



Real-time Stats Exporter Specification

National Football League

Revised: 9/30/2009

Document Identification

Title: Real-time Stats Exporter
Subject: Real-time Stats Exporter Specification
Version: 1.2
Date: 30-Sep-09
Author: John Cave

Document Revision History

Version	Date	Revised By	Comments
1.0	4/18/02	John Cave	Draft document created.
1.1	8/1/08	John Cave	Added sample code for Visual Basic .NET
1.2	9/30/09	John Cave	Replaced sample code for Visual Basic .NET

TABLE OF CONTENTS

1	OVERVIEW	4
2	LIMITATIONS ON USE	4
3	HIGH LEVEL DESIGN	5
3.1	DATAFLOW DIAGRAM.....	5
4	NETWORKING REQUIREMENTS	6
5	TRANSPORT SPECIFICATION	6
5.1	FILE TRANSFER PROTOCOL (FTP).....	6
5.2	MICROSOFT MESSAGE QUEUE (MSMQ)	6
5.2.1	Sample Program.....	7
6	DATA SPECIFICATION	14
6.1	CUMULATIVE STATISTICS XML	14
6.2	CUMULATIVE STATISTICS CSV	15
6.3	GAME ROSTER.....	15
6.4	SEASON-TO-DATE STATS (HOME AND VISITING TEAMS).....	16
6.5	TEAM RANKINGS	16
6.6	INDIVIDUAL LEADERS	16
6.7	CURRENT WEEK OUT-OF-TOWN SCORES	16
6.8	CURRENT WEEK LEADERS	16
7	OPERATION	17
7.1	STARTING REAL-TIME STATS EXPORTER.....	17

1 Overview

The GSIS (Game Statistics and Information System) application running at all NFL stadiums has the ability to provide a real-time feed of statistics to in-stadium computer systems controlling such devices as matrix boards, character generators, kiosks etc. GSIS can provide the following information:

- Game Rosters
- Local game statistics including cumulative game-to-date statistics as well as individual play statistics
- Season-to-date cumulative statistics for the home and visiting teams
- Out-of-town NFL scores for the current week including game leaders in rushing, passing and receiving.
- Current week leaders in rushing, passing, receiving, defense and kicking.

2 Limitations On Use

Information made available via the GSIS Real-time Stats Exporter is limited to usage within the property boundaries of the stadium. Any redistribution of this information outside of the property of the stadium without the express written permission by the National Football League is prohibited.

3 High Level Design

3.1 Dataflow Diagram

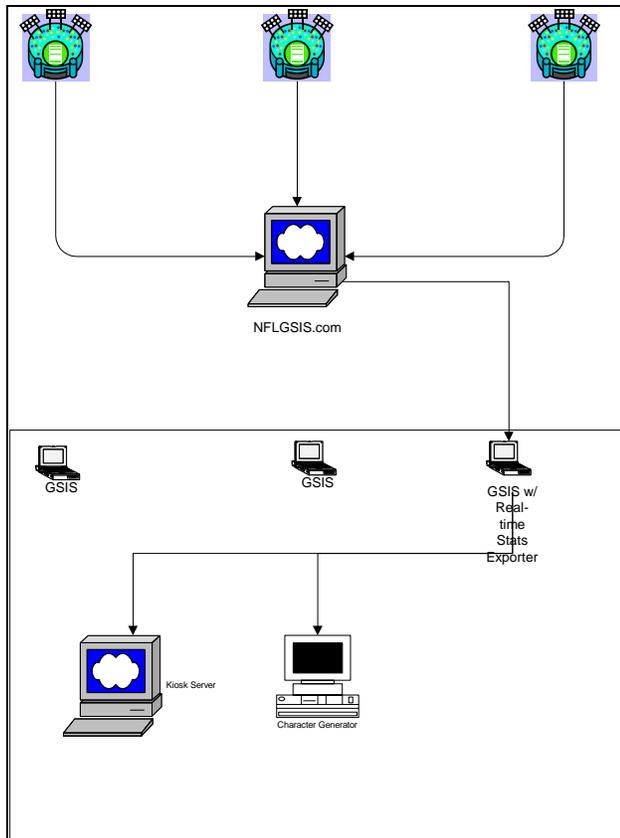


Figure 1

4 Networking Requirements

Any computer receiving a statistics feed from GSIS must be connected to the “.17” virtual LAN segment of the GSIS Switch. There are four ports (21-24) on the switch that reside on the “.17” virtual LAN segment. Only three of those ports are available for connection as one port is used to connect the VLAN to the router.

The IP address of a client computer must be set to 172.17.xxx.yyy where xxx is the 3 IP address octet used in the stadium and yyy is the unique address assigned to your computer. Use the following matrix to assign an IP address to your computer:

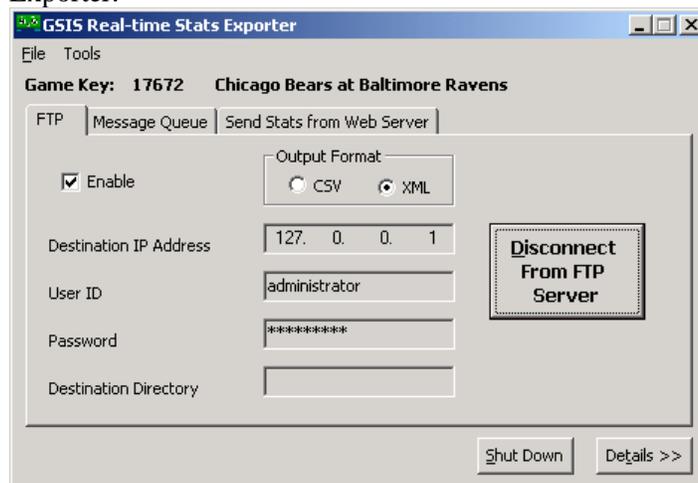
Client Computer	IP Address Range
Scoreboard and Matrix Boards	100 - 109
Broadcast Truck	110 - 119
In-Stadium Kiosk Server	120 - 129

5 Transport Specification

5.1 File Transfer Protocol (FTP)

When using FTP as the transport mechanism there can a maximum of 1 recipient. When an update is available for sending, the program will perform a PUT operation of the file onto the destination FTP server. The FTP mechanism is supported for legacy applications only because it is limited to a single destination. Any newly developed applications should use the Microsoft Message Queue mechanism.

Here’s an example screen shot of the *FTP* configuration tab of the GSIS Real-time Stats Exporter:



5.2 Microsoft Message Queue (MSMQ)

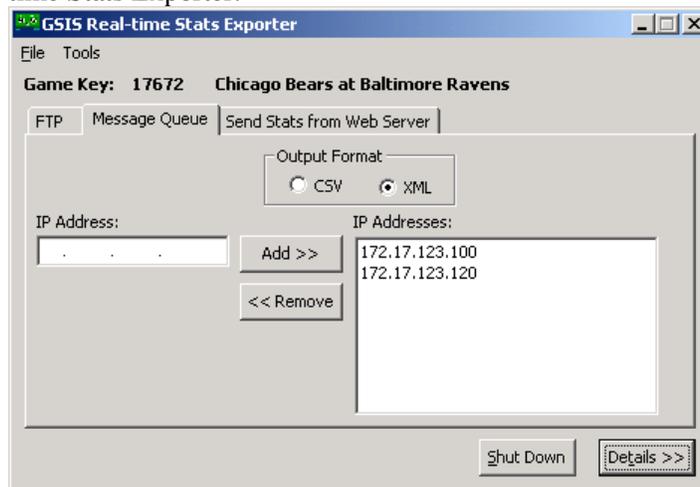
Message queuing is the recommended transport mechanism for use with the GSIS Real-time Stats Exporter. Message queuing allows the real-time stats exporter to send to

multiple destinations without having to be concerned whether or not each destination received the file. The message queuing subsystem will be responsible for ensuring the proper delivery of the message.

In order to receive stats via Microsoft Message Queuing you will need to have Windows 2000 service pack 2 or greater with Microsoft Message Queue installed and configured.

The GSIS Real-time Stats Exporter always sends messages to each of the IP addresses in the list. The Label property of the message is set to the filename (i.e. GSISGameStats.xml, BLT_SeasonToDateStats.xml etc.). The Body property of the message is set to the contents of the file.

Here's an example screen shot of the *Message Queue* configuration tab of the GSIS Real-time Stats Exporter:



5.2.1 Sample Program (Visual Basic 6.0)

The following sample program was written in Visual Basic 6.0. It will receive messages from the Real-time Stats Exporter and write the messages to disk as a file. This code sample illustrates how you may receive and store Microsoft Message Queue messages. It does not process the content of the messages. You are free to use the source code in this document as necessary.

```

'*****
'* This project references the Microsoft Message Queue 2.0 Object
Library
'* MQOA.dll
'*****

Option Explicit

Dim oQueue As MSMQ.MSMQQueue
Dim WithEvents oEvent As MSMQ.MSMQEvent
Dim oQueueInfo As MSMQ.MSMQQueueInfo
Dim oTransDisp As MSMQ.MSMQTransactionDispenser
Dim oTrans As MSMQ.MSMQTransaction

Private Type SECURITY_ATTRIBUTES

```

```
        nLength As Long
        lpSecurityDescriptor As Long
        bInheritHandle As Long
End Type

Private Declare Function CreateFile Lib "kernel32" _
    Alias "CreateFileA" (ByVal lpFileName As String, _
        ByVal dwDesiredAccess As Long, ByVal dwShareMode As Long, _
        lpSecurityAttributes As SECURITY_ATTRIBUTES, _
        ByVal dwCreationDisposition As Long, _
        ByVal dwFlagsAndAttributes As Long, ByVal hTemplateFile As Long) _
    As Long

Private Declare Function WriteFile Lib "kernel32" (ByVal hFile As Long,
    lpBuffer As Any, ByVal nNumberOfBytesToWrite As Long,
    lpNumberOfBytesWritten As Long, ByVal lpOverlapped As Long) As Long
Private Declare Function FlushFileBuffers Lib "kernel32" (ByVal hFile As
    Long) As Long
Private Declare Function CloseHandle Lib "kernel32" (ByVal hObject As
    Long) As Long
Private Const GENERIC_WRITE = &H40000000
Private Const GENERIC_READ = &H80000000
Private Const OPEN_ALWAYS = 4
Private Const CREATE_ALWAYS = 2
Private Const INVALID_HANDLE_VALUE = -1
Private Const FILE_ATTRIBUTE_NORMAL = &H80

Private Sub Form_Load()
    Set oEvent = CreateObject("MSMQ.MSMQEvent")
    Set oQueueInfo = CreateObject("MSMQ.MSMQQueueInfo")
    Set oTransDisp = CreateObject("MSMQ.MSMQTransactionDispenser")

    'Set the queue name information
    oQueueInfo.PathName = ".\Private$\GSIS Realtime Stats Exporter"
    oQueueInfo.Label = "GSIS Real-time Stats Exporter"

    'Create the queue if it doesn't already exist
    On Error Resume Next
    oQueueInfo.Create True
    On Error GoTo ErrorHandling

    'Open the queue for receiving
    Set oQueue = oQueueInfo.Open(MQ_RECEIVE_ACCESS, MQ_DENY_NONE)

    'Hang an asynchronous read on the queue
    oQueue.EnableNotification oEvent

Exit Sub

ErrorHandling:
    MsgBox Err.Description
    Unload Me

End Sub

Private Sub oEvent_Arrived(ByVal Queue As Object, ByVal Cursor As Long)
    On Error GoTo ErrorHandling
    Dim objQueue As MSMQ.MSMQQueue
    Dim oMessage As MSMQ.MSMQMessage

    'Make the Queue object early bound
    Set objQueue = Queue

    'If we're not already in a Transaction then Start an MSMQ Transaction
    If oTrans Is Nothing Then
        Set oTrans = oTransDisp.BeginTransaction
    End If
```

```

'Receive the message
Set oMessage = objQueue.Receive(oTrans)

WriteToFile "c:\Temp\" & oMessage.Label, oMessage.Body,
oMessage.BodyLength, oMessage.Label

'Commit the transaction
oTrans.Commit
Set oTrans = Nothing

'Hang an asynchronous read on the queue
oQueue.EnableNotification oEvent

Exit Sub

ErrorHandling:
MsgBox Err.Description

End Sub

Private Sub WriteToFile(FileName As String, FileContents() As Byte,
FileContentsLength As Long, ShortFileName As String)
On Error GoTo ErrorHandling
Dim hFile As Long
Dim lNumberOfBytesWritten As Long
Dim lSuccess As Long
Dim lBodyLength As Long
Dim lDiskFileSize As Long
Dim lLastError As Long
Dim lSecurity As SECURITY_ATTRIBUTES
Dim lDiskSpaceRequired As Long

hFile = CreateFile(FileName, GENERIC_WRITE Or GENERIC_READ, 0,
Security, CREATE_ALWAYS, FILE_ATTRIBUTE_NORMAL, 0)

If hFile = INVALID_HANDLE_VALUE Then
Err.Raise vbObjectError + 4, , "Create file failed"
End If

lSuccess = WriteFile(hFile, FileContents(LBound(FileContents)),
FileContentsLength, lNumberOfBytesWritten, 0)
If lSuccess Then
'Flush the file buffers to force writing of the data.
lSuccess = FlushFileBuffers(hFile)
Else
lSuccess = CloseHandle(hFile)
Err.Raise vbObjectError + 5, , "WriteFile failed - Last Error = "
& Err.LastDllError
Exit Sub
End If

'Close the file.
lSuccess = CloseHandle(hFile)

Exit Sub

ErrorHandling:
Err.Raise Err.Number, Err.Description
End Sub

```

5.2.2 Sample Program (Visual Basic .NET 2005)

`Imports System.Messaging`

```
Imports System.Xml

Public Class frmMain
    Private Const QUEUE_NAME As String = "GSIS Realtime Stats Exporter"
    Public gGSISRealTimeStatsReceiverThread As System.Threading.Thread
    Public gGamekey As Long
    Private gFilePath As String = "C:\Temp"

    Private gShutdown As Boolean
    Private gThreadStarted As Boolean
    Private gThreadExited As Boolean

    Delegate Sub DelegateUpdateStatus(ByVal sLabel As String, ByVal sStatus
As String)

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        'Create the Queue that the GSIS Real Time Stats Exporter will use to
send stats data to us
        Try
            CreateQueue(QUEUE_NAME, "GSIS Real-time Stats Exporter", True)
        Catch exc As Exception
            MsgBox("Error creating 'GSIS Real-time Stats Exporter' Queue.
Automatic Import of data from GSIS will not work.", MsgBoxStyle.Critical,
"Create Message Queue Error")
            Exit Sub
        End Try

        'Start a thread that will continuously loop reading messages from the
queue
        gGSISRealTimeStatsReceiverThread = New
System.Threading.Thread(AddressOf ReceiveMessagesThread)
        gGSISRealTimeStatsReceiverThread.Name = "ReceiveMessages"
        gGSISRealTimeStatsReceiverThread.Start()
        gThreadStarted = True

    End Sub

    Friend Shared Sub CreateQueue(ByVal QueueName As String, ByVal QueueLabel
As String, ByVal IsTransactional As Boolean)
        'Create the queue on this machine
        Dim oQueue As New System.Messaging.MessageQueue
        Dim sPathName As String

        sPathName = ".\Private$\\" & QueueName
        If Not oQueue.Exists(sPathName) Then
            oQueue = oQueue.Create(sPathName, True)
            oQueue.Label = QueueLabel
            oQueue.Formatter = New BinaryMessageFormatter
        End If
    End Sub

    Private Sub UpdateStatus(ByVal sLabel As String, ByVal sFilename As
String)
```

```
    Try
    If (lvwStatus.InvokeRequired) Then
        Dim oDelegate As New DelegateUpdateStatus(AddressOf
UpdateStatus)
        Me.Invoke(oDelegate, New Object() {sLabel, sFilename})
    Else
        Dim oItem As ListViewItem = lvwStatus.Items.Add(Now)
        oItem.SubItems.Add(sFilename)
        oItem.EnsureVisible()

        Select Case UCase(sLabel)
            Case "SCORES.XML"
                ProcessScores(sFilename)
            Case "GSISGAMESTATS.XML"
                ProcessCurrentGame(sFilename)
        End Select
    End If
    Catch ex As Exception
        MessageBox.Show(ex.Message)
    End Try
End Sub

Private Sub ProcessScores(ByRef sFilename As String)
    Dim oDoc As New XmlDocument
    Dim oNodeList As XmlNodeList
    Dim oNode As XmlNode
    Dim oItem As ListViewItem

    Try
        oDoc.Load(sFilename)
    Catch ex As Exception
        MsgBox("Failed to load " & sFilename & ": " & ex.Message)
    End Try

    lvwScores.Items.Clear()
    oNodeList = oDoc.SelectNodes("Scoreboard/Game")
    For Each oNode In oNodeList
        oItem =
lvwScores.Items.Add(oNode.Attributes("VisitorClubName").Value)
        oItem.SubItems.Add(oNode.Attributes("VisitorTeamScore").Value)
        oItem.SubItems.Add(oNode.Attributes("HomeClubName").Value)
        oItem.SubItems.Add(oNode.Attributes("HomeTeamScore").Value)

    Next

End Sub

Private Sub ProcessCurrentGame(ByRef sFilename As String)
    Dim oDoc As New XmlDocument
    Dim oNodeList As XmlNodeList
    Dim oNode As XmlNode

    Try
```

```
        oDoc.Load(sFilename)
    Catch ex As Exception
        MsgBox("Failed to load " & sFilename & ": " & ex.Message)
    End Try

    'Get the scores
    oNodeList = oDoc.SelectNodes("CumulativeStatisticsFile/Score")
    lblVisitor.Text = oNodeList(0).Attributes("VisitingTeam").Value
    lblHome.Text = oNodeList(0).Attributes("HomeTeam").Value
    lblVisitorScore.Text = oNodeList(0).Attributes("VisitorScore").Value
    lblHomeScore.Text = oNodeList(0).Attributes("HomeScore").Value

    'Get the quarter, clock time and situation
    oNodeList =
oDoc.SelectNodes("CumulativeStatisticsFile/CumeStatHeader")
    If IsNumeric(oNodeList(0).Attributes("Phase").Value) Then
        lblQtrTime.Text = "Q" & oNodeList(0).Attributes("Phase").Value &
" " & oNodeList(0).Attributes("GameClock").Value
    Else
        lblQtrTime.Text = oNodeList(0).Attributes("Phase").Value
    End If

    If UCase(oNodeList(0).Attributes("Phase").Value) <> "F" Then
        lblSituation.Text =
Replace(oNodeList(0).Attributes("PossessionTeam").Value, " ", "") & " " &
Replace(oNodeList(0).Attributes("Down").Value, " ", "") & "-" &
oNodeList(0).Attributes("Distance").Value & " " &
oNodeList(0).Attributes("YardLine").Value
    End If

    'Get the last play description
    oNodeList = oDoc.SelectNodes("CumulativeStatisticsFile/Play")
    txtPlayDescription.Text = oNodeList(oNodeList.Count -
1).Attributes("PlayDescription").Value

End Sub

Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    gShutdown = True

    If gThreadStarted Then
        While Not gThreadExited
            System.Threading.Thread.Sleep(100)
        End While
    End If
End Sub

Public Sub ReceiveMessagesThread()
    Dim oTrans As New MessageQueueTransaction
    Dim oMessage As System.Messaging.Message
    Dim oReceiveQueue As MessageQueue
    Dim sFilename As String
```

```
Try
    'Open the Queue
    oReceiveQueue = New MessageQueue(".\Private$" & QUEUE_NAME)
    oReceiveQueue.Formatter = New BinaryMessageFormatter
Catch exc As Exception
    MsgBox("Error opening 'GSIS Real-time Stats Exporter' Queue.
Automatic Import of data from GSIS will not work.", MsgBoxStyle.Critical,
"Open Message Queue Error")
    ' Add entry in error log
    Exit Sub
End Try

Do 'This will loop until instructed to shutdown
    'Begin an MQ transaction
    oTrans.Begin()
    Do 'This will loop until a message is received or shutdown
        Try
            oMessage = oReceiveQueue.Receive(New TimeSpan(0, 0, 1),
oTrans)
            Exit Do
        Catch e As MessageQueueException
            If gShutdown Then
                Exit Do
            End If
        End Try
    Loop

    If gShutdown Then Exit Do

    'Process the received message here
    'In the case of this example we will just write the file to disk
and
    'raise an event so the main form can be updated.
    sFilename = gFilePath & "\" & oMessage.Label
    Dim oFileStream As New System.IO.FileStream(sFilename,
IO.FileMode.Create)

    Dim oWriter As New System.IO.BinaryWriter(oFileStream)

    Dim brReader As New IO.BinaryReader(oMessage.BodyStream)
    Dim docByteArray() As Byte

    docByteArray = brReader.ReadBytes(oMessage.BodyStream.Length)

    oWriter.Write(docByteArray, 0, docByteArray.Length)
    oWriter.Close()
    oFileStream.Close()
    oFileStream.Dispose()

    'Commit the transaction
    If oTrans.Status = MessageQueueTransactionStatus.Pending Then
        oTrans.Commit()
    End If
```

```
'Update the status on the main UI thread
UpdateStatus(oMessage.Label, sFilename)

Loop

gThreadExited = True

End Sub

End Class
```

6 Data Specification

6.1 Cumulative Statistics XML

This file is sometimes referred to as the STATXML file. This file is generated by four different events.

GSIS Real-time Stats Exporter Startup

First, when the GSIS Real-time Stats Exporter is started a file will be generated and sent to the destination computers. Generally this is done prior to the start of the game and thus the file will contain all zero stats. However, if the Real-time Stats Exporter is started while the game is in progress all of the cumulative stats may have numbers other than zero. Also, the Play element and the PlayStat element will contain all plays-to-date. **IMPORTANT:** If the receiving program uses these elements then it should be written to test if the number of plays is greater than one. If so, it should delete every play and replace it with the information contained within this file.

Play Entry Completed

A new file is generated when the GSIS operator completes entry of a play. This file will contain a single Play element with corresponding PlayStat elements for this new play only. This is done to minimize the amount of time required to supply the entire list of plays and stats.

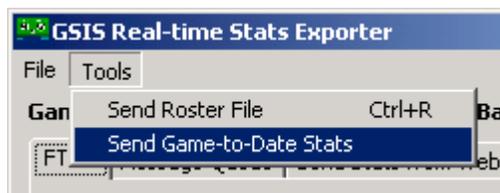
Correction to a Previously Completed Play

A new file will be generated each time the GSIS operator makes a correction to a previously completed play. For example, if the entry operator goes back to correct the *Rushed By* and *Tackled By* attributes of a given play a file will be generated when the *Rushed By* attributed is changed and when the *Tackled By* attributed is changed. Thus generating two additional files. Each file will contain a single Play element with corresponding PlayStat elements for this new play only.

Manually

You may force the generation of a complete game to date stats file to be sent to the receiving computers. This file will be the identical to the file that would be generated if you stopped and restarted the program.

To manually force a generation of the cumulative stats files, choose **Send Game-to-Date Stats** from the **Tools** menu.



FTP File Name	GSISGameStats.xml
MSMQ Message Label	GSISGameStats.xml

For detailed information pertaining to the contents of this file, please refer to the document titled: **StatXMLRptDocumentation**. You may find this file at:
<http://www.nflgsis.com/gsis/documentation/partners/StatXMLRptDocumentation.pdf>

6.2 Cumulative Statistics CSV

This file is sometimes referred to as the STAT file. This file format is provided for legacy applications. This file is generated by the same four events as the “Cumulative Statistics XML” file.

Please note that the file names are different based upon the transport method you use. When using FTP, the file name is the name of the home team club. This is done for legacy purposes. However, if the transport is Message Queue the file will be named GSISGameStats.csv.

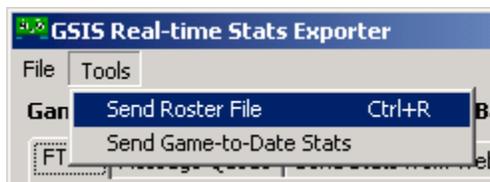
FTP File Name	<home team club name>
MSMQ Message Label	GSISGameStats.csv

For detailed information pertaining to the contents of this file, please refer to the document titled: **StatPBPRptDocumentation**. You may find this file at:
<http://www.nflgsis.com/gsis/documentation/partners/StatPBPRptDocumentation.pdf>

6.3 Game Roster

Prior to the start of the game and as soon as the stats crew has completed verifying the accuracy of the roster, a file containing the game roster information may be sent. The file name will be simply “Roster”.

To manually force a generation of the roster file, choose **Send Roster File** from the **Tools** menu.



For detailed information pertaining to the contents of this file, please refer to the document titled: **GSISRosterRptDocumentation**. You may find this file at:
<http://www.nflgsis.com/gsis/documentation/partners/GSISRosterRptDocumentation.pdf>

6.4 Season-to-Date Stats (Home and Visiting Teams)

This file is updated immediately following the close of a game. For an example of information available click on the "Season Stats" tab on NFLGSIS.com the click on the "Season Statistics Report" button.

For a sample file, please visit <http://www.nflgsis.com/gsis/documentation/StatsExporter>.

6.5 Team Rankings

This file is updated at the completion of each game and daily at 11:00 AM ET. The file contains team specific total stats for Offense, Passing, Receiving, Rushing, etc.

For an example of the data available visit NFLGSIS.com and click on the "Season Stats" tab then click on the "Team" button. Note that you can sort by any column by simply clicking on the column header.

For a sample file, please visit <http://www.nflgsis.com/gsis/documentation/StatsExporter>.

6.6 Individual Leaders

This file is updated at the completion of each game and daily at 11:00 AM ET. The name of the file is a little misleading because it contains more than just the leaders of each stat category. The file contains a record of every player that has a stat in any given category and is sorted by the best in each category. For an example of the data available visit NFLGSIS.com and click on the "Season Stats" tab then click on the "Leaders" button. Note that you can sort by any column by simply clicking on the column header.

For a sample file, please visit <http://www.nflgsis.com/gsis/documentation/StatsExporter>.

6.7 Current Week Out-of-Town Scores

We update this file every 60 seconds at the distribution center. This file contains all of the scoreboard information for every game of a particular week. If you log in to the NFLGSIS web site you will see all of the information for the current week. The web site uses this file to display the scores of all games.

For a sample file, please visit <http://www.nflgsis.com/gsis/documentation/StatsExporter>.

6.8 Current Week Leaders

We update this file every 60 seconds at the distribution center. This file contains the Rushing, Passing, Receiving and Kicking leaders for the current week.

For a sample file, please visit <http://www.nflgsis.com/gsis/documentation/StatsExporter>.

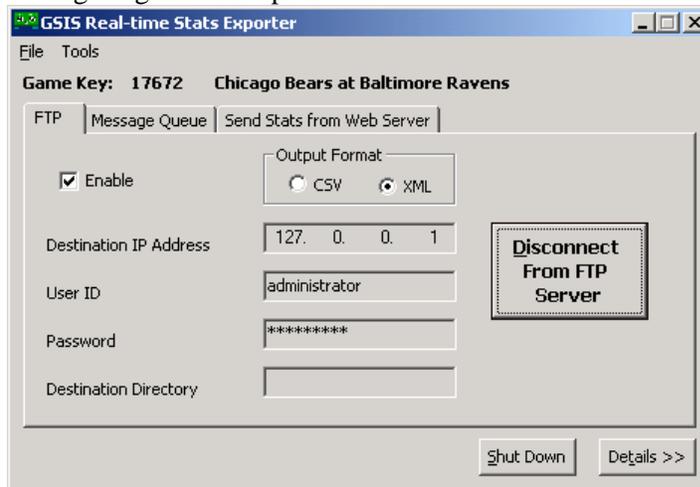
7 Operation

7.1 Starting Real-time Stats Exporter

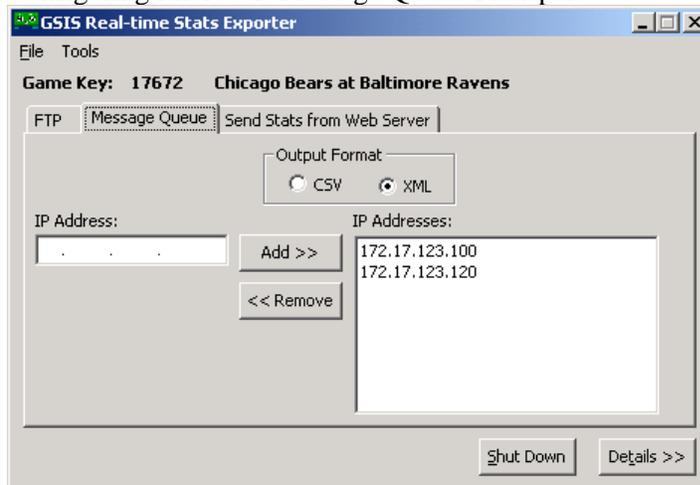
- Simply click **Real-time Stats Exporter** on the **Tools** menu.



- Configuring FTP transport



- Configuring Microsoft Message Queue Transport



- Sending Stats from Web Server

