Query:

Friend Const GetHIPAAInsuranceInfoCall As String = "{Call hipaaInsuranceInfo(?,?,?)}"

**Procedure:**

**create** **procedure** hipaainsuranceinfo(rxnbr **integer**,

ibinnbr **integer**,dtrxfilldate **date**,

idebug **smallint** **default** 0)

**RETURNING** **integer**,**integer**, **integer**,**DATE**,**char**(1),**char**(30),**char**(40),**integer**,**integer**,**date**,**decimal**(9,4),**decimal**(9,4),**smallint**,**smallint**,

**datetime** **year** **to** **second**,**integer**,**decimal**(9,2),**decimal**(9,2),

**char**(30),**char**(13),**decimal**(9,4),**varchar**(90),

**date**,**char**(1),**char**(35),**char**(35),**char**(30),**char**(25),**char**(2),

**char**(10),**varchar**(90),**char**(35),**char**(35),**char**(30),**char**(30),**char**(25),

**char**(2),**char**(10),**varchar**(30),**smallint**,**integer**,**integer**,**decimal**(7,2),

**char**(20),**smallint**,**smallint**,**decimal**(9,2);

**define** irxnbr **integer**;

**define** irxid **integer**;

**define** ipresID **integer**;

**define** dtrx\_written\_date **DATE**;

**define** ccompound\_ind **char**(1);

**define** cpres\_prod\_name **char**(30);

**define** ccompound\_name **char**(40);

**define** irx\_input\_id **integer**;

**define** irx\_fill\_id **integer**;

**define** dtrx\_fill\_date **date**;

**define** dcpresc\_qty **decimal**(9,4);

**define** dcdispensed\_qty **decimal**(9,4);

**define** sdays\_supply **smallint**;

**define** sdaw\_code **smallint**;

**define** dtpos\_ts **datetime** **year** **to** **second**;

**define** iphm\_signature\_id **integer**;

**define** dcpos\_sale\_amt **decimal**(9,2);

**define** dcpos\_due\_amt **decimal**(9,2);

**define** cdisp\_prod\_name **char**(30);

**define** cndc\_nbr **char**(13);

**define** dcpack\_size **decimal**(9,4);

**define** cpatient\_name **varchar**(90);

**define** dtpat\_birth\_date **date**;

**define** cpat\_gender\_code **char**(1);

**define** cpat\_addr1 **char**(35);

**define** cpat\_addr2 **char**(35);

definecpat\_home\_phone **char**(30);

**define** cpat\_city\_name **char**(25);

**define** cpat\_state **char**(2);

**define** cpat\_zip\_code **char**(10);

**define** cpresc\_name **varchar**(90);

**define** cpresc\_addr1 **char**(35);

**define** cpresc\_addr2 **char**(35);

**define** cpresc\_home\_phone **char**(30);

definecpresc\_work\_phone **char**(30);

**define** cpresc\_city\_name **char**(25);

**define** cpresc\_state **char**(2);

**define** cpresc\_zip\_code **char**(10);

**define** cpresc\_lic\_nbr **varchar**(30);

**define** srefill\_auth\_code **smallint**;

**define** icarrier\_id **integer**;

**define** iplan\_id **integer**;

**define** ccardholder\_id **char**(20);

**define** dctxn\_amt **decimal**(7,2);

**define** stp\_status\_code **smallint**;

**define** crefills\_remain **char**(6);

**define** srefill\_auth\_nbr **smallint**;

**define** dcpat\_due\_amt **decimal**(9,2);

**define** CONST\_TP **smallint**;

**define** CONST\_PAT\_SUBJ\_CODE **smallint**;

**define** CONST\_PRESC\_SUBJ\_CODE **smallint**;

**define** CONST\_PAT\_ADDR\_TYPE\_CODE **smallint**;

**define** CONST\_COMM\_METHOD **smallint**;

**define** CONST\_HOME\_SEQ\_NBR **smallint**;

**define** CONST\_WORK\_SEQ\_NBR **smallint**;

**define** CONST\_PRES\_AGENCY\_CODE **smallint**;

**define** CONST\_PRES\_LICENSE\_CODE **smallint**;

**define** CONST\_STATUS\_CLAIMSUB **integer**;

**define** CONST\_STATUS\_NOCHARGE **integer**;

-- For Canada

**define** CONST\_COUNTRY\_US **varchar**(2);

**define** CONST\_COUNTRY\_CA **varchar**(2);

**define** cCountryCode **varchar**(3);

**define** CONST\_CA\_PRES\_LICENSE\_CODE **smallint**;

-- SCCSID: @(#) %name: % stored\_procedure scid=%version: % %date\_modified: % %derived\_by: %

**IF** idebug = 1 **THEN**

**SET** **DEBUG** **FILE** **TO** '/tmp/hipaainsuranceinfo.trace';

**TRACE** **ON**;

**END** **IF**

-- assigning values to variables and constants

**let** CONST\_TP = 1201;

**let** CONST\_PAT\_SUBJ\_CODE = 300;

**let** CONST\_PRESC\_SUBJ\_CODE = 303;

**let** CONST\_PAT\_ADDR\_TYPE\_CODE = 3;

**let** CONST\_COMM\_METHOD = 2;

**let** CONST\_HOME\_SEQ\_NBR = 1;

**let** CONST\_WORK\_SEQ\_NBR = 2;

**let** CONST\_PRES\_AGENCY\_CODE = 100;

**let** CONST\_PRES\_LICENSE\_CODE = 800;

**let** CONST\_STATUS\_CLAIMSUB = 21005;

**let** CONST\_STATUS\_NOCHARGE = 21011;

**LET** CONST\_CA\_PRES\_LICENSE\_CODE = 31101;

**let** CONST\_COUNTRY\_CA = "CA";

**let** CONST\_COUNTRY\_US = "US" ;

**Execute** **procedure** rx21c:fngetcountrycode() **into** cCountryCode ;

**If** cCountryCode = CONST\_COUNTRY\_CA **then**

**LET** CONST\_PRES\_LICENSE\_CODE = CONST\_CA\_PRES\_LICENSE\_CODE ;

**end** **if**

**FOREACH**

**select** rx.rx\_nbr,rx.rx\_id, rx.prescriber\_id, rx.rx\_written\_date,rx.compound\_ind,

**case** **when** rx.compound\_ind = 'N' thenpp1.short\_name

**end** presProdName,

**case** **when** rx.compound\_ind = 'Y' **then**

co.compound\_name

**end** CompoundName ,

-- rx and fill information

rxi.input\_id,rf.rx\_fill\_id ,rf.fill\_date, rx.fill\_qty ,

rf.fill\_qty , rf.fill\_days\_sply\_qty,rf.daw\_code,rf.pos\_ts,

rf.phm\_signature\_id,rf.pos\_sale\_amt,rf.pos\_due\_amt,

-- Dispensed Product Information

pp.short\_name DispProdName, pi.ndc\_nbr,

pi.inner\_pack\_qty Packsize,

-- Patient Information

**trim**(nvl(pat.last\_name,'')) || ',' || **trim**(nvl(pat.first\_name,'')) || ' ' || **trim**(nvl(pat.middle\_name,'')) Patientname,

pat.birth\_date,pat.gender\_code,sa.address\_line\_1,sa.address\_line\_2,

sc.communication\_id HomePhone,sa.city\_name, sa.state\_prov\_code, sa.postal\_code,

-- Prescriber Information

**trim**(nvl(pres.last\_name,'')) || ',' || **trim**(nvl(pres.first\_name,'')) || ' ' ||

**trim**(nvl(pres.middle\_name,'')) PrescriberName,

sad.address\_line\_1,sad.address\_line\_2,scd.communication\_id PrescHomePhone,

scd1.communication\_id PrescWorkPhone,sad.city\_name,sad.state\_prov\_code,sad.postal\_code,pl.license\_nbr,

-- TP Information

rx.refill\_auth\_code,rsd.aff\_party\_id, rsd.ins\_carr\_plan\_id,rsd.txn\_amt,

rsd.aff\_prty\_ref\_id, rsd.status\_code,

rx.refills\_auth\_nbr,

rf.patient\_due\_amt

**INTO**

irxnbr,irxid, ipresID,dtrx\_written\_date,ccompound\_ind,cpres\_prod\_name,ccompound\_name,

irx\_input\_id,irx\_fill\_id,dtrx\_fill\_date,dcpresc\_qty,dcdispensed\_qty,sdays\_supply,

sdaw\_code,dtpos\_ts,iphm\_signature\_id,dcpos\_sale\_amt,dcpos\_due\_amt,cdisp\_prod\_name, cndc\_nbr,dcpack\_size,cpatient\_name, dtpat\_birth\_date,cpat\_gender\_code,cpat\_addr1, cpat\_addr2,cpat\_home\_phone,cpat\_city\_name,cpat\_state,cpat\_zip\_code,cpresc\_name,

cpresc\_addr1,cpresc\_addr2,cpresc\_home\_phone,cpresc\_work\_phone,cpresc\_city\_name,

cpresc\_state,cpresc\_zip\_code,cpresc\_lic\_nbr,srefill\_auth\_code,icarrier\_id, iplan\_id,dctxn\_amt,ccardholder\_id,stp\_status\_code,

srefill\_auth\_nbr,dcpat\_due\_amt

**FROM**

rx rx, rx\_input rxi, rx\_fill rf, rx\_fill\_item rfi, patient pat,

prescriber pres,

pharmacy\_offering:pharmacy\_productpp,

pharmacy\_offering:pharmacy\_item pi,

rx\_sale\_detail rsd, phm\_company phm,

phm\_carr\_processor phmcp,

**outer** phm\_signature ps,

**outer** subject\_address sa,**outer** subject\_comm sc,

**outer** subject\_address sad, **outer** subject\_comm scd,

**outer** subject\_comm scd1,

**outer** pharmacy\_offering:pharmacy\_product pp1,

**outer** (rx\_compound rxc , compound co),

**outer** prescriber\_license pl,

**outer** pharmacy\_code\_txt pct

**WHERE**

rx.patient\_id = pat.patient\_id **AND** pat.patient\_id=sa.subject\_id

**AND** pat.patient\_id=sc.subject\_id **AND** rx.prescriber\_id = pres.prescriber\_id

**AND** pres.prescriber\_id = sad.subject\_id **AND** pres.prescriber\_id = scd.subject\_id

**AND** Pres.prescriber\_id = scd1.subject\_id **AND** pres.prescriber\_id = pl.prescriber\_id

**AND** rx.input\_id = rxi.input\_id **AND** rx.rx\_id = rf.rx\_id

**AND** rf.rx\_fill\_id = rfi.rx\_fill\_id **AND** rfi.item\_mds\_fam\_id = pi.mds\_fam\_id

**AND** pi.product\_mds\_fam\_id =pp.product\_mds\_fam\_id

**AND** rx.product\_mds\_fam\_id = pp1.product\_mds\_fam\_id

**AND** rx.rx\_id = rxc.rx\_id **AND** rxc.compound\_id = co.compound\_id

**AND** rf.phm\_signature\_id = ps.phm\_signature\_id

**AND** rsd.rx\_fill\_id = rf.rx\_fill\_id **AND** rsd.aff\_party\_id=phm.phm\_co\_id

**AND** rx.refill\_auth\_code = pct.code\_id

**AND** rsd.aff\_prty\_type\_code=CONST\_TP

**AND** rx.rx\_nbr = rxnbrAND ( ( rsd.status\_code = CONST\_STATUS\_CLAIMSUB)

**or** ( rsd.status\_code = CONST\_STATUS\_NOCHARGE) )

**AND** sa.subject\_type\_code= CONST\_PAT\_SUBJ\_CODE

**AND** sc.subject\_type\_code = CONST\_PAT\_SUBJ\_CODE

**AND** sa.addr\_type\_code = CONST\_PAT\_ADDR\_TYPE\_CODE

**AND** sc.comm\_method\_code = CONST\_COMM\_METHOD

**AND** sc.comm\_seq\_nbr = CONST\_HOME\_SEQ\_NBR

**AND** sad.subject\_type\_code= CONST\_PRESC\_SUBJ\_CODE

**AND** scd.subject\_type\_code = CONST\_PRESC\_SUBJ\_CODE

**AND** scd.comm\_method\_code = CONST\_COMM\_METHODAND scd.comm\_seq\_nbr = CONST\_HOME\_SEQ\_NBR

**AND** scd1.subject\_type\_code = CONST\_PRESC\_SUBJ\_CODE

**AND** scd1.comm\_method\_code = CONST\_COMM\_METHOD

**AND** scd1.comm\_seq\_nbr = CONST\_WORK\_SEQ\_NBR

-- AND pl.issue\_agency\_code = CONST\_PRES\_AGENCY\_CODE

ANDpl.license\_type\_code = CONST\_PRES\_LICENSE\_CODE

**AND** rsd.aff\_party\_id = phmcp.carrier\_co\_id

**AND** rsd.ins\_carr\_plan\_id = phmcp.ins\_carr\_plan\_id

**AND** phmcp.bin\_nbr = ibinnbr

**AND** rf.fill\_date = dtrxfilldate

**RETURN** irxnbr,irxid,ipresID,dtrx\_written\_date,ccompound\_ind,cpres\_prod\_name,ccompound\_name,

irx\_input\_id,irx\_fill\_id,dtrx\_fill\_date,dcpresc\_qty,dcdispensed\_qty,sdays\_supply,

sdaw\_code,dtpos\_ts,iphm\_signature\_id,dcpos\_sale\_amt,dcpos\_due\_amt,cdisp\_prod\_name, cndc\_nbr,dcpack\_size,cpatient\_name, dtpat\_birth\_date,cpat\_gender\_code,cpat\_addr1, cpat\_addr2,cpat\_home\_phone,cpat\_city\_name,cpat\_state,cpat\_zip\_code,cpresc\_name,

cpresc\_addr1,cpresc\_addr2,cpresc\_home\_phone,cpresc\_work\_phone,cpresc\_city\_name,

cpresc\_state,cpresc\_zip\_code,cpresc\_lic\_nbr,srefill\_auth\_code,icarrier\_id, iplan\_id,dctxn\_amt,ccardholder\_id,stp\_status\_code,

srefill\_auth\_nbr,dcpat\_due\_amt

**WITH** **RESUME**;

**END** **FOREACH**;

**IF** iDebug = 1 **THEN**

**TRACE** **OFF**;

**END** **IF**

**END** **PROCEDURE**;

Query:

Friend Const InsertReportTransactionlogCall As String = "{Call insertreporttranslog(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)}"

Procedure:

**CREATE** **PROCEDURE** insertreporttranslog(irqst\_part\_id **integer**,

srqstor\_type\_cd **smallint**,

dtrequest\_ts **datetime** **year** **to** **second**,

crequestor\_name **char**(30),dtrequest\_start\_ts **datetime** **year** **to** **second**,

dtrequest\_end\_ts **datetime** **year** **to** **second**,

dtdisclosure\_ts **datetime** **year** **to** **second**,

isignature\_id **integer**,

ccreate\_useridchar(10),

crequest\_txt **varchar**(255),

srelationship\_code **smallint**,

sreport\_type\_code **smallint**,

ipatient\_id **integer** **DEFAULT** 0,

irx\_id **integer** **DEFAULT** 0,

cSQLWhereQry LIST(**lvarchar** **not** **null**),cemail\_addr\_txt **char**(60) **DEFAULT** **NULL**,

cmulti\_location\_ind **char**(1) **DEFAULT** 'N',

irequest\_format\_type\_cd **integer** **DEFAULT** 10,

csold\_rx\_ind **char**(1) **DEFAULT** 'Y',

cprscbr\_dea\_ind **char**(1) **DEFAULT** 'Y',

irestrict\_type\_code **integer** **DEFAULT** 401,

irqst\_status\_code **INTEGER** **DEFAULT** 1001,

isubj\_assn\_id **INTEGER** **DEFAULT** **NULL**,

cdrug\_redaction\_ind **CHAR**(1) **DEFAULT** **NULL**,

ccust\_id\_valdt\_ind **CHAR**(1) **DEFAULT** **NULL**,

iDebug **smallint** **DEFAULT** 0)**returning** **integer**,**integer** ;

**define** in\_trans **smallint**;

**define** sql\_err\_code **integer**;

**define** isam\_err\_code **integer**;

**define** err\_desc **varchar**(255);

**define** p\_ephi\_bu\_parm **char**(40);

**define** ilrx\_id **integer**;

**define** ihipaa\_tran\_id **integer**;

**define** srx\_ver\_seq\_nbr **smallint**;

**define** lstmt **lvarchar**;

**define** lretVal **lvarchar**;

**define** strExecSQL **lvarchar**;

**define** CONSTRXBUILD **integer**;

**define** CONSTPSEUDOREPORTTYPE **integer**;

**define** CONSTEPHIFLAG **integer**;

**define** CONSTMESRPT **integer**;

**define** CONSTDRSRPT **integer**;

**define** CONSTHIPAARQSTR **integer**;

-- SCCSID: @(#) name: %M% stored\_procedure scid=version: %I% date\_modified: %E% derived\_by: \_\_PWR\_CHG\_USER\_\_

**on** **exception** **set** sql\_err\_code,isam\_err\_code,err\_desc

**if** in\_trans=0**then**

**rollback** **work**;

**return** -1,-1;

**end** **if**

**raise** **exception** sql\_err\_code,isam\_err\_code,err\_desc;

**end** **exception**;

**on** **exception** **in** (-535,-744)

**let** in\_trans=1;

**end** **exception** **with** **resume**;

**IF** iDebug = 1 **THEN**

**SET** **DEBUG** **FILE** **TO** '/tmp/insertreporttranslog.trace';

**TRACE** **ON**;

**END** **IF**

**let** in\_trans=0;

**if** in\_trans=0

**then**

**begin** **work**;

**end** **if**

-- Validate Input Parameters

**let** ilrx\_id = 0;

**let** p\_ephi\_bu\_parm = "FALSE";

**let** CONSTRXBUILD = 20600;

**let** CONSTPSEUDOREPORTTYPE = 9639 ;

**let** CONSTEPHIFLAG = 32725 ;

**let** CONSTMESRPT = 9600 ;

**let** CONSTDRSRPT = 9601 ;

**let** CONSTHIPAARQSTR = 9500;

--Get EPHI flag value

**SELECT** NVL(**TRIM**(phm\_parm\_value),'FALSE')

**INTO** p\_ephi\_bu\_parm

**FROM** bu\_phm\_parm

**WHERE** phm\_parm\_code = CONSTEPHIFLAG

**AND** phm\_bu\_id **IN** (**SELECT** bu\_nbr **FROM** bus\_unit\_addr);

**if** (isignature\_id <=0 ) **then**

**LET** isignature\_id=**NULL**;

**end** **if**

**if** srelationship\_code <=0 **then**

**LET** srelationship\_code=**NULL**;

**end** **if**

-- Insert into Hippa Info Request Table

**INSERT** **INTO** Hipaa\_info\_Request

(rqstg\_party\_id, requestor\_type\_cd, request\_ts, requestor\_name,

request\_start\_ts, request\_end\_ts, disclosure\_ts, phm\_signature\_id,

create\_ts, create\_userid, request\_txt, relationship\_code,request\_type\_code)**VALUES**

(irqst\_part\_id,srqstor\_type\_cd, dtrequest\_ts, crequestor\_name,

dtrequest\_start\_ts,dtrequest\_end\_ts, dtdisclosure\_ts,isignature\_id,

**CURRENT**,ccreate\_userid,crequest\_txt,srelationship\_code,sreport\_type\_code);

**LET** ihipaa\_tran\_id = dbinfo("SQLCA.SQLERRD1");

-- HIPAA EPHI Chnages - Start

--Insert into hipaa\_ephi\_request table for processing of HIPAA MES and DRS requests

**IF** p\_ephi\_bu\_parm ="TRUE" **THEN**

**IF** ihipaa\_tran\_id > 0 **AND** (sreport\_type\_code = CONSTMESRPT **OR** sreport\_type\_code= CONSTDRSRPT)

**AND** srqstor\_type\_cd = CONSTHIPAARQSTR **THEN**

**INSERT** **INTO** hipaa\_ephi\_request

(hipaa\_info\_rqst\_id,email\_address\_id,multi\_location\_ind,

rqst\_format\_type\_code,sold\_rx\_ind,prescriber\_dea\_ind,

restrict\_type\_code,rqst\_status\_code,subj\_assn\_id,

drug\_redaction\_ind,cust\_id\_valdt\_ind)

**VALUES**

(ihipaa\_tran\_id,cemail\_addr\_txt,cmulti\_location\_ind,

irequest\_format\_type\_cd,csold\_rx\_ind,cprscbr\_dea\_ind,

irestrict\_type\_code,irqst\_status\_code,isubj\_assn\_id,cdrug\_redaction\_ind,ccust\_id\_valdt\_ind);

**END** **IF**

**END** **IF**

--End EPHI Changes

-- This patient id comes when the agency looks at the MES, DES, DIS

**if** ipatient\_id > 0 **then**

**INSERT** **INTO** Hipaa\_Rqst\_patient

(hipaa\_info\_rqst\_id, patient\_id)

**VALUES**

(ihipaa\_tran\_id,ipatient\_id);

**end** **if**

-- The rx\_id comes during the insurance looks at the patient rxs.

**if** irx\_id > 0 **then**

**select** rx.rx\_version\_seq\_nbr

**into** srx\_ver\_seq\_nbr

**from** rx **where** rx\_id = irx\_id;

**INSERT** **INTO** Hipaa\_Info\_Rqst\_Rx

(hipaa\_info\_rqst\_id, rx\_id, rx\_version\_seq\_nbr)

**VALUES**

(ihipaa\_tran\_id,irx\_id,srx\_ver\_seq\_nbr);

**end** **if**

**if** ipatient\_id = 0

**and** irx\_id =0 **and** sreport\_type\_code <> CONSTPSEUDOREPORTTYPE **THEN**

**FOREACH** **SELECT** a.execfromlist **INTO** strExecSQL **FROM** **table**(cSQLWhereQry)a(execfromlist)

**IF** strExecSQL = 'NO RXS' **then**

**EXIT** **FOREACH**;

**END** **IF**

**LET** lstmt = ' insert into Hipaa\_Info\_Rqst\_Rx (rx\_id,rx\_version\_seq\_nbr,hipaa\_info\_rqst\_id)';

**LET** lstmt = lstmt || ' ' || ' select distinct rx.rx\_id, rx.rx\_version\_seq\_nbr, ' || ihipaa\_tran\_id ;

**LET** lstmt = lstmt || ' ' || ' from rx where rx\_id in ' || strExecSQL ;**LET** lretval = **exec**(lstmt);

**END** **FOREACH**

**end** **if**

**if** ipatient\_id = 0

**and** irx\_id =0 **and** sreport\_type\_code = CONSTPSEUDOREPORTTYPE **THEN**

**FOREACH** **SELECT** a.execfromlist **INTO** strExecSQL **FROM** **table**(cSQLWhereQry)a(execfromlist)

IFstrExecSQL = 'NO RXS' **then**

**EXIT** **FOREACH**;

**END** **IF**

**LET** lstmt = ' insert into Hipaa\_rqst\_patient (patient\_id, hipaa\_info\_rqst\_id)';

**LET** lstmt = lstmt || ' ' || ' select distinct pat.patient\_id, ' || ihipaa\_tran\_id ;

**LET** lstmt = lstmt || ' ' || ' from patient pat where pat.patient\_id in ' || strExecSQL ;

**LET** lretval = **exec**(lstmt);

**END** **FOREACH**

**end** **if**

**if** in\_trans=0

**then**

**commit** **work**;

**end** **ifque**

**return** 1,ihipaa\_tran\_id;

**IF** iDebug = 1 **THEN**

**TRACE** **OFF**;

**END** **IF**

**END** **PROCEDURE**;

DAL

''' <summary>

''' Inserts report transaction log information into the database

''' </summary>

''' <param name="siteId"></param>

''' <param name="SQLWhereQuery">SQL where query from the business logic layer</param>

''' <param name="reportTransactionLog">report Transaction log entity</param>

''' <returns>report transaction log</returns>

''' <remarks>Refactored</remarks>

Public Function InsertReportTransactionLog(ByVal siteId As Integer,

ByVal sqlWhereQuery As String,

ByVal reportTransactionLog \_

As ReportTransactionLogBE, Optional ByRef hipaaRequestId As Integer = 0) As Integer Implements IHIPAADAL.InsertReportTransactionLog

Using tracer As New Tracer(ConnexusCommonConstants.ConnexusTracer)

Dim paramList As New List(Of DbParameter)

paramList.AddParameter(DbType.Int32, reportTransactionLog.RequestingPartyId)

paramList.AddParameter(DbType.Int16, reportTransactionLog.RequestorTypeCode)

paramList.AddParameter(DbType.DateTime, reportTransactionLog.RequestTS)

paramList.AddParameter(DbType.String, reportTransactionLog.RequestorName)

paramList.AddParameter(DbType.DateTime, reportTransactionLog.RequestStartTS)

paramList.AddParameter(DbType.DateTime, reportTransactionLog.RequestEndTS)

paramList.AddParameter(DbType.DateTime, reportTransactionLog.DisclosureTS)

paramList.AddParameter(DbType.Int32, reportTransactionLog.SignatureId)

paramList.AddParameter(DbType.String, reportTransactionLog.UserId)

paramList.AddParameter(DbType.String, reportTransactionLog.RequestText)

paramList.AddParameter(DbType.Int16, reportTransactionLog.RelationCode)

paramList.AddParameter(DbType.Int16, reportTransactionLog.ReportTypeCode)

paramList.AddParameter(DbType.Int32, reportTransactionLog.PatientId)

paramList.AddParameter(DbType.Int32, reportTransactionLog.RxId)

paramList.AddParameter(DbType.String, sqlWhereQuery)

paramList.AddParameter(DbType.String, reportTransactionLog.EmailAddressTxt)

paramList.AddParameter(DbType.String, reportTransactionLog.MultiLocationInd)

paramList.AddParameter(DbType.Int32, reportTransactionLog.ReqFormatTypeCode)

paramList.AddParameter(DbType.String, reportTransactionLog.SoldRxInd)

paramList.AddParameter(DbType.String, reportTransactionLog.PrscbrDEAInd)

paramList.AddParameter(DbType.Int32, reportTransactionLog.RestrictTypeCode)

paramList.AddParameter(DbType.Int32, reportTransactionLog.RequestStatusCode)

'HIPAA - Personal reps Changes - Start

'Change : Added a param to save the requestor details to the hipaa\_ephi\_request table

If reportTransactionLog.RequestorId <= 0 Then

paramList.AddParameter(DbType.Int32, System.DBNull.Value)

Else

paramList.AddParameter(DbType.Int32, reportTransactionLog.RequestorId)

End If

'HIPAA Personal Representatives Phase 2 project changes - Start

'Passing 'photo id verified' and 'redaction indicator' to SP

paramList.AddParameter(DbType.String, reportTransactionLog.RedactionInd)

paramList.AddParameter(DbType.String, reportTransactionLog.IdConfirmed)

'HIPAA Personal Representatives Phase 2 project changes - End

'HIPAA- Personal reps Changes - End

Dim returnTransactionLog As Integer

Using reader As IDataReader = ExecuteReader(DatabaseEnum.rx21c, siteId, HipaaQueries.InsertReportTransactionlogCall, paramList)

If reader IsNot Nothing Then

While reader.Read()

returnTransactionLog = reader.GetValueOrDefault(Of Int32)(0)

hipaaRequestId = reader.GetValueOrDefault(Of Int32)(1)

End While

End If

End Using

Return returnTransactionLog

End Using

End Function

Public Shared Function GetNewImage(ByVal siteId As Integer, ByVal columnName As String, ByVal tableName As String, ByVal whereClause As String) As Byte()

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

Dim byteArray As Byte() = Nothing

Dim querySQL As New StringBuilder

'File to save image from database

Dim fileName As String = ConnexusHelper.GetUniqueFilePath(ServicesCommonConstants.ImagesFolder, ServicesCommonConstants.ImageName, ServicesCommonConstants.TifExtension)

'Construct the Query

querySQL.AppendFormat("SELECT lotofile({0},'{1}!','client') raw\_column\_data FROM {2} WHERE {3}", columnName, fileName, tableName, whereClause)

Dim responseTxt As String = Nothing

Using dr As IDataReader = Daab.ExecuteReader(DatabaseEnum.rx21c, siteId, querySQL.ToString())

'TODO: Remove after DB refresh of 5518

'Following try block is done because of data issue. At some data

' it was obeserved that just doing dr.Read gives error. So checking for dr.HasRows

Try

If dr.Read Then

If Not dr.IsDBNull(dr.GetOrdinal("raw\_column\_data")) Then

responseTxt = CStr(dr("raw\_column\_data")).Trim

dr.Close()

dr.Dispose()

Dim retry As Integer

'Retrying in case the file is locked for writing the blob data

While retry <= 2

Try

byteArray = System.IO.File.ReadAllBytes(responseTxt)

Exit While

Catch ex As Exception

LogHandler.LogError(ex, ContextWrapper.ActivityStoreNumber, ContextWrapper.UserStoreNumber, ContextWrapper.UserName, ContextWrapper.WorkStationName, "Error while retrieving image")

Thread.Sleep(1000)

retry = retry + 1

End Try

End While

Try

System.IO.File.Delete(responseTxt)

Catch

End Try

End If

End If

Catch ex As Exception

LogFailedQuery(DatabaseEnum.rx21c, querySQL.ToString, Nothing, ex)

Throw

End Try

End Using

Return byteArray

End Using

End Function