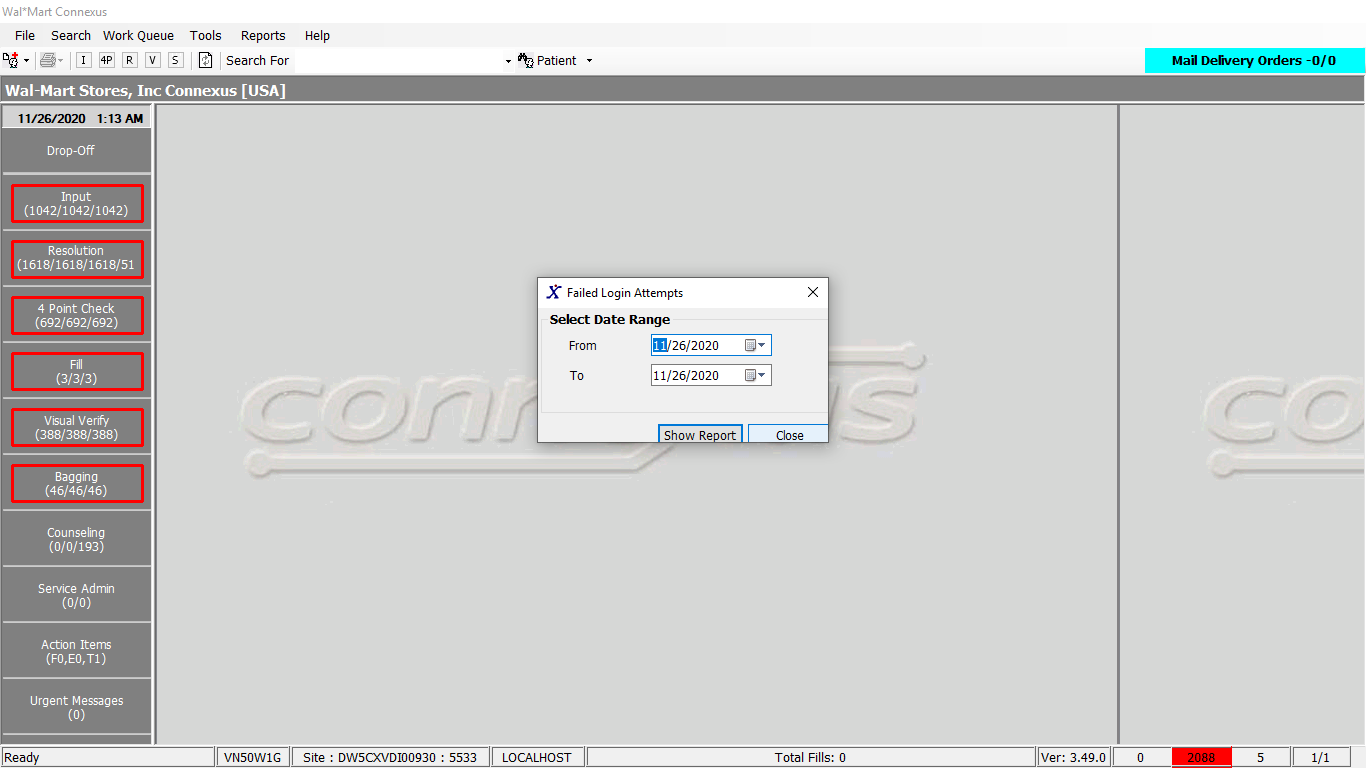
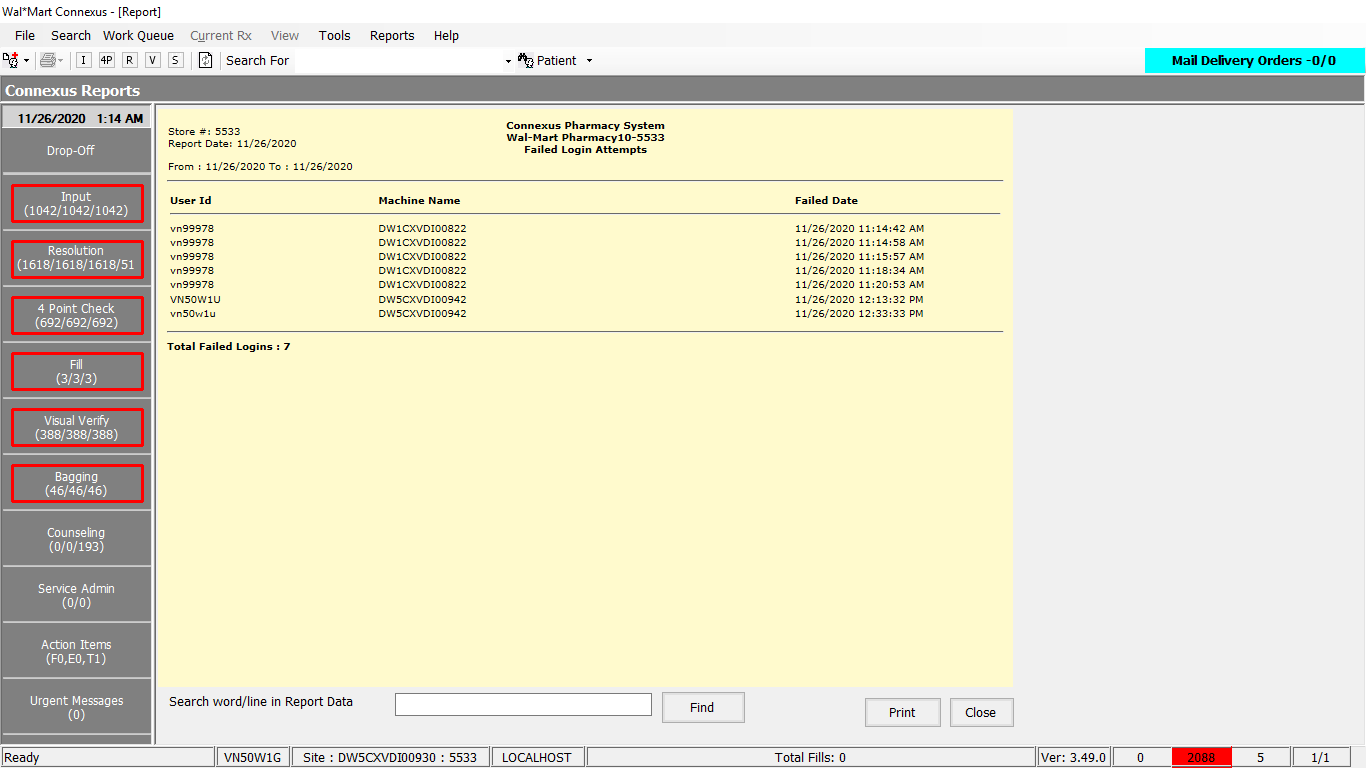
## Reports/ Audit / Failed Login Attempts:

Request:



Response:



Queries:

1.Public Const GetFailedLogOnData As String = " SELECT activity\_ts,activity\_comment,computer\_name " &

" FROM appl\_activity " &

" WHERE DATE(activity\_ts) BETWEEN ? AND ? " &

" AND activity\_type\_code = ?" &

" AND process\_id = ? " &

" AND task\_id = ? " &

" AND activity\_value <> ? " &

" ORDER BY activity\_ts"

2. Public Const GetDateTime As String = "Select CURRENT dtmCurrent FROM systables where tabname = 'systables'"

DAL:

Public Function GetFailedLogOnData(ByVal siteId As Integer, ByVal startDate As Date, ByVal endDate As Date) As List(Of GetFailedLoginInfoBE) Implements IReportDAL.GetFailedLogOnData

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

Dim info As List(Of GetFailedLoginInfoBE) = Nothing

Dim AuditSerializer As Xml.Serialization.XmlSerializer = CacheWrapperDal.AuditSerializer

Dim auditInfo As List(Of Audit.AuditEntry) = Nothing

Dim reader As IO.TextReader = Nothing

Dim parameterlist As New List(Of DbParameter)

parameterlist.Add(ParameterHelper.BuildParameter("ActivityDate", DbType.Date, startDate))

parameterlist.Add(ParameterHelper.BuildParameter("ActivityDate", DbType.Date, endDate))

parameterlist.Add(ParameterHelper.BuildParameter("ActivityTypeCode", DbType.Int32, ServicesCommonConstants.LoginActivityTypeCode))

parameterlist.Add(ParameterHelper.BuildParameter("ProcessId", DbType.Int32, ServicesCommonConstants.AuditConnexusProcessID))

parameterlist.Add(ParameterHelper.BuildParameter("TaskId", DbType.Int32, ServicesCommonConstants.AuditConnexusTaskID))

parameterlist.Add(ParameterHelper.BuildParameter("ActivityValue", DbType.Int32, ServicesCommonConstants.LoginSuccessfulActivityValue))

Using dr As IDataReader = ExecuteReader(DatabaseEnum.Error\_And\_Audit, siteId, ReportQueries.GetFailedLogOnData, parameterlist)

While dr.Read

Dim item As New GetFailedLoginInfoBE

If Not dr.IsDBNull(dr.GetOrdinal("activity\_ts")) Then

item.LastChangeTs = dr.GetDateTime(dr.GetOrdinal("activity\_ts"))

End If

If Not dr.IsDBNull(dr.GetOrdinal("activity\_comment")) Then

reader = New IO.StringReader(dr.GetString(dr.GetOrdinal("activity\_comment")))

auditInfo = CType(AuditSerializer.Deserialize(reader), List(Of Audit.AuditEntry))

Dim userInfo As Audit.AuditEntry = auditInfo.Find(Function(data As Audit.AuditEntry) \_

data.Name.Equals(ServicesCommonConstants.UserName,

StringComparison.InvariantCultureIgnoreCase))

If userInfo IsNot Nothing Then

item.UserId = userInfo.Value

End If

End If

If Not dr.IsDBNull(dr.GetOrdinal("computer\_name")) Then

item.MachineName = dr.GetString(dr.GetOrdinal("computer\_name"))

End If

If info Is Nothing Then

info = New List(Of GetFailedLoginInfoBE)

End If

info.Add(item)

End While

End Using

Return info

End Using

End Function

Public Shared ReadOnly Property GetDBDateTime(ByVal siteId As Integer) As DateTime

Get

Using (New Tracer(ConnexusCommonConstants.ConnexusTracer))

Dim dbDateTime As DateTime

Dim offset As TimeSpan

Dim isOffsetValid As Boolean

'Get the offset time from cache

isOffsetValid = TimeSpan.TryParse(Convert.ToString(CnxCache(CStr(CacheKeys.OffsetTime))), offset)

If (Not isOffsetValid) Then

'get it from db

'execute query

dbDateTime = CDate(ExecuteScalar(DatabaseEnum.rx21c, siteId, BaseQueries.GetDateTime))

offset = dbDateTime.Subtract(DateTime.Now)

'Add it to Cache

CnxCache.Add(CStr(CacheKeys.OffsetTime), offset, CacheHelper.GetExpirationPolicy(CStr(CacheKeys.OffsetTime)))

End If

Return DateTime.Now.Add(offset)

End Using

End Get

End Property