMPL ALTER COLUMN EMP\_NAME SET DATA TYPE varchar(120);

 ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSEMPL ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

DROP INDEX ERSEARNINGS\_IDX

DROP INDEX ERSEARNING\_UNQ\_IDX

 ALTER TABLE ERSEARNINGS ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

CREATE INDEX ERSEARNINGS\_IDX

 ON ERSEARNINGS (Employee\_ID ASC, Pay\_Per\_End\_Date ASC);

 ALTER TABLE ERSCSEMPL DROP CONSTRAINT ERSCSEmpl\_PK

 ALTER TABLE ERSCSEMPL ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSCSEMPL ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

ALTER TABLE ERSCSEMPL ADD CONSTRAINT ERSCSEMPL\_PK

PRIMARY KEY (FS\_ID, EMPLOYEE\_ID, START\_DATE)

 ALTER TABLE ERSUSER ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSCRITERIAEMPLOYEE DROP CONSTRAINT ERS\_CE\_PK

 ALTER TABLE ERSCRITERIAEMPLOYEE ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSCRITERIAEMPLOYEE ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

ALTER TABLE ERSCRITERIAEMPLOYEE ADD CONSTRAINT ERS\_CE\_PK

 PRIMARY KEY (SC\_ID, Employee\_ID)

 ALTER TABLE ERSLATEPAY DROP CONSTRAINT ERSLatePay\_PK

 ALTER TABLE ERSLATEPAY ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSLATEPAY ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

 ALTER TABLE ERSLATEPAY ADD CONSTRAINT ERSLATEPAY\_PK PRIMARY KEY(Employee\_ID,Period\_ID)

 ALTER TABLE ERSPOSSIBLEER ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11) ;

 ALTER TABLE ERSPI DROP CONSTRAINT ERSPI\_PK

 ALTER TABLE ERSPI ALTER COLUMN EMP\_ID SET DATA TYPE varchar(11) ;

 ALTER TABLE ERSPI ALTER COLUMN EMP\_ID SET NOT NULL;

 ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_PK PRIMARY KEY (SP\_ID, Emp\_ID)

 ALTER TABLE ERSEXCLUDEUSERTOKEN DROP CONSTRAINT ERS\_EUToken\_PK

 ALTER TABLE ERSEXCLUDEUSERTOKEN ALTER COLUMN EMPLOYEE\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSEXCLUDEUSERTOKEN ALTER COLUMN EMPLOYEE\_ID SET NOT NULL;

ALTER TABLE ERSEXCLUDEUSERTOKEN ADD CONSTRAINT ERS\_ERSEUTOKEN\_PK

 PRIMARY KEY (Employee\_ID)

 DROP INDEX ERS\_AER\_UNQ

 ALTER TABLE ERSACTUALER ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSACTUALER ALTER COLUMN Employee\_ID SET NOT NULL;

 CREATE UNIQUE INDEX ERS\_AER\_UNQ ON ERSACTUALER( employee\_id, period\_id );

 ALTER TABLE ERSINCLUDEEMPL DROP CONSTRAINT ERSINCLUDEEMPL\_PK

 ALTER TABLE ERSINCLUDEEMPL ALTER COLUMN Employee\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSINCLUDEEMPL ALTER COLUMN Employee\_ID SET not null;

 ALTER TABLE ERSINCLUDEEMPL ADD CONSTRAINT ERSIncludeEmpl\_PK

 PRIMARY KEY (Employee\_ID)

 ALTER TABLE ERSPIEMPLTANK DROP CONSTRAINT ERSPIEmplTank\_PK

 ALTER TABLE ERSPIEMPLTANK ALTER COLUMN Emp\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSPIEMPLTANK ALTER COLUMN Emp\_ID SET not null;

ALTER TABLE ERSPIEMPLTANK ADD CONSTRAINT ERSPIEmplTank\_PK PRIMARY KEY (Project\_ID, Emp\_ID)

 ALTER TABLE ERSPILOTPARTICIPANT DROP CONSTRAINT ERSPilotPart\_PK

 ALTER TABLE ERSPILOTPARTICIPANT ALTER COLUMN emp\_id SET DATA TYPE varchar(11);

 ALTER TABLE ERSPILOTPARTICIPANT ALTER COLUMN emp\_id SET not null;

ALTER TABLE ERSPILOTPARTICIPANT ADD CONSTRAINT ERSPilotPart\_PK

 PRIMARY KEY (period\_id, Emp\_ID)

 ALTER TABLE ERSEMPL ADD CONSTRAINT ERSEmpl\_PK PRIMARY KEY(Employee\_ID);

ALTER TABLE ERSCSEMPL ADD CONSTRAINT ERSCSEmpl\_fk\_emp

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSCRITERIAEMPLOYEE ADD CONSTRAINT ERS\_CE\_FK\_EMP

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

**Step 2 (continued): Employee Update from char(9) to varchar(11)**

ALTER TABLE ERSLATEPAY ADD CONSTRAINT ERSLatePay\_FK

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSEXCLUDEUSERTOKEN ADD CONSTRAINT ERS\_EUTOKEN\_FK\_EMP

FOREIGN KEY (EMPLOYEE\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSPILOTPARTICIPANT ADD CONSTRAINT ERSPilotPart\_FK\_E

FOREIGN KEY (EMP\_ID) REFERENCES ERSEmpl (Employee\_ID);

 ALTER TABLE ERSPOSSIBLEER ADD CONSTRAINT ERS\_PER\_FK\_EMP FOREIGN KEY (EMPLOYEE\_ID)

 REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSPI ADD CONSTRAINT ERSPI\_FK\_EMP FOREIGN KEY (EMP\_ID)

REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSUSER ADD CONSTRAINT ERSUser\_fk\_emp

FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

ALTER TABLE ERSEARNINGS ADD CONSTRAINT ERSEarn\_FK\_emp

 FOREIGN KEY (Employee\_ID) REFERENCES ERSEmpl (Employee\_ID);

**Step 3: Supplemental Employee Updates**

 DROP INDEX ERSARCHIVEDREPORT\_INDEX2

 ALTER TABLE ERSARCHIVEDREPORT ALTER COLUMN EMP\_ID SET DATA TYPE varchar(11);

 ALTER TABLE ERSARCHIVEDREPORT ALTER COLUMN EMP\_ID SET NOT NULL;

 CREATE INDEX ERSARCHIVEDREPORT\_INDEX2 ON ERSARCHIVEDREPORT( emp\_id )

**Step 4: Clean Employee IDs**

 update ERSEMPL set employee\_id=trim(employee\_id)

**Step 5: Home Department Updates**

**DB2: Execute DB2 Scripts in this order (SQL Server script below is more thorough)**

 ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;

 ALTER TABLE ERSHOMEDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

 ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);

 ALTER TABLE ERSEMPL ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

 ALTER TABLE ERSORGDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

**Execute DB2 Scripts in this order**

 ALTER TABLE ERSCRITERIAHOMEDEPARTMENT DROP CONSTRAINT ERS\_CHD\_PK

 ALTER TABLE ERSORGDEPT DROP CONSTRAINT ERSORGDEPTPK

 ALTER TABLE ERSEMPL DROP CONSTRAINT ERSEmpl\_FK

 ALTER TABLE ERSHOMEDEPT DROP CONSTRAINT ERSHomeDept\_PK;

 ALTER TABLE ERSHOMEDEPT ADD CONSTRAINT ERSHomeDept\_PK PRIMARY KEY(DEPT\_ID);

ALTER TABLE ERSEMPL ADD CONSTRAINT ERSEmpl\_FK

 FOREIGN KEY (Dept\_ID) REFERENCES ERSHomeDept (Dept\_ID)

 ALTER TABLE ERSORGDEPT ADD CONSTRAINT ERSORGDEPTPK PRIMARY KEY(Org\_CD, Dept\_ID)

 ALTER TABLE ERSSCHDEPT DROP CONSTRAINT ERSSchDept\_PK

 ALTER TABLE ERSSCHDEPT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10);

 ALTER TABLE ERSSCHDEPT ALTER COLUMN DEPT\_ID SET NOT NULL;

 ALTER TABLE ERSSCHDEPT ADD CONSTRAINT ERSSchDept\_PK

 PRIMARY KEY (Sch\_CD, Dept\_ID)

ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET DATA TYPE varchar(10)

 ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ALTER COLUMN DEPT\_ID SET NOT NULL;

 ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ADD CONSTRAINT ERS\_CHD\_PK

 PRIMARY KEY (SC\_ID, DEPT\_ID)

 ALTER TABLE ERSCRITERIAHOMEDEPARTMENT ADD CONSTRAINT

 ERS\_CHD\_FK\_HD FOREIGN KEY (DEPT\_ID)

 REFERENCES ERSHOMEDEPT (DEPT\_ID)

**Step 5 (continued): Home Department Updates**

 UPDATE ERSHOMEDEPT set DEPT\_ID=TRIM(DEPT\_ID)

**Step 6: Add new Labor Ledger columns to ERSEARNINGS table**.

 alter table ERSEARNINGS add Run\_Id varchar(10)

 alter table ERSEARNINGS add Empl\_Rcd integer

 alter table ERSEARNINGS add Eff\_Date TIMESTAMP

 alter table ERSEARNINGS add Eff\_Seq integer

 alter table ERSEARNINGS add Off\_Cycle varchar(1)

 alter table ERSEARNINGS add Run\_Id\_Earn varchar(10)

 alter table ERSEARNINGS add Addl\_Seq varchar(10)

 alter table ERSEARNINGS add Journal\_Id varchar(10)

 alter table ERSEARNINGS add Journal\_Line varchar(10)

 alter table ERSEARNINGS add Journal\_Line\_Ref varchar(10)

 alter table ERSEARNINGS add Business\_Unit varchar(5)

 alter table ERSEARNINGS add Pay\_Cat varchar(2)

 alter table ERSEARNINGS add Time\_Code varchar(2)

 alter table ERSEARNINGS add Restatement varchar(1)

 alter table ERSEARNINGS add Interface varchar(1)

 Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_TIME\_CODE SET DATA TYPE varchar(2);

 Alter TABLE ERSEARNINGS ALTER COLUMN DOS\_PAY\_CATEGORY SET DATA TYPE varchar(2);

 Alter TABLE ERSEARNINGS ALTER COLUMN TITLE\_CODE SET DATA TYPE varchar(6)

**Step 6B: Set database version to 11.0**

 update ERSVERSION set version\_id='11.0'

**Step 7: ERSCONSTANTS update**... + ERSEarnings (set default INTERFACE to 'P' PAR)

 **-- Initialize restatement and interface columns**

 **-- perform the following update one year at a time (otherwise transaction log will fill and query will fail)**

 **Run the following update earnings commands**

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2010

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2011

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2012

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2013

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2014

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2015

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2016

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) = 2017

 update ERSEARNINGS set RESTATEMENT='N', INTERFACE='P' where year(PAY\_CYCLE\_END\_DATE) > 2017

**Step 7B: Update ERS DOS Codes for Off Quarter processing**

 insert into ERSDOSCODE values ('9AC','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('9TC','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('9CC','O', current timestamp,'\*INSTALL\*',null,null)

also add these, per KH and FI's emails on 7/24/18

 insert into ERSDOSCODE values ('9AS','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('9CS','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('9DC','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('AAC','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('ACS','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('ACT','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('ADC','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('APA','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('ASN','O', current timestamp,'\*INSTALL\*',null,null)

 insert into ERSDOSCODE values ('SSC','O', current timestamp,'\*INSTALL\*',null,null)

**Step 7C: Update ERS DOS Codes for Off Quarter processing**

 insert into ERSPAYCAT values('1','R', current timestamp,'\*INSTALL\*',null,null) -- regular reporting period

 insert into ERSPAYCAT values('1','O', current timestamp,'\*INSTALL\*',null,null) -- off quarter reporting period

**Step 7D: Create Unique PAR Earnings and I-129 Earnings unique indexes for duplicate earning check**

**I-129**

CREATE INDEX ERS\_EARNINGS\_I129\_UNIQUE ON ERSEARNINGS (Interface, Empl\_Rcd, Eff\_date, Eff\_Seq, Run\_Id, Run\_Id\_Earn, Business\_Unit, Off\_Cycle, Dist\_Dos, Journal\_Id, Journal\_Line, Journal\_Line\_Ref, Addl\_Seq)

**PAR**

CREATE INDEX ERS\_EARNINGS\_PAR\_UNIQUE ON ERSEARNINGS(Interface, Pay\_Cycle\_End\_Date, Pay\_Cycle\_Code, Record\_Type, Pri\_Gross\_Ctl, Entry\_Seq\_No)

**Step 7E: Expand ERSEarnings Paid\_Amount, Paid\_Hours, and Pay\_Rate columns**

 ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Hours SET DATA TYPE decimal(6,2)

 ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Hours SET NOT NULL;

 ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Amount SET DATA TYPE decimal(12,2);

 ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Amount SET NOT NULL;

 ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET DATA TYPE decimal(14,4);

 ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET NOT NULL;

**Step 7F: Create six digit UCPATH Department ID’s in ERS Scheduler.**

Skip for UCSB. See Larry's 2/15/18 1:57pm email. UCSB is different from UCLA. UCLA's scheme is more complicated. UCSB doesn’t use ERSSchDeptBU.

**Step 7G: Add Non-Academic Appointment Indicators (3,B,C)**

insert into ERSPERSPGMCODE values('1','3', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

insert into ERSPERSPGMCODE values('1','B', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

insert into ERSPERSPGMCODE values('1','C', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

**Step 7H: Update Journal\_Line\_Ref for I-129 records (set to “PAYROLL”)**

Not run for UCSB. There aren’t any L rows because L means Labor Ledger. Since I-129 hasn’t ever been run there are no L records. This step might still be needed in the future, if I-129 has been run.

-- ~~update ERSEarnings set Journal\_Line\_Ref ='PAYROLL' where interface='L'~~

**Step 7I: 7/3/18 - deleted these two erspaycat INSERTS of 2/R/P and 2/O/P. Per Kerry Hester at UCOP, they are not needed.**

**See the Version Summary comments for v1.14 and v.1.16.**

-- insert into ERSPayCat values('2','R',getdate(), 'sys.admin',getdate(),'sys.admin')

 insert into ERSPAYCAT values('2','R', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

-- insert into ERSPayCat values('2','O',getdate(), 'sys.admin',getdate(),'sys.admin')

 insert into ERSPAYCAT values('2','O', CURRENT TIMESTAMP, 'sys.admin', CURRENT TIMESTAMP,'sys.admin')

**Step 7J: Add ‘Over The Cap’ to ERSEarnings table.**

alter table ERSEARNINGS add Over\_The\_Cap varchar(10)

**Step 7K: ERSDOSCODE table and ERSPAYCAT table are now indexed using ‘Interface’**

**(same codes in PPS and I-129 are used, but identify ‘Effort’ differently)**

 alter table **ERSTIMECODE** add Interface varchar(1)

 alter table **ERSPAYCAT** add Interface varchar(1)

 Update ERSTimeCode set interface = 'P'

ALTER TABLE ERSTimeCode DROP CONSTRAINT ERSTimeCode\_PK

ALTER TABLE ERSTIMECODE ALTER COLUMN Interface Varchar(1) NOT NULL

ALTER TABLE ERSTIMECODE ALTER COLUMN Interface SET NOT NULL

ALTER TABLE ERSTIMECODE ADD CONSTRAINT ERSTimeCode\_PK

 PRIMARY KEY (DOS\_Time\_Code, Period\_Type, Interface)

insert into ERSTIMECODE values ('1','O', CURRENT TIMESTAMP, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('2','O', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('5','O', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('1','R', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('2','R', current timestamp, '\*UPGRADE\*',null,null,'L')

insert into ERSTimeCode values ('5','R', current timestamp,'\*UPGRADE\*',null,null,'L')

**See the Version Summary comments for v1.14 and v.1.16.**

 Update ERSPayCat set interface = 'P' where DOS\_PAY\_CAT = 'N';

 Update ERSPayCat set interface = 'L' where DOS\_PAY\_CAT IN ('1', '2');

 ALTER TABLE ERSPAYCAT DROP CONSTRAINT ERSPayCat\_PK

 ALTER TABLE ERSPAYCAT ALTER COLUMN Interface SET DATA TYPE Varchar(1)

 ALTER TABLE ERSPAYCAT ALTER COLUMN Interface SET NOT NULL

ALTER TABLE ERSPAYCAT ADD CONSTRAINT ERSPayCat\_PK

 PRIMARY KEY (DOS\_Pay\_Cat, Period\_Type, Interface)

 Update ERSPayCat set interface = 'L' where DOS\_Pay\_Cat='1'

**Step 7L: Add ‘Over The Cap’ and ‘ Salary Cap’**

alter table ERSEARNINGS add Cap\_Rate Decimal (14,4)

update ersearnings set cap\_rate=0

**Step 7M - expand the employee ids in table ERSPIEMPL**

ALTER TABLE ERSPIEMPL ALTER COLUMN PI\_EMP\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSPIEMPL ALTER COLUMN CO\_PI\_EMP\_ID SET DATA TYPE varchar(11);

**Step 7N - expand the employee ids in table ERSSPONSOREDPROJECT**

ALTER TABLE ERSSPONSOREDPROJECT ALTER COLUMN PI\_EMP\_ID SET DATA TYPE varchar(11);

ALTER TABLE ERSSPONSOREDPROJECT ALTER COLUMN CO\_PI\_EMP\_ID SET DATA TYPE varchar(11);

**Step 8: Set system generate key values for ERSEarnings table inserts by PAR & I-129 loads**

 select max(Import\_Seq\_Nbr) from ERSEarnings

 ALTER TABLE ERSEarnings ALTER COLUMN Import\_Seq\_Nbr RESTART WITH 3100000

 **Step 9: Set ERS sys.admin password allowing user to login to ERS**

Skip for UCSB.

?Revisit later?

 (DO THIS FROM ERS LOGIN SCREEN – S.B. PERFORMED BY .WAR FILE INSTALLER)

**all 10.x for now**

**Step 10: Archive Migration from DTO format to PDF format**

This code is contained only in ERS 10.12 B010

The archive code in ERS 10.12 B010 now builds archives in PDF format, not DTO format

All archive code in releases prior to ERS 10.12 B010 build archives in DTO (data) format

The ERS 10.12 B010 batch program to be run is **InterfaceArchiveReportMigation10to11.java**

**Overview**

ERS 10.X created archives in DTO format. ERS 11.0 creates archives in PDF format

This step changes the ERS 10.x archives from a **DTO** (data transfer object) to a **PDF** format

(necessary to solve the issue of archiving from one version of ERS and restoring to another)

The ERS Archive tables are **ERSARCHIVEDREPORT** and **ERSARCHIVEDFS**.

All archive data is contained in ERSARCHIVEDREPORT and ERSARCHIVEDFS

**\*\*\* IMPORTANT\*\*\***

**Make sure you run this process in a QA environment and verify the archives in a QA environment.**

**It would be a good precaution, to make a long term backup of the production database.**

**Process:**

These are the steps necessary to convert ERS DTO archives to PDF archives:

**10.1:** Check for data in database table **ERSARCHIVEDREPORT.**

If this table is empty, skip this step. No archive migration is necessary.

**10.2:** Check the year of the archives

<2018 is DTO format. 2018+ is PDF format.

This will tell you if the archives are in a DTO format or PDF format.

If DTO format, migration is necessary. If PDF format, no migration is necessary.

Run this SQL: **select distinct year(archived\_date) from ERSARCHIVEDREPORT**

If you only see year 2018 (or greater), your archives are in PDF format and skip the migration.

If you see year less than 2018, the migration required.

**10.3:** Preparing for migration

Migration is performed in a non-production environment to minimize ERS production downtime.

Archive migration may run for days.

Load the ERS 10.12 B010 war file to your QA environment

Copy your production ERS database to your QA environment

Verify your QA ERS database connection and ERS code (ERS signon screen display version 10.12 B010)

**10.4:** Run the migration program

In you QA environment, run ERS batch program InterfaceArchiveReportMigation10to11.java

It is necessary to set your ERS config directory in the classpath (shown in screenshot)

It also is necessary to pass two parameters to the archive migration process.

1: The path to the ERS Jasper report objects (red underline) . Explode you ers.war file to a temp directory

2: The path to your ExternalizedString.xml file (blue underline)



**10.5: Replace production DTO archives with QA PDF Archives**

Tables **ERSARCHIVEDREPORT**and **ERSARCHIVEDFS** in QA environment **REPLACE** the same tables in production ERS environment. Copy/Replace these tables from QA to PROD

**Archive migration program execution comments**

The migration program can easily consume all memory on your server causing the process to crash.

To prevent ‘out-of-memory errors’, the migration program processes 100,000 archived effort reports at a time and then stops to releases memory. The migration process commits updates in 5000 archive increments.

As the archive migration run, the DTO pdf is read from table ERSAarchivedReport and converted to PDF (using the JasperReport objects and the campuses customizations in ErternalizedStrings.xml). Once the PDF is built, the DTO in ERSArchivedReports is overlaid with the PDF . The same process is used for the archive payroll report. The process continues until all ERSArchivedReport rows are processed or 100,000 rows are processed.

Larger campuses may have millions of rows in the ERSArchivedReport table, making it necessary to run the migration process 10+ times. Each iteration of the migration process will likely take a few days.

Run this SQL:

**select year(archived\_date) , count(\*) from ERSARCHIVEDREPORT group by year(archived\_date)**

If you see any archive dates < 2018, run the process again

Once all archive dates < 2018 are cleared, the ERS Archive Migration is complete.

**Performance:**

When this process was run for UCSB in QA (Jan 2018), it took 15 hours to migrate 100,000 archived effort reports.

**Step 11:** **Closing comments**

Please be aware that the ERS WOS (without salary) interface should now contain UCPATH employee ids and not PPS employee ids

Also, It is highly recommended that you long term archive a copy of the ERS 10.12 production database

If any issues arise after the UCPATH implementation, it may be necessary to load the ERS 10.12 system in a QA environment to retrieve or repair ERS data.

**Done with ERS 11.0 Upgrade !**

**Appendix 1**

Programming Note 1:

Any changes to the ERSEarnings table need to be replicated in ERS 11.0 Java class

DrillDownQuery.java, which is used by DrillDownBuilder.java to construct the payroll report.

See ERSMappings folder, files payrollReport2.png and payrollReport3.png

Programming Note 2:

**The following methods should be part of BaseEmployee.java but lost when employee.hbm changed.**

Methods were lost when employee.hmb is compiled by hibernate synchronizer

Methods should not be part off employee.hbm

I manually added it back to BaseEmployee.java.

It is called as part of a JUnit test

public void addToUsers (edu.ucop.ers.domain.User user) {

 if (null == getUsers()) setUsers(new java.util.HashSet<edu.ucop.ers.domain.User>());

 getUsers().add(user);

 }

public void addToEarnings (edu.ucop.ers.domain.Earning earning) {

 if (null == getEarnings()) setEarnings(new java.util.HashSet<edu.ucop.ers.domain.Earning>());

 getEarnings().add(earning);

 }

public void addToReportHeaders (edu.ucop.ers.domain.ReportHeader reportHeader) {

 if (null == getReportHeaders()) setReportHeaders(

new java.util.HashSet<edu.ucop.ers.domain.ReportHeader>());

 getReportHeaders().add(reportHeader);

 }

public void addToCostSharingCommitments (edu.ucop.ers.domain.CostSharingCommitment

 costSharingCommitment) {

 if (null == getCostSharingCommitments()) setCostSharingCommitments(new

 java.util.HashSet<edu.ucop.ers.domain.CostSharingCommitment>());

 getCostSharingCommitments().add(costSharingCommitment);

 }

public void addToPiProjects (edu.ucop.ers.domain.PrincipalInvestigator principalInvestigator) {

 if (null == getPiProjects()) setPiProjects(new

 java.util.HashSet<edu.ucop.ers.domain.PrincipalInvestigator>());

 getPiProjects().add(principalInvestigator);

 }

Programming Note 3: LaborLedger Date Format

 Labor Ledger record date format sometimes changes requiring format change to

 ERSCONSTANTS.java - public static final String DATE\_FORMAT\_PAR = "MMddyyyy";

Programming Note 4: EarningMatcher.java - method MatchStatus

 Checks PayCategory, DOS Code, amd Time Code to determine if an earning should be certified.

 This check should be skipped for ERS 11.0

 Notice that column 'interface' (P=PAR, L=LaborLedger) has been added to ERSEarnings table.

 and code to populate in InterfaceLaborLedgerProcessor, and InterfacePARProcessor

 \*\* method MatchStatus determines if a LaborLedger earning should be certified only if derived% >0. This may not be sufficient.

Configuation Note 1: ERSConfig.xml - used in ConfigLoader.java

 // which payroll system being used (PPS or PeopleSoft). Default to PeopleSoft

 String payrollSystemPps = (String) cfg.getProperty(

 ERSConstants.***CFG\_OPTIONS\_PAYROLL\_SYSTEM\_PPS***);

 **if** ( payrollSystemPps != **null** ) {

 ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

**new** Boolean( bypass11Over12Check ) );

 } **else** {

 ctx.setAttribute( ERSConstants.***PAYROLL\_SYSTEM\_PPS***,

ERSConstants.***FALSE*** );

 }

 see ERSConfig.xml

 <bypass-11over12-appointment-check>true</bypass-11over12-appointment-check>

JUNIT Note: (JUnits run correctly in EffortReporting R10\_11\_B001, but not EffortReporting R11\_Support)

 1: Latheef and Bascar's work

 RunUcpathEmployeeConversionTest.java

 UcpathEmployeeConversionProcessorTest.java

 2: See step 1: Some JUnit required methods are missing from employee.hbm

 Methods were lost when employee.hmb is compiled by hibernate synchronizer

 Methods should not be part off employee.hbm

3: JUnit TestFundingSrcUniquenessTest is looking for file Z:/interfaces/Test

Files/UCD/fin\_systems\_interface.31-MAR-2005

 File can not be found so JUnit is commented out.

 4: JUnit class UcpathEmployeeConversionProcessorTest method testAddEmployee does not run....

has bugs, commented out

 5: JUnit RunUcpathEmployeeConversionTest fails - commented out

 6: JUnit SearchDeadlockFinder pukes all-over itself

 see class: AbstractSavableSearchImpl method:addToCachedSelectedItemsSet( List moreKeys ) {

 error is: java.lang.ClassCastException: java.lang.NoSuchMethodException: Unknown property 'dummy'

 The following methods have been commented out

 - testSaveAndDelete()

 - testForDeadlock()

 - testSaveAndDeleteAdHoc()

 7: JUint class SearchManagerImpl method testCopySearch - added try / catch block to force success

 8; JUnit: InterfacePARProcessorTest

 Notice junit test class "extends HibernateUnitTester" This sets up the database

 \* insert\_base\_test\_fixture\_data.sql

 \* insert\_standard\_test\_fixture\_data.sql

 9: JUnit: InterfaceLaborLedgerTest - needs to be developed

**Appendix 2:**

These steps were run at UCLA to remove Effort Reports and Earnings prior to 2014.

This shunk the size of the UCLA ERS earnings table from 35M earnings to 12M earnings.

Database size was reduced approximately by half.

This was needed for UCLA database to ‘fit’ on UCOP ERS DEV DB server

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

 and not import\_seq\_nbr in

 (select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  1

 - A cursor with the name 'C1' does not exist. [SQLSTATE 34000] (Error 16916) – This is OK, do not stop.

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  2

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  3

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  4

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  5

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  6

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  7

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  8

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  9

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  10

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  11

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  12

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  13

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  14

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  15

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  16

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  17

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  18

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  19

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  20

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  21

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  22

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  23

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  24

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  25

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  26

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  27

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  28

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  29

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  30

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  31

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  32

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  33

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  34

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  35

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  36

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  37

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  38

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  39

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  40

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  41

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  42

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  43

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  44

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  45

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  46

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  47

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  48

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  49

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  50

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  51

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  52

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  53

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  54

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  55

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  56

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  57

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  58

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  59

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  60

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  61

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  62

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  63

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  64

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  65

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  66

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  67

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  68

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  69

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  70

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  71

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  72

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  73

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  74

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  75

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  76

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  77

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  78

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  79

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  80

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  81

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  82

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  83

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  84

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  85

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  86

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  87

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  88

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  89

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  90

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  91

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  92

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  93

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  94

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  95

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  96

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  97

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  98

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  99

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  100

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  101

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  102

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  103

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  104

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  105

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  106

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  107

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  108

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  109

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  110

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  111

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  112

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  113

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  114

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  115

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  116

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  117

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  118

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  119

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  120

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  121

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  122

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  123

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  124

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  125

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  126

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  127

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  128

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  129

exec ersdb\_2017July21\_ERS11\_IT3\_Validate.dbo.RemoveEffortReportsForSinglePeriod  130

delete from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ersearnings

where year(pay\_per\_end\_date) < 2014

 and not import\_seq\_nbr in

 (select import\_seq\_nbr from [ersdb\_2017Oct16\_ERS11\_PPT2.5\_Expanded]..ERSEarningsVersion V)

**Must do this for UCSB but wait till after system up and running**

 **all 10.x for now**

**Step 10: Archive Migration from DTO format to PDF format**

This code is contained only in ERS 10.12 B010

The archive code in ERS 10.12 B010 now builds archives in PDF format, not DTO format

All archive code in releases prior to ERS 10.12 B010 build archives in DTO (data) format

The ERS 10.12 B010 batch program to be run is **InterfaceArchiveReportMigation10to11.java**

**Overview**

ERS 10.X created archives in DTO format. ERS 11.0 creates archives in PDF format

This step changes the ERS 10.x archives from a **DTO** (data transfer object) to a **PDF** format

(necessary to solve the issue of archiving from one version of ERS and restoring to another)

The ERS Archive tables are **ERSARCHIVEDREPORT** and **ERSARCHIVEDFS**.

All archive data is contained in ERSARCHIVEDREPORT and ERSARCHIVEDFS

**\*\*\* IMPORTANT\*\*\***

**Make sure you run this process in a QA environment and verify the archives in a QA environment.**

**It would be a good precaution, to make a long term backup of the production database.**

**Process:**

These are the steps necessary to convert ERS DTO archives to PDF archives:

**10.1:** Check for data in database table **ERSARCHIVEDREPORT.**

If this table is empty, skip this step. No archive migration is necessary.

**10.2:** Check the year of the archives

<2018 is DTO format. 2018+ is PDF format.

This will tell you if the archives are in a DTO format or PDF format.

If DTO format, migration is necessary. If PDF format, no migration is necessary.

Run this SQL: **select distinct year(archived\_date) from ERSARCHIVEDREPORT**

If you only see year 2018 (or greater), your archives are in PDF format and skip the migration.

If you see year less than 2018, the migration required.

**10.3:** Preparing for migration

Migration is performed in a non-production environment to minimize ERS production downtime.

Archive migration may run for days.

Load the ERS 10.12 B010 war file to your QA environment

Copy your production ERS database to your QA environment

Verify your QA ERS database connection and ERS code (ERS signon screen display version 10.12 B010)

**10.4:** Run the migration program

In you QA environment, run ERS batch program InterfaceArchiveReportMigation10to11.java

It is necessary to set your ERS config directory in the classpath (shown in screenshot)

It also is necessary to pass two parameters to the archive migration process.

1: The path to the ERS Jasper report objects (red underline) . Explode you ers.war file to a temp directory

2: The path to your ExternalizedString.xml file (blue underline)



**10.5: Replace production DTO archives with QA PDF Archives**

Tables **ERSARCHIVEDREPORT**and **ERSARCHIVEDFS** in QA environment **REPLACE** the same tables in production ERS environment. Copy/Replace these tables from QA to PROD

**Archive migration program execution comments**

The migration program can easily consume all memory on your server causing the process to crash.

To prevent ‘out-of-memory errors’, the migration program processes 100,000 archived effort reports at a time and then stops to releases memory. The migration process commits updates in 5000 archive increments.

As the archive migration run, the DTO pdf is read from table ERSAarchivedReport and converted to PDF (using the JasperReport objects and the campuses customizations in ErternalizedStrings.xml). Once the PDF is built, the DTO in ERSArchivedReports is overlaid with the PDF . The same process is used for the archive payroll report. The process continues until all ERSArchivedReport rows are processed or 100,000 rows are processed.

Larger campuses may have millions of rows in the ERSArchivedReport table, making it necessary to run the migration process 10+ times. Each iteration of the migration process will likely take a few days.

Run this SQL:

**select year(archived\_date) , count(\*) from ERSARCHIVEDREPORT group by year(archived\_date)**

If you see any archive dates < 2018, run the process again

Once all archive dates < 2018 are cleared, the ERS Archive Migration is complete.

**Performance:**

When this process was run for UCSB in QA (Jan 2018), it took 15 hours to migrate 100,000 archived effort reports.

**Step 11:** **Closing comments**

It is highly recommended that you long term archive a copy of the ERS 10.12 production database

If any issues arise after the UCPATH implementation, it may be necessary to load the ERS 10.12 system in a QA environment to retrieve or repair ERS data.

**Done with ERS 11.0 Upgrade !**

**Appendix 3 - Common DB2 Error Codes**

SQLCODE errors (some overlap with the SQLSTATE errors)

-205: column-name IS NOT A COLUMN OF TABLE table-name

-289: see 57011

-440: see 42884

-601: name already in use

-612: see 42711

-668: Access problem. Run reorg, then try again.
-20054 see 55019

SQLSTATE errors (some overlap with the SQLCODE errors)

42601: A character, token, or clause is invalid or missing

42710: name already in use

42711: A non-unique name was specified where a unique name is required. Column and period names must be unique within an index, a table, or a view, and in the UPDATE OF clause of a trigger definition.

42703: column-name IS NOT A COLUMN OF TABLE table-name

42704: object not defined

42884: It happened trying to run this command:

 insert into ERSDOSCODE values ('9AC','O',getdate(),'\*INSTALL\*',null,null)

Fixed by removing the 'getdate()' and replacing it with 'CURRENT TIMESTAMP' in the above command

42889: The table already has a primary key

55019: It happened trying to run either of these commands:

 ALTER TABLE ERSEARNINGS ALTER COLUMN Pay\_Rate SET NOT NULL;

 ALTER TABLE ERSEARNINGS ALTER COLUMN Paid\_Amount SET NOT NULL;

[Error Code: -20054, SQL State: 55019] DB2 SQL error: SQLCODE: -20054, SQLSTATE: 55019, SQLERRMC: ERSDEV4.ERSEARNINGS;23

Reason 23: The maximum number of REORG-recommended alters have been performed. Up to three REORG-recommended operations are allowed on a table before a reorg must be performed, to update the tables rows to match the current schema.

Fixed by running a reorg

57011: Failed during reorg. Tablespace ran out of space. DBA fixed by extending **ERSTS801** tablespace

See RITM0189264

57016: Access problem. Run reorg, then try again.

.