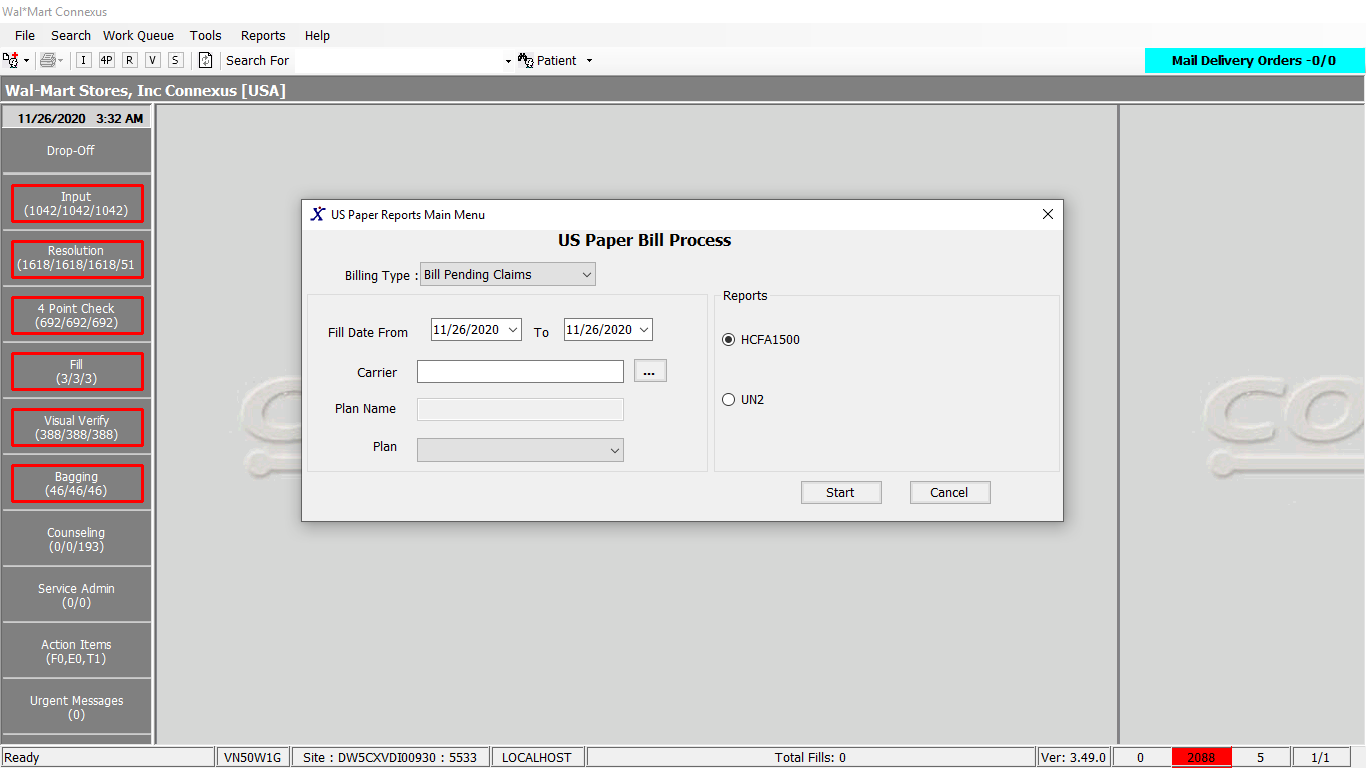
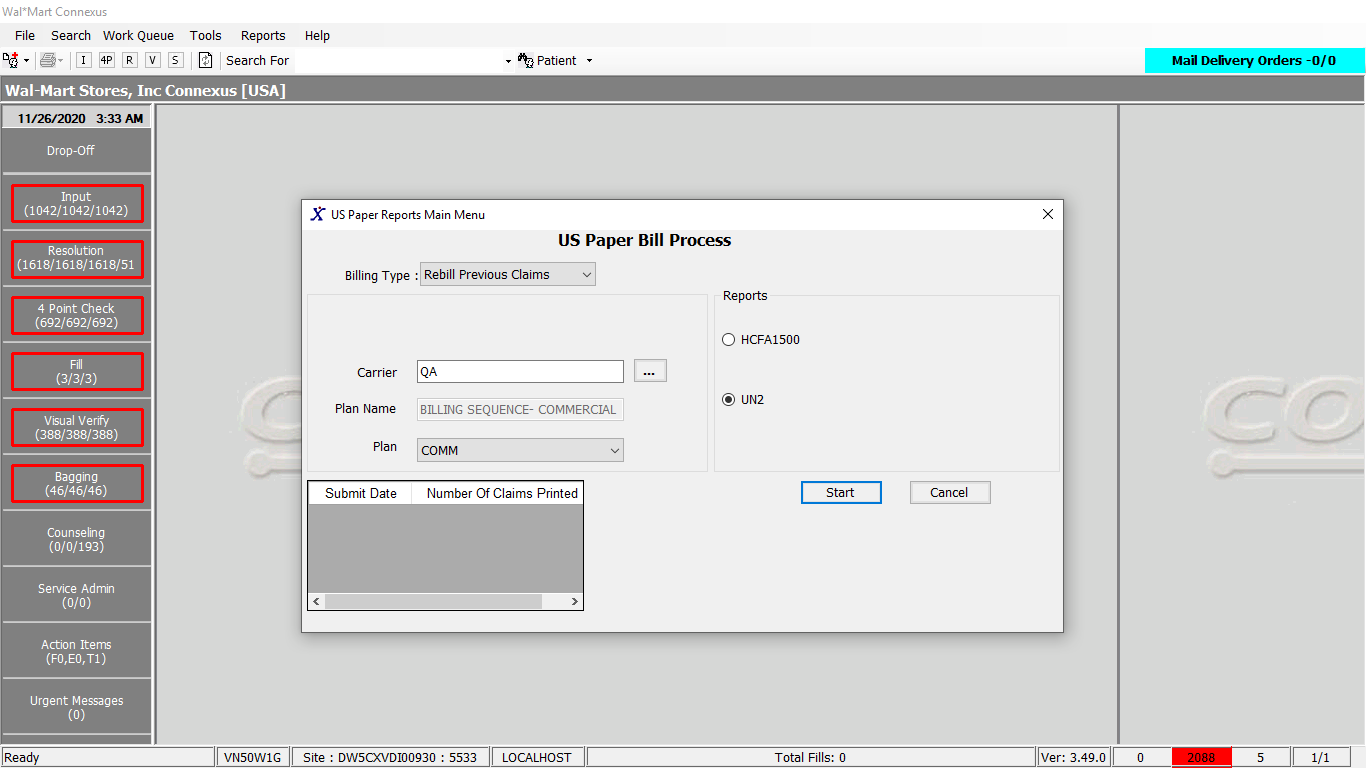
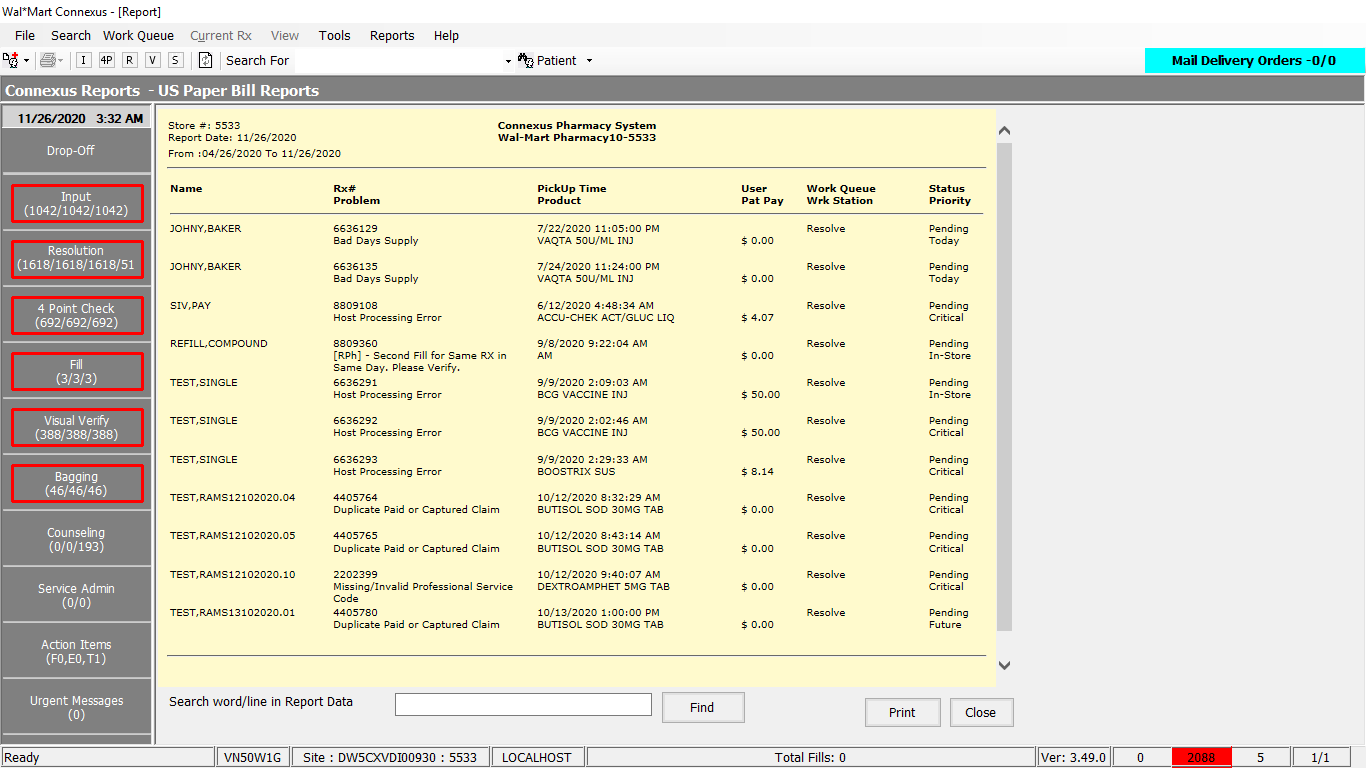
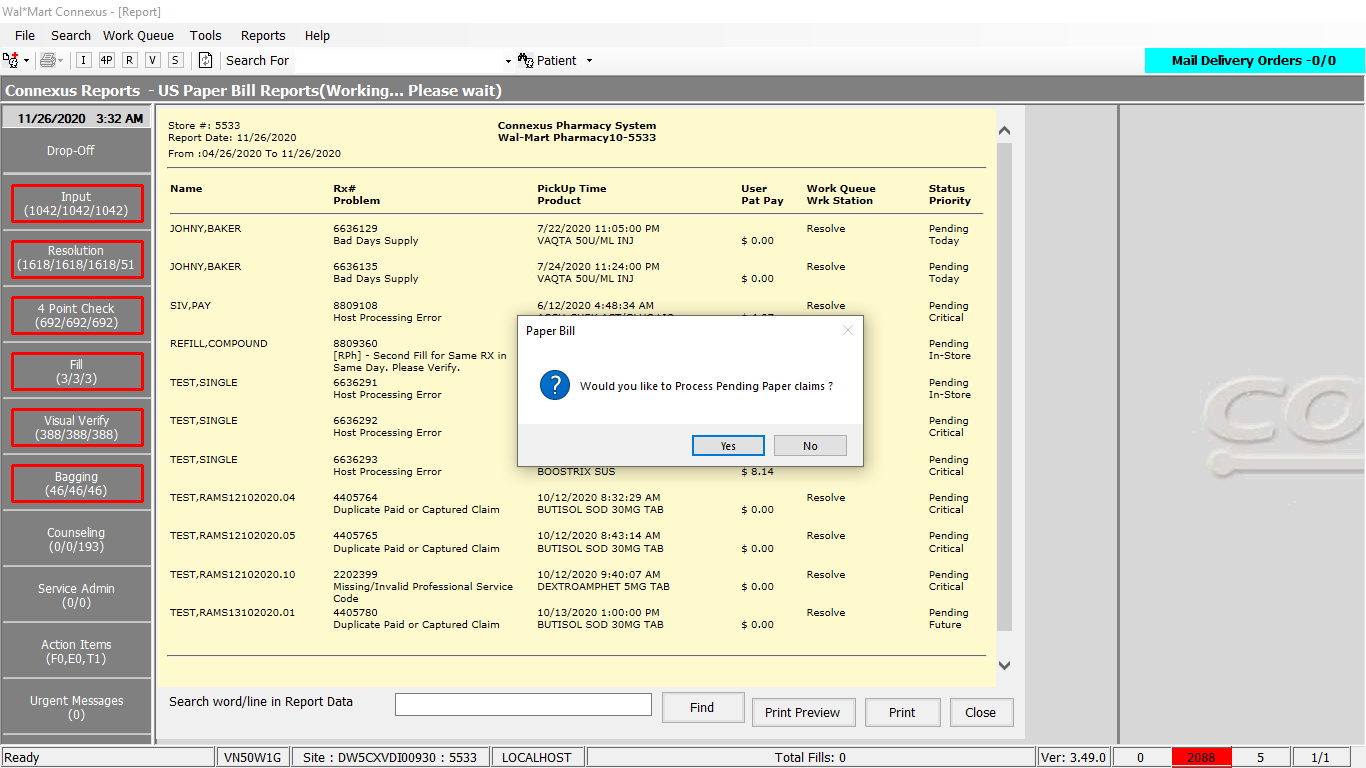
## Reports/ US Paper Bill Reports:

Request:



Response:



Queries:

1. Public Const GetTPCoverageType As String = " SELECT coverage\_option\_id " & \_

" From covg\_opt\_restrict " & \_

" WHERE carrier\_co\_id = ? " & \_

" AND ins\_carr\_plan\_id = ? " & \_

" AND restriction\_type = ? " & \_

" AND coverage\_option\_id = ? " & \_

" AND restriction\_nbr = 'Y'; "

2. Public Const SendOrderToTP As String = "{ call run\_tp\_and\_dur(?,?,?,?,?,?,?)}"

3. Friend Const GetCAExceptReportSelectPart1 As String = "SELECT rod.order\_id, trim(nvl(pt.last\_name,'')) || ',' || trim(nvl(pt.first\_name,'')) || ' ' || trim(nvl(pt.middle\_name,'')) Name, rx.rx\_nbr, rod.est\_pickup\_ts, trim(wqt.work\_que\_abbr) work\_que\_abbr, trim(pct.code\_desc) code\_desc, trim(rod.assigned\_userid) assigned\_userid, CASE WHEN rx.compound\_ind = 'Y' THEN ( SELECT comp.compound\_name FROM compound comp, rx\_compound rxc WHERE rx.rx\_id = rxc.rx\_id AND rxc.compound\_id = comp.compound\_id ) WHEN (rod.request\_type\_code = ? OR rod.request\_type\_code = ? ) THEN ( SELECT drug\_name FROM phm\_rx\_transfer prt WHERE prt.rx\_id = rod.rx\_id AND prt.transfer\_stat\_code = ? ) ELSE trim(pp.short\_name) END product\_name, CASE WHEN next\_work\_que\_code = ? AND rxf.status\_code IN (?, ?, ?) THEN '"

4. Friend Const GetCAExceptReportSelectPart2 As String = "' WHEN next\_work\_que\_code = ? THEN trim(ppt.phm\_problem\_desc) ELSE trim(rp.problem\_comment) END comment, concat('$', round(rxf.patient\_due\_amt,2)) Price, pct2.code\_desc prioritydesc , rod.wrkstatn\_host\_name, rod.order\_seq\_nbr, rod.next\_work\_que\_code, rod.priority\_id, ppb.trblsht\_method\_cd, rod.rx\_fill\_id, rxf.status\_code fillstatus, rod.request\_type\_code FROM rx\_order\_detail rod, rx rx, rx\_fill rxf, rx\_sale\_detail rsd, rx\_fill\_item rfi, patient pt, pharmacy\_offering:pharmacy\_item pi, OUTER pharmacy\_offering:pharmacy\_product pp, pharmacy\_code\_txt pct, pharmacy\_code\_txt pct2, work\_que\_txt wqt, rx\_ord\_dtl\_problem rp, phm\_problem\_txt ppt, phm\_problem ppb WHERE rx.rx\_id = rod.rx\_id AND rx.rx\_id = rxf.rx\_id AND rod.status\_code = ? AND rsd.rx\_fill\_id = rx.last\_rx\_fill\_id AND rsd.rx\_fill\_id = rxf.rx\_fill\_id AND rsd.aff\_prty\_type\_code = ? AND rp.order\_id = rod.order\_id AND rp.order\_seq\_nbr = rod.order\_seq\_nbr AND rp.problem\_stat\_code IN (?, ?) AND ppb.phm\_problem\_code = rp.phm\_problem\_code AND rp.problem\_seq\_nbr = ( SELECT MIN(problem\_seq\_nbr) FROM rx\_ord\_dtl\_problem rodp WHERE rodp.order\_id = rod.order\_id AND rodp.order\_seq\_nbr = rod.order\_seq\_nbr AND rodp.problem\_stat\_code IN (?, ?)) AND ppt.phm\_problem\_code = rp.phm\_problem\_code AND wqt.work\_que\_code = ? AND ppb.trblsht\_method\_cd in (7,8,41,42,43,44,45) AND ppt.language = ? AND pt.patient\_id = rx.patient\_id AND pct.code\_id = rod.work\_que\_stat\_code AND pct2.code\_id = rod.priority\_id AND rod.work\_que\_stat\_code <> ? AND rod.next\_work\_que\_code <> ? AND pct.language\_code = ? AND wqt.work\_que\_code = rod.next\_work\_que\_code AND wqt.language\_code = ? AND rod.rx\_fill\_id = rfi.rx\_fill\_id AND rod.rx\_fill\_id= rxf.rx\_fill\_id AND rxf.rx\_fill\_id = rfi.rx\_fill\_id AND pi.mds\_fam\_id = rfi.item\_mds\_fam\_id AND rfi.item\_seq\_nbr = 1 AND pp.product\_mds\_fam\_id = pi.product\_mds\_fam\_id AND rsd.aff\_party\_id = ? AND rsd.ins\_carr\_plan\_id = ? AND rxf.fill\_date BETWEEN ? AND ? "

DAL:

Public Function CheckTPGenCoverageType(ByVal siteId As Integer, ByVal carrierId As Integer, ByVal planId As Integer, ByVal resType As Integer) As Boolean Implements IThirdPartyDAL.CheckTPGenCoverageType

Using New Tracer(ConnexusCommonConstants.ConnexusTracer)

'check from cache first

Dim TpCoverageInfo =

From cor In GetCoverageOptionRestrictions(siteId, carrierId, planId, ServicesCommonConstants.CoverageOptionId, resType).RemoveNulls()

Where cor.RestrictionNbr IsNot Nothing AndAlso cor.RestrictionNbr.Equals("Y")

Select cor.CoverageOptionId

If TpCoverageInfo.HasRecords() Then

CheckTPGenCoverageType = True

Else

'get it from DB

LogHandler.LogWarning("There was a Cache Miss for CheckTPGenCoverageType", ContextWrapper.ActivityStoreNumber, ContextWrapper.UserStoreNumber, ContextWrapper.UserName, ContextWrapper.WorkStationName)

Dim paramList As New List(Of DbParameter)

paramList.AddParameter(DbType.Int32, carrierId)

paramList.AddParameter(DbType.Int32, planId)

paramList.AddParameter(DbType.Int16, resType)

paramList.AddParameter(DbType.Int16, ServicesCommonConstants.CoverageOptionId)

Using dr As IDataReader = ExecuteReader(DatabaseEnum.rx21c, siteId, ThirdPartyQueries.GetTPCoverageType, paramList)

If dr.Read() Then

CheckTPGenCoverageType = True

Else

CheckTPGenCoverageType = False

End If

End Using

End If

End Using

End Function

Public Function CheckTPGenCoverageType(ByVal siteId As Integer, ByVal carrierId As Integer, ByVal planId As Integer, ByVal resType As Integer) As Boolean Implements IThirdPartyDAL.CheckTPGenCoverageType

Using New Tracer(ConnexusCommonConstants.ConnexusTracer)

'check from cache first

Dim TpCoverageInfo =

From cor In GetCoverageOptionRestrictions(siteId, carrierId, planId, ServicesCommonConstants.CoverageOptionId, resType).RemoveNulls()

Where cor.RestrictionNbr IsNot Nothing AndAlso cor.RestrictionNbr.Equals("Y")

Select cor.CoverageOptionId

If TpCoverageInfo.HasRecords() Then

CheckTPGenCoverageType = True

Else

'get it from DB

LogHandler.LogWarning("There was a Cache Miss for CheckTPGenCoverageType", ContextWrapper.ActivityStoreNumber, ContextWrapper.UserStoreNumber, ContextWrapper.UserName, ContextWrapper.WorkStationName)

Dim paramList As New List(Of DbParameter)

paramList.AddParameter(DbType.Int32, carrierId)

paramList.AddParameter(DbType.Int32, planId)

paramList.AddParameter(DbType.Int16, resType)

paramList.AddParameter(DbType.Int16, ServicesCommonConstants.CoverageOptionId)

Using dr As IDataReader = ExecuteReader(DatabaseEnum.rx21c, siteId, ThirdPartyQueries.GetTPCoverageType, paramList)

If dr.Read() Then

CheckTPGenCoverageType = True

Else

CheckTPGenCoverageType = False

End If

End Using

End If

End Using

End Function

Public Function GetCAExceptRep(ByVal siteId As Integer, ByVal carrCOId As Integer,

ByVal insCarrPlanId As Integer,

ByVal fromDate As DateTime, ByVal toDate As DateTime) \_

As List(Of GetCAExceptRepInfoBE) Implements IReportDAL.GetCAExceptRep

'Declare the method variables

Dim sqlString As New StringBuilder

Dim caExceptRepInfoBEList As New List(Of GetCAExceptRepInfoBE)

Dim paramList As New List(Of DbParameter)

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

sqlString.Append(ReportQueries.GetCAExceptReportSelectPart1)

sqlString.Append(ServicesCommonConstants.NeedsVoid)

sqlString.Append("' WHEN next\_work\_que\_code = ? AND rxf.status\_code = ? ")

sqlString.Append(" AND rxf.pos\_ts IS NULL THEN '")

sqlString.Append(ServicesCommonConstants.NeedsSold)

sqlString.Append("' WHEN next\_work\_que\_code = ? AND rxf.status\_code = ? ")

sqlString.Append(" AND rxf.pos\_ts IS NOT NULL AND rod.need\_pickup\_ind = 'Y'")

sqlString.Append(" THEN '")

sqlString.Append(ServicesCommonConstants.NeedsVoid)

sqlString.Append(ReportQueries.GetCAExceptReportSelectPart2)

paramList.Add(ParameterHelper.BuildParameter("requestTypeCode", DbType.Int16,

RequestTypeCodes.TypeTransferRequest))

paramList.Add(ParameterHelper.BuildParameter("requestTypeCode", DbType.Int16,

RequestTypeCodes.TypeTransferResponse))

paramList.Add(ParameterHelper.BuildParameter("transferStatCode", DbType.Int16,

TransferStatus.TransferPending))

paramList.Add(ParameterHelper.BuildParameter("nextWorkQueueCode", DbType.Int16,

Station.stationPos))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

FillStatusCodes.FillCancelled))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

FillStatusCodes.FillShrink))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

FillStatusCodes.FillWriteOff))

paramList.Add(ParameterHelper.BuildParameter("nextWorkQueueCode", DbType.Int16,

Station.stationPos))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

FillStatusCodes.FillActive))

paramList.Add(ParameterHelper.BuildParameter("nextWorkQueueCode", DbType.Int16,

Station.stationPos))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

FillStatusCodes.FillActive))

paramList.Add(ParameterHelper.BuildParameter("nextWorkQueueCode", DbType.Int16,

Station.StationTrouble))

paramList.Add(ParameterHelper.BuildParameter("statusCode", DbType.Int16,

ServicesCommonConstants.RxOrdAct))

paramList.Add(ParameterHelper.BuildParameter("affPartyTypeCode", DbType.Int16,

AffPartyTypeCode.AffPartyTypeThirdParty))

paramList.Add(ParameterHelper.BuildParameter("problemStatCode", DbType.Int16,

ServicesCommonConstants.PrblActive))

paramList.Add(ParameterHelper.BuildParameter("problemStatCode", DbType.Int16,

ServicesCommonConstants.PrblPending))

paramList.Add(ParameterHelper.BuildParameter("problemStatCode", DbType.Int16,

ServicesCommonConstants.PrblActive))

paramList.Add(ParameterHelper.BuildParameter("problemStatCode", DbType.Int16,

ServicesCommonConstants.PrblPending))

paramList.Add(ParameterHelper.BuildParameter("workQueueCode", DbType.Int16,

Station.StationTrouble))

paramList.Add(ParameterHelper.BuildParameter("language", DbType.String,

ServicesCommonConstants.LangEng))

paramList.Add(ParameterHelper.BuildParameter("workQueueStatCode", DbType.Int16,

WorkQueStatus.WorkQueComplete))

paramList.Add(ParameterHelper.BuildParameter("nextWorkQueueCode", DbType.Int16,

WorkQueueSettings.DropOff))

paramList.Add(ParameterHelper.BuildParameter("languageCode", DbType.String,

ServicesCommonConstants.LangEng))

paramList.Add(ParameterHelper.BuildParameter("languageCode", DbType.String,

ServicesCommonConstants.LangEng))

paramList.Add(ParameterHelper.BuildParameter("affPartyId", DbType.Int32,

carrCOId))

paramList.Add(ParameterHelper.BuildParameter("insCarrierPlanId", DbType.Int32,

insCarrPlanId))

paramList.Add(ParameterHelper.BuildParameter("fillDate", DbType.Date, fromDate))

paramList.Add(ParameterHelper.BuildParameter("fillDate", DbType.Date, toDate))

Dim dataReader As IDataReader = ExecuteReader(DatabaseEnum.rx21c, siteId,

sqlString.ToString, paramList)

Dim caExceptRepInfoBE As GetCAExceptRepInfoBE

While dataReader.Read()

caExceptRepInfoBE = New GetCAExceptRepInfoBE

If Not dataReader.IsDBNull(dataReader.GetOrdinal("assigned\_userid")) Then

caExceptRepInfoBE.AssignedUserId = CStr(dataReader("assigned\_userid")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("code\_desc")) Then

caExceptRepInfoBE.CodeDesc = CStr(dataReader("code\_desc")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("comment")) Then

caExceptRepInfoBE.Comment = CStr(dataReader("comment")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("est\_pickup\_ts")) Then

caExceptRepInfoBE.EstPickupTs = CDate(dataReader("est\_pickup\_ts"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("next\_work\_que\_code")) Then

caExceptRepInfoBE.NextWorkQueCode = CInt(dataReader("next\_work\_que\_code"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("order\_id")) Then

caExceptRepInfoBE.OrderId = CInt(dataReader("order\_id"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("Name")) Then

caExceptRepInfoBE.PatName = CStr(dataReader("Name")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("Price")) Then

caExceptRepInfoBE.Price = CDec(dataReader("Price"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("priority\_id")) Then

caExceptRepInfoBE.PriorityId = CInt(dataReader("priority\_id"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("product\_name")) Then

caExceptRepInfoBE.ProductName = CStr(dataReader("product\_name")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("request\_type\_code")) Then

caExceptRepInfoBE.RequestTypeCode = CInt(dataReader("request\_type\_code"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("rx\_fill\_id")) Then

caExceptRepInfoBE.RxFillId = CInt(dataReader("rx\_fill\_id"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("rx\_nbr")) Then

caExceptRepInfoBE.RxNbr = CInt(dataReader("rx\_nbr"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("fillstatus")) Then

caExceptRepInfoBE.StatusCode = CInt(dataReader("fillstatus"))

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("trblsht\_method\_cd")) Then

caExceptRepInfoBE.TroubleshootMethodCode = CStr(dataReader("trblsht\_method\_cd")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("work\_que\_abbr")) Then

caExceptRepInfoBE.WorkQueAbbr = CStr(dataReader("work\_que\_abbr")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("wrkstatn\_host\_name")) Then

caExceptRepInfoBE.WorkStationHostName = CStr(dataReader("wrkstatn\_host\_name")).Trim

End If

If Not dataReader.IsDBNull(dataReader.GetOrdinal("prioritydesc")) Then

caExceptRepInfoBE.PriorityDesc = CStr(dataReader("prioritydesc")).Trim

End If

'Add the BE item to the BEList

caExceptRepInfoBEList.Add(caExceptRepInfoBE)

End While

'Return the BE List

If caExceptRepInfoBEList.Count > 0 Then

Return caExceptRepInfoBEList

Else

Return Nothing

End If

End Using

End Function

--------------------------- ---------------------------------------------

Public Function GetPaperBillOrders(ByVal siteId As Integer, ByVal fillStartDate As Date, ByVal fillEndDate As Date, ByVal carrierId As Integer, ByVal planId As Integer, ByVal statusCode As Integer, Optional ByVal cardHolderId As String = "") As List(Of PaperBillOrdersBE) Implements IThirdPartyDAL.GetPaperBillOrders

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

Dim strSQL As New StringBuilder

Dim paperBillOrderList As New List(Of PaperBillOrdersBE)

strSQL.Append(ThirdPartyQueries.GetPaperBillOrders)

If Not String.IsNullOrEmpty(cardHolderId) Then

strSQL.Append(" AND rsd.aff\_prty\_ref\_id = ?")

End If

Dim parameters As New List(Of DbParameter)

parameters.AddParameter(DbType.Int32, statusCode)

parameters.AddParameter(DbType.Int32, ServicesCommonConstants.PaperBillQueue)

parameters.AddParameter(DbType.Date, fillStartDate)

parameters.AddParameter(DbType.Date, fillEndDate)

parameters.AddParameter(DbType.Int32, carrierId)

parameters.AddParameter(DbType.Int32, planId)

If Not String.IsNullOrEmpty(cardHolderId) Then

parameters.AddParameter(DbType.String, cardHolderId)

End If

Using dr As IDataReader = ExecuteReader(DatabaseEnum.rx21c, siteId, strSQL.ToString(), parameters)

While dr.Read()

Dim paperBillOrder As New PaperBillOrdersBE

paperBillOrder.OrderId = dr.GetValueOrDefault(Of Int32)("order\_id")

paperBillOrder.OrderSeqNumber = dr.GetValueOrDefault(Of Int16)("order\_seq\_nbr")

paperBillOrder.FillDate = dr.GetValueOrDefault(Of DateTime)("fill\_Date")

paperBillOrder.RxFillId = dr.GetValueOrDefault(Of Int32)("rx\_fill\_id")

paperBillOrder.RxId = dr.GetValueOrDefault(Of Int32)("rx\_id")

paperBillOrderList.Add(paperBillOrder)

End While

End Using

If paperBillOrderList.Count > 0 Then

Return paperBillOrderList

Else

Return Nothing

End If

End Using

End Function

Public Sub SendOrderToTP(ByVal siteId As Integer, ByVal rxOrderDetailInfo As RxOrderDetailInfoBE) Implements IOrderDAL.SendOrderToTP

'MISSING ARGUMENTS ARE ADDED FOR THE FUNCTION AND ALSO BEEPERID WHICH IS NOT IN THE TABLE

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

Dim paramList As New List(Of DbParameter)

paramList = BuildGeneralParamList(rxOrderDetailInfo.OrderId,

rxOrderDetailInfo.OrderSeqNbr, paramList)

Dim paramrxid As DbParameter = New Odbc.OdbcParameter

paramrxid.DbType = DbType.Int32

paramrxid.Value = rxOrderDetailInfo.RxId

paramList.Add(paramrxid)

Dim paramFillId As DbParameter = New Odbc.OdbcParameter

paramFillId.DbType = DbType.Int32

paramFillId.Value = rxOrderDetailInfo.RxFillId

paramList.Add(paramFillId)

Dim paramWorkQueCode As DbParameter = New Odbc.OdbcParameter

paramWorkQueCode.DbType = DbType.Int32

paramWorkQueCode.Value = rxOrderDetailInfo.NextWorkQueueCode

paramList.Add(paramWorkQueCode)

Dim paramworkQueStatCode As DbParameter = New Odbc.OdbcParameter

paramworkQueStatCode.DbType = DbType.Int32

paramworkQueStatCode.Value = rxOrderDetailInfo.WorkQueStateCode

paramList.Add(paramworkQueStatCode)

Dim paramDurInd As DbParameter = New Odbc.OdbcParameter

paramDurInd.DbType = DbType.String

paramDurInd.Value = rxOrderDetailInfo.NeedDurInd

paramList.Add(paramDurInd)

ExecuteNonQuery(DatabaseEnum.rx21c, siteId, OrderQueries.SendOrderToTP, paramList)

End Using

End Sub