

HNF-29536-FP  
Revision 0

# **HAMMER Courseware Management System**

## **System Development and Implementation**

Prepared for the U.S. Department of Energy  
Assistant Secretary for Environmental Management

Project Hanford Management Contractor for the  
U.S. Department of Energy under Contract DE-AC06-96RL13200

**FLUOR**<sup>®</sup>  
P.O. Box 1000  
Richland, Washington

**Approved for Public Release;  
Further Dissemination Unlimited**

HNF-29536-FP  
Revision 0

# HAMMER Courseware Management System

## System Development and Implementation

P. R. Gardner  
Fluor Hanford

Date Published  
April 2006

To Be Presented at  
Columbia Basin College Course: CS 260 Data Structures in C++

Columbia Basin College  
Pasco, Washington

May 1, 2006

Prepared for the U.S. Department of Energy  
Assistant Secretary for Environmental Management

Project Hanford Management Contractor for the  
U.S. Department of Energy under Contract DE-AC06-96RL13200

**FLUOR.**  
P.O. Box 1000  
Richland, Washington

**Copyright License**

By acceptance of this article, the publisher and/or recipient acknowledges the U.S. Government's right to retain a nonexclusive, royalty-free license in and to any copyright covering this paper.

 4/28/2006  
Release Approval Date

Approved for Public Release;  
Further Dissemination Unlimited

**LEGAL DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This document is available to the U.S. Department of Energy and its contractors, in paper from the Office of Scientific and Technical Information (OSTI). It is available for sale to the public from the National Technical Information Service (NTIS).

This report has been reproduced from the best available copy. Available in paper copy.

Printed in the United States of America

# **HAMMER Courseware Management System**

## **System Development and Implementation**

**Author:** PR Gardner  
**Date:** April 15, 2006

## HAMMER Courseware Management System System Development and Implementation

### I. Description

HAMMER Courseware Management System (HAMMERCMS) is the official name of the system Fluor Hanford, Inc., uses to

- facilitate development of,
- deliver, and
- track

training presented in some electronic form (mainly, web-based training) to Hanford Site employees, subcontractors, and vendors.

#### *Early Picture (5/9/2001)*

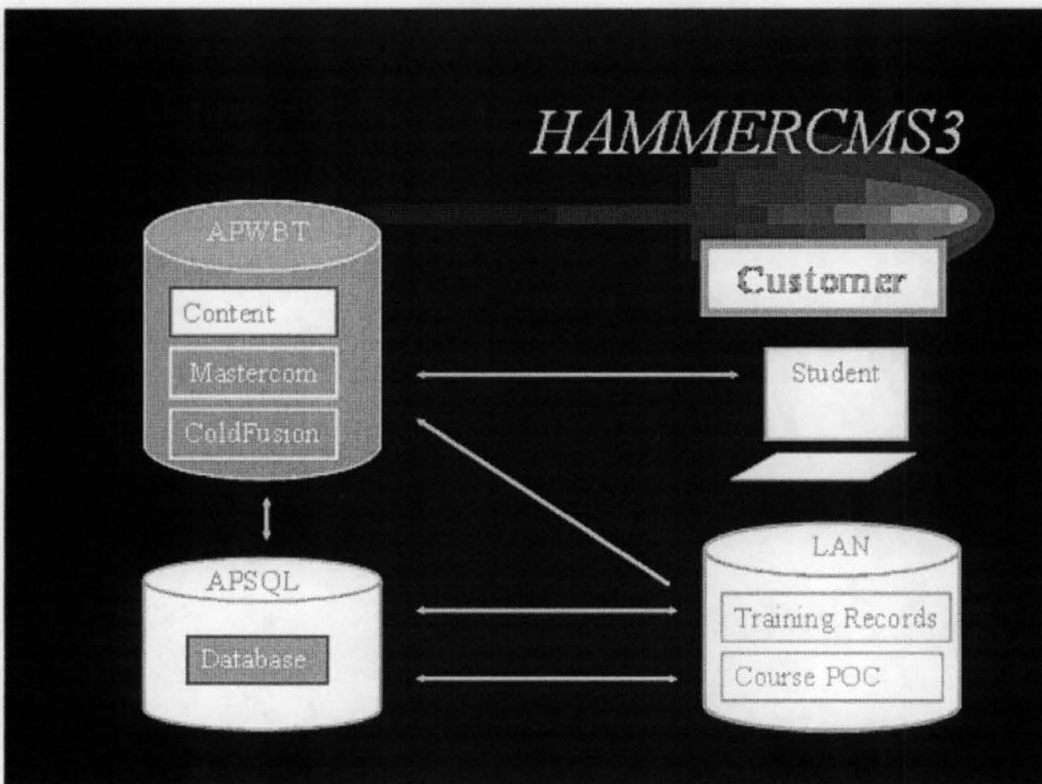


figure 1

This picture is still pretty good for describing the vision and fundamental process! However, the system has grown increasingly complex.

#### *Recent Picture (1/6/2006)*

## HAMMER Courseware Management System System Development and Implementation

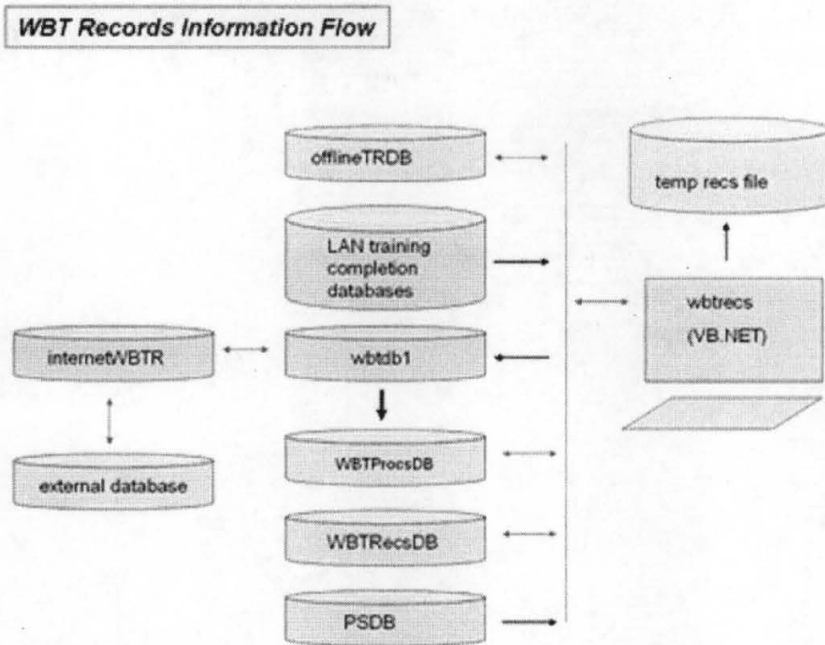


figure 2.

This picture represents the expansion of the bottom row of the first picture; that of communicating between various online training records databases and the official training records database.

## Fluor Hanford Software Development Process

### General

The process is thoroughly described in considerable detail in company procedures and addresses such topics as

- requirements definition
- software acquisition
- custom design and development
- quality assurance practices
  - documentation
  - testing
- customer acceptance
- Production Readiness Review Board (PRRB)
- implementation
- maintenance

The requirements are adjusted somewhat depending on the risk, complexity, and “reach” of the software ( e.g., how significant is a failure, how difficult will the software be to

## HAMMER Courseware Management System System Development and Implementation

maintain, how many people will be affected, what is the probability of affecting other Local Area Network resources?).

### ***Application to HAMMERCMS***

Unfortunately, HAMMERCMS is relatively complex, as might be gleaned from the Recent Picture, above, and affects roughly 10,000 people annually. That means that HAMMER, the organization owning and managing the software, must ensure several steps are followed for the foreseeable future: document, test, monitor, fix, test, present to PRRB, update, present to PRRB, etc. Prior to operating this system for the very first time, we were required to write the following documents:

- Approved Statement of Work
- Project Management Plan
- Requirements Specification
- System Design Description (Table of Contents attached)
- Requirements Traceability Matrix
- Internal/External Interface Agreements
- Test Plan
- Test Cases
- User Acceptance
- Version Design Description
- Y2K Certification
- Findings Resolution
- Configuration Management Plan
- System Security Plan
- Application Risk Assessment

Within Fluor Hanford, HAMMER is somewhat unique for both owning and developing a system used by multiple site contractors. Historically, this situation arose because (a) the organization had experience in developing and delivering computer-based training requiring specialized hardware and software and (b) when web-based training was relatively new (1990s), our first two attempts at obtaining a commercial web-based training system failed.

Note that much of the required documentation assumes a customer – supplier relationship. In this case, the “customer” was the supplier’s manager so that

- the “approved statement of work” consisted of the direction, “Put General Employee Training” on the web,
- the project management plan was written approximately 80% of the way into the project in preparation for the Production Readiness Review Board presentation,
- the requirements specification was written by the system designer/engineer/programmer to meet the needs of the site,

## HAMMER Courseware Management System System Development and Implementation

- the internal/external interface agreements simply documented understandings reached by members of the same organization on how to pass data.

Writing down the requirements and a plan to test those requirements is also beneficial. Whenever performing a change, it helps enormously to return to the original test plan, modify it, and run the updated tests. In one case, because we had a thorough test plan that the system had passed, we were able to argue that a persistent problem was caused by a driver supplied by our hardware supplier, not by our program. (The hardware supplier ultimately changed their driver.)

### Coding Considerations

Ideally, system designers select the languages in which to write various system processes based on the functional requirements for those processes. In HAMMERCMS, some of the code exists solely because, when the very first web-based training courses were written, circa 1996, there were few options and little expertise available. Information that now might be kept in a session variable, HAMMERCMS keeps in a cookie. A very early version of the system kept, solely in document.cookie, for the General Employee Training course, about 25 lesson completion records with the student's responses to about 10 questions per lesson. Because there was no real opportunity for an employee to start the course on one machine and finish on another, we advised them to complete the course all in one sitting. On completion, JavaScript opened a form containing the completion results and instructed the user to email that form to a Training Records mailbox. Since that time, the system has matured.

Here is a table of some of functions the system is required to perform together with the languages selected to implement those functions.



## HAMMER Courseware Management System System Development and Implementation

<b>Function</b>	<b>Code Considerations</b>
Present a page of static information in a web browser	HTML.
Present a “flashier” page of information; perhaps including video and audio.	HTML and JavaScript; more and more in Flash.
React to student input to questions	Javascript to check for input validity; ColdFusion to record response in a SQL database.
Display database information for reports	For screen display: SQL, ColdFusion for data retrieval; HTML, some JavaScript. For text reports for training administrators: SQL, VB6, VB.Net.
Retrieve information from multiple databases (SLQ and Access) for entry into an official Training Records system.	VB.Net.
Enable web-based training developers to write and manage their own courses.	HTML, Javascript, Flash templates for content; HTML, JavaScript, SQL, ColdFusion for database interface.

### **A Code Example: wbtrecs.exe**

wbtrecs.exe (not its real name) is the VB.Net program operated by the Training Records staff to retrieve training completion records from the online training records databases, then format those records for input into the official Training Records database. This is the program identified in figure 2, above. Attached are snippets from that code, some routines of which may be useful in other contexts. These snippets highlight a number of interesting features.

#### ***Feature 1. A Configuration File***

Configuration files are widely prevalent so why is this even a “feature?”

## HAMMER Courseware Management System System Development and Implementation

1. Exactly one organization runs this program on exactly one computer. In this context, why even bother with a configuration file? Two reasons: one organization runs this program on one computer in production; the developer gets to test the program, sometimes using live data, sometimes using test data, sometimes using some combination. For example, if a member of the Training Records staff reports that a record failed to come over from externalTR, then the programmer can modify the test version of the configuration file on the developer's machine until the problem is resolved. Note in particular the multiple warnings to check the configuration file.

2. The file happens to be a pure text file even though VB.Net introduced an XML method for configuration files. Experimenting with the suggested technique drove me to fall back to the text file; choosing simplicity over sophistication (which I have done many times, and for which I've been glad). XML may be useful for computer-computer communication but there seemed no advantage in this case.

### ***Feature 2. Continuing Evolution***

Apparent in the comments; this has proved useful in attempting to diagnose problems.

### ***Feature 3. Black Box Functions***

VB.Net, also being an evolutionary product, has multiple ways of performing routine functions. After considerable experimentation and approaching deadlines, I once again resorted to simplicity over sophistication. There are multiple ways to communicate with a database, for example. In the end, I wrote the functions I always use, such as

```
RetrieveData(thisDBName, thisQuery, queryName)
```

that mimic the simple syntax used in ColdFusion. Looking at the specific functions, it is easy to see they do NOT work efficiently with the database (e.g., connections are opened and dropped multiple times; internal tables are dropped and reconstructed). However, I never have to debug them again and they work much faster than required for actual training records processing. Similar considerations apply to file I/O. The SaveTextAs function, for example, simply stores everything in RAM until it is ready to write. With hardware advances, I anticipate the program will never have a problem with internal storage.

### ***Feature 4. Extra Reports***

One of the saving design features is the ProcessReport that writes out, each time the process runs, what happened at various stages. See the attached example. This is an invaluable aid in locating and resolving exceptions. For example,

## HAMMER Courseware Management System System Development and Implementation

### Process Flow

The process normally flows as follows:

1. At 4:00 pm, XP automatically launches the program.
2. The program gets information from the configuration file.
3. For each database, the program gets the time records were last officially retrieved.
4. Setting the officialToTime to the current time minus 30 minutes, the program retrieves all records, from each database, for which completions were recorded from the last official time to the new official time.
5. The program performs several validity checks on the data including
  - a. removing duplicate records,
  - b. checking for people already known to Training Records,
  - c. checking for valid charge codes.
6. The program writes incoming information, together with any exception reports, to databases and to text files.

### A Bug

On February 9, it was reported that PersonA completed course 100099 on the internet but failed to receive credit in PSDB. See [wbtrecs\\_02082006\\_problem.doc](#).

### Exercise

Use the attached information to locate and identify the needed correction.

### Solution

See [wbtrecs\\_02082006\\_fix.doc](#).

## A Final Reflection

Much (all?) of the complexity of HAMMERCMS is due to the multitude and variety of interfaces, both human and machine. Programmers need to stay close, and be open to their human users, an exhortation required because it is so much easier to stay close to the machine.

That's enough!

# **HAMMER Courseware Management System**

## **Attachment 1.**

### **System Design Description**

#### **Table of Contents**

## HAMMERCMS System Design Description Table of Contents

I.	Background	3
II.	System Functional Overview	3
	A. Functions	3
	B. Content Delivery	4
	C. Course Performance Data	4
	D. Developer's Interface	4
	E. Training Completion Records	4
III.	System Infrastructure	4
	A. Web-Based Training Course Object Model	5
	B. Code Languages and Systems	6
	C. Associated Databases	6
	D. Mastercom	9
	1. Login component	9
	2. Course management component	10
	E. Records Administration	12
IV.	System Operation	16
	A. User Categories	17
	1. Students	17
	3. Student Administrators	17
	4. Course Administrators (Points-of-Contact)	18
	5. Developers (Owners)	18
	6. Reviewers	18
	7. Training Authenticators	19
	8. System Administrators	19
	B. Courseware Delivery	19
	1. Curricula	19
	2. Student Operation	20
	3. Data Flow	22
	C. Training Administration	23
	D. Administration of Offline Training	25
	1. Overview	25
	2. Authorized HAMMERCMS Offline Courses	25
	3. Auth HAMMERCMS Offline Course Authenticators	26
	4. Authenticator Access	26
	5. Training Completion Form - Authenticators	27
	6. Training Completion Form – Records Administrators	32
V.	HAMMERCMS DevNumber2	33
VII.	Special Processes	34
	A. Topic Knowledge Check Process	34
	1. Overview	34
	2. Entity Relationships	34
	3. Operation	34
	4. Construction	36
	B. Personnel Administration	36

## HAMMERCMS System Design Description Table of Contents

1.	Overview	36
2.	Administrator Assignments	36
3.	Course Administrators	36
4.	Student Administrators	37
5.	System Administrators.	38
C.	Action Page Development	38
1.	Description	38
2.	Templates	39
4.	Completion Control	40
D.	General Employee Training (GET) Assignment	40
1.	GET versions	40
2.	GET assignment processing	41
3.	GET Table of Courses processing	42
E.	Prerequisites Processing	42
1.	Prerequisite Overrides	42
2.	Prerequisites Programming	43
F.	WHSP Refresher RPP WBT	43
1.	Operation	44
2.	Prerequisites.	44
3.	Terminology	45
4.	Access	45
5.	Lesson Operation	46
6.	Quirks	47
G.	Customer Course Data	48
H.	Database: wbtodb1	49
1.	Studentlog table management	49
2.	Completions table management	49
3.	owner_orgs table management	50
4.	Other Personnel tables management	50
I.	Database: wbtrecsdb	50
J.	Database: WBTProcsDB	50
K.	Database: internetTR	51
Initials and Acronyms		51
Appendix A. HAMMERCMS Database wbtodb1		1
Appendix B. HAMMERCMS Database WBTProcsDB		2
Appendix C. HAMMERCMS Database WBTRecsDB		2
Appendix D. HAMMERCMS Database offlineTRDB		2

# **HAMMER Courseware Management System**

## **Attachment 2.**

### **Web-Based Training Records Pre-Processor**

#### **Code Snippets**

## Web-Based Training Records Pre-Processor Code Snippets

```
' CHECK CONFIGURATION FILE BEFORE RUNNING!!!
'
' CHECK THE CONFIGURATION FILE RIGHT NOW!!
'
' DID YOU CHECK THE CONFIGURATION FILE?
'
' Background photo: Mt. Rainier, Courtesy of U.S. National Park Service
'
' March 17, 2006 -- tweaked to eliminate spaces in DB value retrieval.
' February 3, 2006 -- Joe said to add "Packet Size=4096" to the SQL DB connection strings according
to http://support.microsoft.com/kb/903002/en-us
' October 18, 2005 -- To account for "", took entire remainder of line for parameter value;
' therefore, no comments can be on a configuration line.
' October 14, 2005 -- restructured using a configuration file to facilitate configuration changes;
' added ProgrammersReportsDirectory,
' trapped exceptions for failed directories
' October 13, 2005 -- added configuration location for writing reports because down!
' March 2, 2005 -- corrected hh:mm to HH:mm
' Jaunary 29, 2004 -- modified to run on launch and exit upon completion
' Blackbox Functions are:
'
'     numberOfRowsRetrieved = RetrieveData(thisDBName, thisQuery, queryName)
'     Private Function RetrieveData(ByVal thisDBName As String, ByVal thisQuery As String, ByVal
queryName As String) As Long
'         (Returns -1 if exception raised).
'         Data retrieved into ProjectDataSet.Tables(queryName);
'         e.g., get column values by
'         For Each tableRow In ProjectDataSet.Tables(queryName).Rows
'             ProcessReport &= vbCrLf & tableRow.Item(0).ToString & "."
'         Next
'
'         or
'         columnValue = ProjectDataSet.Tables(queryName).Rows(0).Item(0).ToString
'
'     NOTE: Assumes either Access (.mdb) or SQL database
'     NOTE: Re-running same queryName clears out old data! (Save desired data somewhere else)
'
'     numberOfRowsAffected = UpdateDB(thisDBName, thisCommand)
'     Private Function UpdateDB(ByVal thisDBName As String, ByVal thisCommand As String) As Long
'     For non-retrieval queries.
'     (Returns -1 if exception raised).
```



## Web-Based Training Records Pre-Processor Code Snippets

```

'
'     NOTE: Assumes either Access (.mdb) or SQL database
'     NOTE: if thisDateTime is a date variable, then
'           sqlQuery &= " CONVERT(DATETIME,'" & thisDateTime.ToString & "','102)," for SQL databases
'           sqlQuery &= "#" & Format(thisDateTime, "MM/dd/yyyy hh:mm:ss tt") & "#," for Access
databases
'     is useful for inserting that value into a datetime field in the database.
'
'     maxValue = GetMaxValue(thisDBName, thisTableName, thisColumnName)
'     Private Function GetMaxValue(ByVal thisDBName As String, ByVal thisTableName As String, ByVal
thisColumnName As String) As Long
'     Retrieves the largest value in particular column; used to determine new Record Number, for
example.
'     (Returns -1 if exception raised).
'
'     thisText = InputText(sourceFileName)
'     Private Function InputText(ByVal sourceFileName As String) As String
'     Reads in entire contents of sourceFileName into InputText
'
'     success = SaveTextAs(sourceText, targetFileName)
'     Private Function SaveTextAs(ByRef sourceText As String, ByVal targetFileName As String) As
Boolean
'     Saves the contents of sourceText to targetFileName
'
'     success = ArchiveFile(baseDirectory, baseFileName)
'     Private Function ArchiveFile(ByVal baseDirectory As String, ByVal baseFileName As String) As
Boolean
'     Time stamps baseFileName, then archives it to baseDirectory\Archive\FYXXXX
'
'     success = DeleteFile(inputFileFullName)
'     Private Function DeleteFile(ByVal inputFileFullName As String) As Boolean
'     Deletes, without confirmation, inputFileFullName
'
'     newFileName = StampFileName(inputFileName, stampTime)
'     Private Function StampFileName(ByVal inputFileName As String, ByVal stampTime As Date) As
String
'     Appends stampTime to inputFileName to create a unique file name
'
'     leftString = LeftOf(inputString, numberOfCharacters)
'     Private Function LeftOf(ByVal inputString As String, ByVal numberOfCharacters As Integer) As
String

```

## Web-Based Training Records Pre-Processor Code Snippets

```

' Returns up to leftmost numberOfCharacters of inputString to the end of the string
'
' rightString = RightOf(inputString, numberOfCharacters)
' Private Function RightOf(ByVal inputString As String, ByVal numberOfCharacters As Integer) As
String
' Returns up to rightmost numberOfCharacters of inputString to the beginning of the string;
' e.g., may return the entire string if numberOfCharacters > inputString.length
'
' WaitSeconds(numberOfSeconds)
' Private Sub WaitSeconds(ByVal numberOfSeconds As Integer)
' Forces the program to do nothing for numberOfSeconds;
' may be used to wait while running an IE process, for example.
'
' ReportToStatusBox(thisReport)
' Private Sub ReportToStatusBox(ByVal thisReport As String)
' Writes thisReport to the StatusBox
'
' WriteToProcessReport
' Private Sub WriteToProcessReport(componentReport,exceptionReport)
' Writes ComponentReport to ProcessReport if UseComponentReport = True (default=False);
' Writes ExceptionReport to ProcessReport if UseExceptionReport = True (default).

Option Explicit On
Option Strict On

Imports System
Imports System.Data
Imports System.Data.SqlClient
Imports System.Data.OleDb
Imports System.IO ' For writing out reports to text files
Imports System.Net ' For retrieval of external, internetTR records

Public Class StandardForm
    Inherits System.Windows.Forms.Form

    Structure TrainingRecordStructure
        Dim StudentID As String

```

Web-Based Training Records Pre-Processor  
Code Snippets

```
Dim CourseNumber As String  
Dim CompDateTime As DateTime  
Dim CourseTitle As String  
Dim LastName As String  
Dim FirstName As String  
Dim MI As String
```

End Structure

```
Dim TrainingRecord(0) As TrainingRecordStructure  
Dim NumberOfTrainingRecords As Integer
```

#Region " Windows Form Designer generated code "

#End Region

```
Private Function DBConnectionString(ByVal thisDBName As String) As String
```

```
    Select Case thisDBName
```

```
        Case "WbtProcsDB"
```

```
        Case "internetTR"
```

```
        Case "WbtRecsDB"
```

```
        Case "wbtdb1"
```

```
        Case "wbtdb2"
```

```
        Case "offlineTRDB"
```

```
        Case "PSDB"
```

```
        Case "IncTRDB"
```

```
        Case "FitPlusDB.mdb"
```

```
        Case Else
```

Web-Based Training Records Pre-Processor  
Code Snippets

```
DBConnectionString = ""
```

```
End Select
```

```
End Function
```

```
Private Sub StandardForm_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
End Sub
```

```
Private Function GetConfigurationValue(ByVal configurationText As String, ByVal configurationKey As String) As String
```

```
End Function
```

```
Private Sub RunProcess()
```

```
ProcessReport = ControlChars.CrLf & ControlChars.CrLf & "Report Date/Time: " & Now.ToString & "."
```

```
ProcessReport &= "          HAMMERCMS Records Processing Report"
```

```
ProcessReport &= ControlChars.CrLf
```

```
ProcessReport &= ControlChars.CrLf & "Executing " & thisFileName & ", version " & thisFileDateTime & "."
```

```
ProcessReport &= ControlChars.CrLf & "Process initiated at " & Now.ToString & "."
```

```
' Archive the current process reports, Process_Report.txt and Process_Exceptions_Report.txt, if  
time
```

```
LocalProcessReport = ""
```

```
FileWritingStatus = "In progress"
```

```
currentTime = Now()
```

```
endTime = DateAdd(DateInterval.Second, numberOfWaitSeconds, currentTime)
```

```
waitFlag = True
```

```
SaveOldProcessReport()
```

```
While waitFlag
```

```
  If FileWritingStatus = "Complete" Then
```

```
    waitFlag = False
```

```
  Else
```

```
    currentTime = Now()
```

```
    If currentTime > endTime Then
```

## Web-Based Training Records Pre-Processor Code Snippets

```
        waitFlag = False      ' waited long enough; let come what may
    End If
End If
End While
```

```
LocalProcessReport = ProcessReport
```

```
AppendToProcessReport(LocalProcessReport)
```

```
ReportToStatusBox(ProcessReport)
```

```
continueProcess = LoadOfficialFromTimes()
```

```
GetinternetTRRecords()
```

```
UpdateinternetTRTrainingTable()
```

```
LoadTrainingRecords()
```

```
ReportToStatusBox(ProcessReport)
```

```
FilterForDuplicates()
```

```
FilterForKnownEmployees()
```

```
ReportToStatusBox(ProcessReport)
```

```
FilterForChargeCodes()
```

```
WriteToIncoming()
```

```
WriteNewReports()
```

```
UpdateRetrievalsInformation()
```

```
LocalProcessReport = "Process completed at " & Now.ToString & "."
```

```
End If
```

```
End Sub
```

```
#Region "Retrieval Processes 9/22/2003"
```

```
Private Function LoadOfficialFromTimes() As Boolean
```

```
End Function
```

## Web-Based Training Records Pre-Processor Code Snippets

Private Function GetinternetTRRecords() As Long

```
internetTRRecordsURL =  
"http://www.internetWBTR.com/vlsadmin/Report_internetTRSiteData.asp?UserID=internetTRRequest&FromDate=" &  
& OfficialFromDate(3) & "&ToDate=" & OfficialToDate
```

```
webGetData = WebRequest.Create(internetTRRecordsURL)  
' Pass existing Windows logon credentials to proxy  
webGetData.Credentials = CredentialCache.DefaultCredentials  
webGetData.Proxy.Credentials = CredentialCache.DefaultCredentials
```

```
Dim componentReport As String = "Component: GetinternetTRRecords " & Now.ToString & ". "  
Dim exceptionReport As String = ""
```

Try

```
webReturnedData = webGetData.GetResponse  
responseStream = webReturnedData.GetResponseStream  
reader = New StreamReader(responseStream)  
responseHTML = reader.ReadToEnd  
webReturnedData.Close()
```

Catch ex As Exception

```
OKToProcess = False  
UpdateLastCompDateTime(3) = False  
GetinternetTRRecords = -1  
exceptionReport = ex.Message
```

End Try

If exceptionReport.Length > 0 Then

```
WriteToProcessReport(componentReport, exceptionReport)
```

Else

```
LocalProcessReport = "Successfully read data stream from internetTR at " & Now.ToString & ". "  
AppendToProcessReport(LocalProcessReport)  
ProcessReport &= ControlChars.CrLf & LocalProcessReport
```

End If

If OKToProcess Then ' Must have connected successfully

```
' Records are expected to come in on lines as  
' internetTRID: PersonX  
' CourseNumber: 000000
```

Web-Based Training Records Pre-Processor  
Code Snippets

```

' CompletionDate: 1/1/2001
' CompletionTime: 11:00:00 AM
' internetTRID: PersonY
' etc.
'
' Basic process is this:
' Get next line
' If internetTRID, initialize new record
' If one of the other variables, if this field's turn,
' and if there was no intervening spurious data, then
' update with corresponding value
' Note that hitting a spurious line causes no data to be saved
' until hitting a new internetTRID.
' If all fields have values for current internetTRID, mark record as good
'
End Function

Private Function UpdateinternetTRTrainingTable() As Long

    thisRecordID = CInt(ProjectDataSet.Tables(queryName).Rows(0).Item("RecordID"))
    thisCompDateime =
    CStr(Format(CDate(IncominginternetTRRecords(recordNumber).CompletionDate), "MM/dd/yyyy") & " " &
    Format(CDate(IncominginternetTRRecords(recordNumber).CompletionTime), "hh:mm:ss"))
    sqlQuery = "Update internetTRTraining Set " &
    " CompDateime = CONVERT(DATETIME,'" & thisCompDateime & "','102),' " & _
    " BillingCode = '" & DefaultBillingCode & "','" & _
    " SystemInfo = 'Inserted CompDateime', " & _
    " WriteTime = CONVERT(DATETIME,'" & Now.ToString & "','102)' " & _
    " Where RecordID = " & thisRecordID
    numberOfRowsUpdated += UpdatedDB(thisDBName, sqlQuery)

    LocalProcessReport = "Completed updating " & numberOfRowsUpdated.ToString & " current
internetTR Training records at " & Now.ToString & "."
AppendToProcessReport(LocalProcessReport)
ProcessReport &= ControlChars.CrLf & LocalProcessReport

End Function

Private Function LoadTrainingRecords() As Long

' Get dated completions from internal WBT databases

```

Web-Based Training Records Pre-Processor  
Code Snippets

End Function

#End Region

#Region "Filtering Processes 9/22/2003"

Private Sub FilterForDuplicates()

End Sub

Private Sub FilterForKnownEmployees()

End Sub

Private Sub FilterForChargeCodes()

End Sub

#End Region

#Region "Writing Processes 9/22/2003"

Private Function WriteToIncoming() As Long

' All available training records have been retrieved and filtered;  
' write each record to Incoming Table for later reference and trouble-shooting, if needed

End Sub

Private Sub SaveOldProcessReport()

' This report, Process\_Report.txt, is archived monthly; otherwise, new information is just  
appended.

reportFileName = "Process\_Report.txt"

If File.Exists(ReportsDirectory & reportFileName) Then

    If File.GetLastWriteTime(ReportsDirectory & reportFileName).Month < Now.Month Then



## Web-Based Training Records Pre-Processor Code Snippets

```
        success = ArchiveFile(ReportsDirectory, reportFileName)
        archivedFiles = success
        success = DeleteFile(ReportsDirectory & reportFileName)
    End If
End If

End Sub

Private Sub WriteNewReports()

End Sub

Private Sub UpdateRetrievalsInformation()

End Sub

Private Function AppendToProcessReport(ByVal reportText As String) As Boolean

    Dim reportFileName As String
    Dim reportFileFullName As String

    reportFileName = ReportsDirectory & "Process_Report.txt"
    reportText = ControlChars.CrLf & reportText
    AppendToProcessReport = AppendTextTo(reportText, reportFileName)

End Function

Private Function AppendTextTo(ByRef sourceText As String, ByVal targetFileName As String) As Boolean

    ' Need to add this to Common Functions
    Dim fi As FileInfo = New FileInfo(targetFileName)
    Dim fs As FileStream
    Dim sw As StreamWriter

    Dim componentReport As String = "Component: AppendTextTo(" & targetFileName & ")"
    Dim exceptionReport As String = ""

    fs = New FileStream(targetFileName, FileMode.OpenOrCreate)
    fs.Close()
```

Web-Based Training Records Pre-Processor  
Code Snippets

```
fs = New FileStream(targetFileName, FileMode.Append)

AppendTextTo = True
For TryNumber = 1 To MaxNumberOfTries
    ReportToStatusBox("Appending text to " & targetFileName & ".")
    Try
        sw = New StreamWriter(fs)
        sw.Write(sourceText)
        sw.Flush()
        sw.Close()
        fs.Close()
        componentReport &= vbCrLf & "Successfully appended text on attempt number " &
TryNumber.ToString & "."
        ReportToStatusBox("Successfully saved file on attempt number " &
TryNumber.ToString & ".")
        TryNumber = MaxNumberOfTries ' To exit loop
    Catch ex As Exception
        If TryNumber = MaxNumberOfTries Then
            componentReport &= vbCrLf & "In " & MaxNumberOfTries.ToString & " attempts, unable to
save file."
            ReportToStatusBox("In " & MaxNumberOfTries.ToString & " attempts, unable to save
file.")
            exceptionReport = ex.Message
            AppendTextTo = False
        End If
    End Try
Next

WriteToProcessReport(componentReport, exceptionReport)

End Function

#End Region

#Region "Auxiliary Functions 9/22/2003"

Private Function PadString(ByVal inputString As String, ByVal totalLength As Integer) As String
    ' Function used to format text output
```

Web-Based Training Records Pre-Processor  
Code Snippets

```
Dim slotNumber As Integer
Dim padLength As Integer

If inputString.Length > totalLength Then
    PadString = LeftOf(inputString, totalLength)
Else
    PadString = inputString
    padLength = totalLength - inputString.Length
    For slotNumber = 1 To padLength
        PadString &= " "
    Next
End If
```

End Function

```
Private Function FixSingleQuotes(ByVal inputString As String) As String
```

```
    ' Function used to prepare strings for SQL input
    If inputString Is Nothing Then
        Return ""
    ElseIf inputString.Length = 0 Then
        Return ""
    Else
        Return Replace(inputString, "'", "''")
    End If
```

End Function

#End Region

#Region "Black Box Functions 9/25/2003"

```
Private Function RetrieveData(ByVal thisDBName As String, ByVal thisQuery As String, ByVal queryName As String) As Long
```

```
    ' Function returns -1 if exception raised; otherwise, returns the number of rows retrieved
```

```
    WriteToProcessReport(componentReport, exceptionReport)
```

## Web-Based Training Records Pre-Processor Code Snippets

End Function

```
Private Function UpdatedB(ByVal thisDBName As String, ByVal thisCommand As String) As Long
    ' Function returns -1 if exception raised; otherwise, returns the number of rows affected
    WriteToProcessReport(componentReport, exceptionReport)
    Return numberOfRowsAffected
End Function
```

```
Private Function GetMaxValue(ByVal thisDBName As String, ByVal thisTableName As String, ByVal
thisColumnName As String) As Long
```

```
    WriteToProcessReport(componentReport, exceptionReport)
```

End Function

```
Private Function InputText(ByVal sourceFileName As String) As String
```

```
    Dim sr As StreamReader
    Dim inputLine As String
```

```
    Dim componentReport As String = "Component: InputText(" & sourceFileName & ")"
    Dim exceptionReport As String = ""
    InputText = ""
```

```
    ' ReportToStatusBox("Seeking text from " & sourceFileName & ".")
```

```
    For TryNumber = 1 To MaxNumberOfTries
```

```
        ' ReportToStatusBox("Seeking text from " & sourceFileName & "; attempt number " &
TryNumber.ToString & " of " & MaxNumberOfTries & ".")
```

```
        Try
```

```
            sr = File.OpenText(sourceFileName)
            inputLine = sr.ReadLine()
            While Not inputLine Is Nothing
                InputText &= inputLine & vbCrLf
                inputLine = sr.ReadLine()
            End While
            sr.Close()
```

## Web-Based Training Records Pre-Processor Code Snippets

```
        componentReport &= vbCrLf & "Successfully retrieved text on attempt number " &
TryNumber.ToString & "."
        ReportToStatusBox("Successfully retrieved text on attempt number " &
TryNumber.ToString & ".")
        TryNumber = MaxNumberOfTries ' To exit loop
    Catch ex As Exception
        If TryNumber = MaxNumberOfTries Then
            componentReport &= vbCrLf & "In " & MaxNumberOfTries & " attempts, unable to retrieve
text."
            ReportToStatusBox("In " & MaxNumberOfTries & " attempts, unable to retrieve text.")
            exceptionReport = ex.Message
        End If
    End Try
Next

WriteToProcessReport(componentReport, exceptionReport)

End Function

Private Function SaveTextAs(ByRef sourceText As String, ByVal targetFileName As String) As Boolean

    Dim sw As StreamWriter
    SaveTextAs = True

    Dim componentReport As String = "Component: SaveTextAs(" & targetFileName & ")"
    Dim exceptionReport As String = ""

    For TryNumber = 1 To MaxNumberOfTries
        ReportToStatusBox("Saving " & targetFileName & "; attempt number " & TryNumber.ToString & " of
" & MaxNumberOfTries.ToString & ".")
        Try
            If File.Exists(targetFileName) Then
                File.Delete(targetFileName)
            End If
            sw = File.CreateText(targetFileName)
            sw.Write(sourceText)
            sw.Flush()
            sw.Close()
            componentReport &= vbCrLf & "Successfully saved file on attempt number " &
TryNumber.ToString & "."
        End Try
    Next
End Function
```

## Web-Based Training Records Pre-Processor Code Snippets

```

        ' ReportToStatusBox("Successfully saved file on attempt number " &
TryNumber.ToString & ".")
        TryNumber = MaxNumberOfTries ' To exit loop
    Catch ex As Exception
        If TryNumber = MaxNumberOfTries Then
            componentReport &= vbCrLf & "In " & MaxNumberOfTries.ToString & " attempts, unable to
save file."
            ReportToStatusBox("In " & MaxNumberOfTries.ToString & " attempts, unable to save
file.")
            exceptionReport = ex.Message
            SaveTextAs = False
        End If
    End Try
Next

    WriteToProcessReport(componentReport, exceptionReport)

End Function

Private Function ArchiveFile(ByVal baseDirectory As String, ByVal baseFileName As String) As Boolean
    ' This routine stamps the baseFileName with its last Modified Date/Time,
    ' then saves it to an archive\FYXXX subdirectory.\
    ' Note. This can be tricky to test because archive file saved with its last Date Modified
property.

    Dim baseFileFullName As String
    Dim archiveFileFullName As String
    Dim baseSubDirectory As String

    Dim deletedBaseFile As Boolean

    Dim testDate As String
    Dim tempDate As String
    Dim tempInt As Integer

    Dim currentFY As String
    Dim f As FileInfo

    Dim componentReport As String = "Component: ArchiveFile(" & baseDirectory & ", " & baseFileName &
")"
```

## Web-Based Training Records Pre-Processor Code Snippets

```
Dim exceptionReport As String = ""

baseFileFullName = baseDirectory & baseFileName
For TryNumber = 1 To MaxNumberOfTries
    ReportToStatusBox("Attempting to archive " & baseFileFullName & "; attempt number " &
    TryNumber.ToString & " of " & MaxNumberOfTries.ToString & ".")
    Try
        If File.Exists(baseFileFullName) Then
            archiveFileFullName = StampFileName(baseFileName, FileDateTime(baseFileFullName))
            testDate = Format(FileDateTime(baseFileFullName), "yyyy/MM/dd")
            For tempInt = 2000 To 2030
                tempDate = Format(CDate("10/1/" & tempInt.ToString), "yyyy/MM/dd")
                If testDate < tempDate Then
                    currentFY = "FY " & tempInt.ToString
                    Exit For
                End If
            Next
            baseSubDirectory = baseDirectory & "archive\" & currentFY & "\"
            If Not Directory.Exists(baseSubDirectory) Then
                Directory.CreateDirectory(baseSubDirectory)
            End If
            archiveFileFullName = baseSubDirectory & archiveFileFullName
            If File.Exists(archiveFileFullName) Then
                f = New FileInfo(archiveFileFullName)
                f.Attributes = FileAttributes.Normal
                f.Delete()
            End If
            File.Copy(baseFileFullName, archiveFileFullName)
            ArchiveFile = True
            componentReport &= vbCrLf & "On attempt number " & TryNumber.ToString & ", successfully
            copied " & baseFileFullName & " to " & archiveFileFullName & " at " & Now.ToString & "."
            ReportToStatusBox("On attempt number " & TryNumber.ToString & ",
            successfully copied " & baseFileFullName & " to " & archiveFileFullName & " at " & Now.ToString & ".")
            TryNumber = MaxNumberOfTries ' To exit loop
        Else
            If TryNumber = MaxNumberOfTries Then
                ArchiveFile = False
                componentReport &= vbCrLf & "In " & MaxNumberOfTries.ToString & " attempts, unable
            to archive " & baseFileFullName & "."
                ReportToStatusBox("In " & MaxNumberOfTries.ToString & " attempts, unable to archive
            " & baseFileFullName & ".")
            End If
        End Try
    Next
End For
```

## Web-Based Training Records Pre-Processor Code Snippets

```

        exceptionReport = "Could not find the file."
    End If
End If
Catch ex As Exception
    If TryNumber = MaxNumberOfTries Then
        ArchiveFile = False
        componentReport &= vbCrLf & "In " & MaxNumberOfTries.ToString & " attempts, unable to
save " & archiveFileFullName & "."
        exceptionReport = ex.Message
        ReportToStatusBox("In " & MaxNumberOfTries.ToString & " attempts, unable to save " &
archiveFileFullName & ".")
    End If
End Try
Next

    WriteToProcessReport(componentReport, exceptionReport)

End Function

Private Function StampFileName(ByVal inputFileName As String, ByVal stampTime As Date) As String

    Dim tempFileName As String
    Dim fileExtension As String
    Dim dateString As String

    ' An atom; think of LeftOf as a subatomic particle

    If inputFileName.Length = 0 Then
        inputFileName = "dummyXXXfile" ' Hey! We'll return something
    End If
    dateString = Format(stampTime, "MM_dd_yyyy_hh_mm_ss").ToString
    fileExtension = Path.GetExtension(inputFileName)
    tempFileName = LeftOf(inputFileName, inputFileName.Length - fileExtension.Length)
    tempFileName &= "_" & dateString & fileExtension

    Return tempFileName

End Function

Private Function DeleteFile(ByVal inputFileFullName As String) As Boolean

```



## Web-Based Training Records Pre-Processor Code Snippets

```

Dim f As FileInfo
Dim componentReport As String = "Component: DeleteFile(" & inputFileFullName & ")"
Dim exceptionReport As String = ""

For TryNumber = 1 To MaxNumberOfTries
    ReportToStatusBox("Attempting to delete " & inputFileFullName & "; attempt number " &
    TryNumber.ToString & " of " & MaxNumberOfTries.ToString & ".")
    Try
        f = New FileInfo(inputFileFullName)
        f.Attributes = FileAttributes.Normal
        f.Delete()
        DeleteFile = True
        componentReport &= vbCrLf & "Successfully deleted the file on attempt number " &
        TryNumber.ToString & ".
    , ReportToStatusBox("Successfully deleted the file on attempt number " &
    TryNumber.ToString & ".")
        TryNumber = MaxNumberOfTries ' To exit loop
    Catch ex As Exception
        If TryNumber = MaxNumberOfTries Then
            DeleteFile = False
            componentReport &= vbCrLf & "In " & MaxNumberOfTries.ToString & " attempts, failed to
            delete the file."
        End If
        ReportToStatusBox("In " & MaxNumberOfTries.ToString & " attempts, failed to delete the
        file.")
        exceptionReport = ex.Message
    End If
    End Try
Next

WriteToProcessReport(componentReport, exceptionReport)

End Function

Private Function LeftOf(ByVal inputString As String, ByVal numberOfCharacters As Integer) As String
    Dim returnString As String

    If numberOfCharacters <= 0 Then
        returnString = ""
    ElseIf numberOfCharacters > inputString.Length Then

```

Web-Based Training Records Pre-Processor  
Code Snippets

```

returnString = inputString
Else
returnString = inputString.Substring(0, numberOfCharacters)
End If
Return returnString
End Function
Private Function RightOf(ByVal inputString As String, ByVal numberOfCharacters As Integer) As String
End Function
Private Sub WaitSeconds(ByVal numberOfSeconds As Integer)
End Sub
Private Sub ReportToStatusBar(ByVal thisReport As String)
StatusBar.Text = thisReport
Application.DoEvents()
End Sub
Private Sub WriteToProcessReport(ByVal componentReport As String, ByVal exceptionReport As String)
End Sub
#End Region
End Class

```

# **HAMMER Courseware Management System**

## **Attachment 3.**

### **WBT Records**

### **Process Reports**

WBT Records  
Process Reports

Report Date/Time: 2/6/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.

Process initiated at 2/6/2006 4:00:10 PM.

For wbtddb1, OfficialFromDateTime = 2/3/2006 3:30:01 PM; record retrieved at 2/6/2006 4:00:11 PM.

For wbtddb2, OfficialFromDateTime = 9/21/2005 3:30:12 PM; record retrieved at 2/6/2006 4:00:11 PM.

For internetTR, OfficialFromDateTime = 2/2/2006 3:30:11 PM; record retrieved at 2/6/2006 4:00:11 PM.

For FitPlusDB.mdb, OfficialFromDateTime = 2/3/2006 3:30:01 PM; record retrieved at 2/6/2006 4:00:11 PM.

For offlineTRDB, OfficialFromDateTime = 2/3/2006 2:56:55 PM; record retrieved at 2/6/2006 4:00:11 PM.

Successfully read data stream from internetTR at 2/6/2006 4:00:11 PM.

Parsed 0 completion records from internetTR data stream at 2/6/2006 4:00:11 PM.

Successfully connected but found no internetTR records in time interval 2/2/2006 3:30:11 PM to 2/6/2006 3:30:11 PM.

Retrieved 172 completion records from wbtddb1 at 2/6/2006 4:00:11 PM.

Retrieved 0 completion records from wbtddb2 at 2/6/2006 4:00:12 PM.

Retrieved 0 completion records from internetTR at 2/6/2006 4:00:12 PM.

Retrieved 9 unique completion records from FitPlusDB.mdb at 2/6/2006 4:00:24 PM.

Retrieved 151 completion records from offlineTRDB at 2/6/2006 4:00:26 PM.

Total number of completion records retrieved = 332; retrieval completed at 2/6/2006 4:00:26 PM.

Found 332 unique records from 332 submitted completion records; filtering completed at 2/6/2006 4:00:26 PM.

Matched 317 of 332 person records in HR database; filtering completed at 2/6/2006 4:00:28 PM.

Found valid charge codes for 332 of 332 writeable records; filtering completed at 2/6/2006 4:00:28 PM.

Successfully wrote 332 new records to Training's Incoming Table at 2/6/2006 4:00:32 PM.

Completed archiving current records report file(s) at 2/6/2006 4:00:32 PM

Successfully wrote reports\web\_completions.txt at 2/6/2006 4:00:33 PM.

Successfully updated the LastCompDate to 2/6/2006 3:30:11 PM for wbtddb1.

LastCompDate was not updated for wbtddb2.

LastCompDate was not updated for internetTR.

Successfully updated the LastCompDate to 2/6/2006 3:30:11 PM for FitPlusDB.mdb.

Successfully updated the LastCompDate to 2/6/2006 3:30:11 PM for offlineTRDB.

Process completed at 2/6/2006 4:00:33 PM.

WBT Records  
Process Reports

Report Date/Time: 2/7/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.

Process initiated at 2/7/2006 4:00:10 PM.

For wbtddb1, OfficialFromDateTime = 2/6/2006 3:30:11 PM; record retrieved at 2/7/2006 4:00:11 PM.

For wbtddb2, OfficialFromDateTime = 9/21/2005 3:30:12 PM; record retrieved at 2/7/2006 4:00:11 PM.

For internetTR, OfficialFromDateTime = 2/2/2006 3:30:11 PM; record retrieved at 2/7/2006 4:00:11 PM.

For FitPlusDB.mdb, OfficialFromDateTime = 2/6/2006 3:30:11 PM; record retrieved at 2/7/2006 4:00:11 PM.

For offlineTRDB, OfficialFromDateTime = 2/6/2006 3:30:11 PM; record retrieved at 2/7/2006 4:00:11 PM.

Successfully read data stream from internetTR at 2/7/2006 4:00:11 PM.

Parsed 0 completion records from internetTR data stream at 2/7/2006 4:00:11 PM.

Successfully connected but found no internetTR records in time interval 2/2/2006 3:30:11 PM to 2/7/2006 3:30:11 PM.

Retrieved 199 completion records from wbtddb1 at 2/7/2006 4:00:11 PM.

Retrieved 0 completion records from wbtddb2 at 2/7/2006 4:00:11 PM.

Retrieved 0 completion records from internetTR at 2/7/2006 4:00:11 PM.

Retrieved 10 unique completion records from FitPlusDB.mdb at 2/7/2006 4:00:24 PM.

Retrieved 0 completion records from offlineTRDB at 2/7/2006 4:00:24 PM.

Total number of completion records retrieved = 209; retrieval completed at 2/7/2006 4:00:24 PM.

Found 209 unique records from 209 submitted completion records; filtering completed at 2/7/2006 4:00:25 PM.

Matched 206 of 209 person records in HR database; filtering completed at 2/7/2006 4:00:28 PM.

Found valid charge codes for 209 of 209 writeable records; filtering completed at 2/7/2006 4:00:28 PM.

Successfully wrote 209 new records to Training's Incoming Table at 2/7/2006 4:00:32 PM.

Completed archiving current records report file(s) at 2/7/2006 4:00:32 PM

Successfully wrote reports\web\_completions.txt at 2/7/2006 4:00:32 PM.

Successfully updated the LastCompDate to 2/7/2006 3:30:11 PM for wbtddb1.

LastCompDate was not updated for wbtddb2.

LastCompDate was not updated for internetTR.

Successfully updated the LastCompDate to 2/7/2006 3:30:11 PM for FitPlusDB.mdb.

LastCompDate was not updated for offlineTRDB.

Process completed at 2/7/2006 4:00:32 PM.

WBT Records  
Process Reports

Report Date/Time: 2/8/2006 4:00:10 PM.      HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.

Process initiated at 2/8/2006 4:00:10 PM.

For wbtodb1, OfficialFromDateTime = 2/7/2006 3:30:11 PM; record retrieved at 2/8/2006 4:00:11 PM.

For wbtodb2, OfficialFromDateTime = 9/21/2005 3:30:12 PM; record retrieved at 2/8/2006 4:00:11 PM.

For internetTR, OfficialFromDateTime = 2/2/2006 3:30:11 PM; record retrieved at 2/8/2006 4:00:11 PM.

For FitPlusDB.mdb, OfficialFromDateTime = 2/7/2006 3:30:11 PM; record retrieved at 2/8/2006 4:00:11 PM.

For offlineTRDB, OfficialFromDateTime = 2/6/2006 3:30:11 PM; record retrieved at 2/8/2006 4:00:11 PM.

Successfully read data stream from internetTR at 2/8/2006 4:00:11 PM.

Parsed 2 completion records from internetTR data stream at 2/8/2006 4:00:11 PM.

Completed writing 2 new rows to internetTRIncoming table at 2/8/2006 4:00:11 PM.

Successfully retrieved 2 internetTR records at 2/8/2006 4:00:11 PM.

Completed updating 2 current internetTR Training records at 2/8/2006 4:00:11 PM.

Successfully updated 2 internetTRTraining records at 2/8/2006 4:00:11 PM.

Retrieved 215 completion records from wbtodb1 at 2/8/2006 4:00:11 PM.

Retrieved 0 completion records from wbtodb2 at 2/8/2006 4:00:12 PM.

Retrieved 2 completion records from internetTR at 2/8/2006 4:00:12 PM.

Retrieved 11 unique completion records from FitPlusDB.mdb at 2/8/2006 4:00:25 PM.

Retrieved 0 completion records from offlineTRDB at 2/8/2006 4:00:25 PM.

Total number of completion records retrieved = 228; retrieval completed at 2/8/2006 4:00:25 PM.

Found 228 unique records from 228 submitted completion records; filtering completed at 2/8/2006 4:00:25 PM.

Matched 226 of 228 person records in HR database; filtering completed at 2/8/2006 4:00:28 PM.

Found valid charge codes for 228 of 228 writeable records; filtering completed at 2/8/2006 4:00:28 PM.

Successfully wrote 228 new records to Training's Incoming Table at 2/8/2006 4:00:32 PM.

Completed archiving current records report file(s) at 2/8/2006 4:00:32 PM

Successfully wrote reports\web\_completions.txt at 2/8/2006 4:00:32 PM.

Successfully updated the LastCompDate to 2/8/2006 3:30:11 PM for wbtodb1.

LastCompDate was not updated for wbtodb2.

WBT Records  
Process Reports

Successfully updated the LastCompDateTime to 2/8/2006 3:30:11 PM for internetTR.  
Successfully updated the LastCompDateTime to 2/8/2006 3:30:11 PM for FitPlusDB.mdb.  
LastCompDateTime was not updated for offlineTRDB.  
Process completed at 2/8/2006 4:00:32 PM.

Report Date/Time: 2/9/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.

Process initiated at 2/9/2006 4:00:10 PM.

For wbtddb1, OfficialFromDate/Time = 2/8/2006 3:30:11 PM; record retrieved at 2/9/2006 4:00:10 PM.

For wbtddb2, OfficialFromDate/Time = 9/21/2005 3:30:12 PM; record retrieved at 2/9/2006 4:00:11 PM.

For internetTR, OfficialFromDate/Time = 2/8/2006 3:30:11 PM; record retrieved at 2/9/2006 4:00:11 PM.

For FitPlusDB.mdb, OfficialFromDate/Time = 2/8/2006 3:30:11 PM; record retrieved at 2/9/2006 4:00:11 PM.

For offlineTRDB, OfficialFromDate/Time = 2/6/2006 3:30:11 PM; record retrieved at 2/9/2006 4:00:11 PM.

Successfully read data stream from internetTR at 2/9/2006 4:00:11 PM.

Parsed 2 completion records from internetTR data stream at 2/9/2006 4:00:11 PM.

Completed writing 2 new rows to internetTRIncoming table at 2/9/2006 4:00:11 PM.

Successfully retrieved 2 internetTR records at 2/9/2006 4:00:11 PM.

Completed updating 2 current internetTR Training records at 2/9/2006 4:00:11 PM.

Successfully updated 2 internetTR Training records at 2/9/2006 4:00:11 PM.

Retrieved 239 completion records from wbtddb1 at 2/9/2006 4:00:12 PM.

Retrieved 0 completion records from wbtddb2 at 2/9/2006 4:00:12 PM.

Retrieved 0 completion records from internetTR at 2/9/2006 4:00:12 PM.

Retrieved 18 unique completion records from FitPlusDB.mdb at 2/9/2006 4:00:26 PM.

Retrieved 113 completion records from offlineTRDB at 2/9/2006 4:00:26 PM.

Total number of completion records retrieved = 370; retrieval completed at 2/9/2006 4:00:26 PM.

Found 370 unique records from 370 submitted completion records; filtering completed at 2/9/2006 4:00:26 PM.

Matched 370 of 370 person records in HR database; filtering completed at 2/9/2006 4:00:32 PM.

Found valid charge codes for 370 of 370 writeable records; filtering completed at 2/9/2006 4:00:33 PM.

Successfully wrote 370 new records to Trainings Incoming Table at 2/9/2006 4:00:37 PM.

WBT Records  
Process Reports

Completed archiving current records report file(s) at 2/9/2006 4:00:37 PM  
Successfully wrote reports\web\_completions.txt at 2/9/2006 4:00:38 PM.  
Successfully updated the LastCompDate to 2/9/2006 3:30:11 PM for wbtb1.  
LastCompDate Time was not updated for wbtb2.  
LastCompDate Time was not updated for internetTR.  
Successfully updated the LastCompDate to 2/9/2006 3:30:11 PM for FitPlusDB.mdb.  
Successfully updated the LastCompDate to 2/9/2006 3:30:11 PM for offlineTRDB.  
Process completed at 2/9/2006 4:00:38 PM.

Report Date/Time: 2/10/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.  
Process initiated at 2/10/2006 4:00:10 PM.  
For wbtb1, OfficialFromDate = 2/9/2006 3:30:11 PM; record retrieved at 2/10/2006 4:00:11 PM.  
For wbtb2, OfficialFromDate = 9/21/2005 3:30:12 PM; record retrieved at 2/10/2006 4:00:11 PM.  
For internetTR, OfficialFromDate = 2/8/2006 3:30:11 PM; record retrieved at 2/10/2006 4:00:11 PM.  
For FitPlusDB.mdb, OfficialFromDate = 2/9/2006 3:30:11 PM; record retrieved at 2/10/2006 4:00:11 PM.  
For offlineTRDB, OfficialFromDate = 2/9/2006 3:30:11 PM; record retrieved at 2/10/2006 4:00:11 PM.  
Successfully read data stream from internetTR at 2/10/2006 4:00:11 PM.  
Parsed 2 completion records from internetTR data stream at 2/10/2006 4:00:11 PM.  
Completed writing 2 new rows to internetTRIncoming table at 2/10/2006 4:00:11 PM.  
Successfully retrieved 2 internetTR records at 2/10/2006 4:00:11 PM.  
Completed updating 2 current internetTR Training records at 2/10/2006 4:00:11 PM.  
Successfully updated 2 internetTR Training records at 2/10/2006 4:00:11 PM.  
Retrieved 99 completion records from wbtb1 at 2/10/2006 4:00:12 PM.  
Retrieved 0 completion records from wbtb2 at 2/10/2006 4:00:12 PM.  
Retrieved 0 completion records from internetTR at 2/10/2006 4:00:12 PM.  
Retrieved 0 unique completion records from FitPlusDB.mdb at 2/10/2006 4:00:24 PM.  
Retrieved 0 completion records from offlineTRDB at 2/10/2006 4:00:24 PM.  
Total number of completion records retrieved = 99; retrieval completed at 2/10/2006 4:00:24 PM.



WBT Records  
Process Reports

Found 99 unique records from 99 submitted completion records; filtering completed at 2/10/2006 4:00:24 PM.  
Matched 97 of 99 person records in HR database; filtering completed at 2/10/2006 4:00:27 PM.  
Found valid charge codes for 99 of 99 writeable records; filtering completed at 2/10/2006 4:00:28 PM.  
Successfully wrote 99 new records to Training's Incoming Table at 2/10/2006 4:00:30 PM.  
Completed archiving current records report file(s) at 2/10/2006 4:00:30 PM  
Successfully wrote reports\web\_completions.txt at 2/10/2006 4:00:31 PM.  
Successfully updated the LastCompDateTime to 2/10/2006 3:30:11 PM for wbtodb1.  
LastCompDateTime was not updated for wbtodb2.  
LastCompDateTime was not updated for internetTR.  
LastCompDateTime was not updated for FitPlusDB.mdb.  
LastCompDateTime was not updated for offlineTRDB.  
Process completed at 2/10/2006 4:00:31 PM.

Report Date/Time: 2/13/2006 4:00:10 PM.      HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.  
Process initiated at 2/13/2006 4:00:10 PM.  
For wbtodb1, OfficialFromDateTime = 2/10/2006 3:30:11 PM; record retrieved at 2/13/2006 4:00:11 PM.  
For wbtodb2, OfficialFromDateTime = 9/21/2005 3:30:12 PM; record retrieved at 2/13/2006 4:00:11 PM.  
For internetTR, OfficialFromDateTime = 2/8/2006 3:30:11 PM; record retrieved at 2/13/2006 4:00:11 PM.  
For FitPlusDB.mdb, OfficialFromDateTime = 2/9/2006 3:30:11 PM; record retrieved at 2/13/2006 4:00:11 PM.  
For offlineTRDB, OfficialFromDateTime = 2/9/2006 3:30:11 PM; record retrieved at 2/13/2006 4:00:11 PM.  
Successfully read data stream from internetTR at 2/13/2006 4:00:11 PM.  
Parsed 2 completion records from internetTR data stream at 2/13/2006 4:00:11 PM.  
Completed writing 2 new rows to internetTRIncoming table at 2/13/2006 4:00:11 PM.  
Successfully retrieved 2 internetTR records at 2/13/2006 4:00:11 PM.  
Completed updating 2 current internetTR Training records at 2/13/2006 4:00:11 PM.  
Successfully updated 2 internetTRTraining records at 2/13/2006 4:00:11 PM.  
Retrieved 373 completion records from wbtodb1 at 2/13/2006 4:00:12 PM.  
Retrieved 0 completion records from internetTR at 2/13/2006 4:00:12 PM.

WBT Records  
Process Reports

Retrieved 15 unique completion records from FitPlusDB.mdb at 2/13/2006 4:00:26 PM.  
 Retrieved 110 completion records from offlineTRDB at 2/13/2006 4:00:26 PM.  
 Total number of completion records retrieved = 499; retrieval completed at 2/13/2006 4:00:26 PM.  
 Found 499 unique records from 499 submitted completion records; filtering completed at 2/13/2006 4:00:26 PM.  
 Matched 479 of 499 person records in HR database; filtering completed at 2/13/2006 4:00:32 PM.  
 Found valid charge codes for 499 of 499 writeable records; filtering completed at 2/13/2006 4:00:33 PM.  
 Successfully wrote 499 new records to Training's Incoming Table at 2/13/2006 4:00:38 PM.  
 Completed archiving current records report file(s) at 2/13/2006 4:00:38 PM  
 Successfully wrote reports\web\_completions.txt at 2/13/2006 4:00:39 PM.  
 Successfully updated the LastCompDateTime to 2/13/2006 3:30:11 PM for wbtddb1.  
 Successfully updated the LastCompDateTime to 2/13/2006 3:30:11 PM for wbtddb2.  
 LastCompDateTime was not updated for internetTR.  
 Successfully updated the LastCompDateTime to 2/13/2006 3:30:11 PM for FitPlusDB.mdb.  
 Successfully updated the LastCompDateTime to 2/13/2006 3:30:11 PM for offlineTRDB.  
 Process completed at 2/13/2006 4:00:39 PM.

Report Date/Time: 2/14/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.  
 Process initiated at 2/14/2006 4:00:10 PM.  
 For wbtddb1, OfficialFromDateTime = 2/13/2006 3:30:11 PM; record retrieved at 2/14/2006 4:00:11 PM.  
 For wbtddb2, OfficialFromDateTime = 2/13/2006 3:30:11 PM; record retrieved at 2/14/2006 4:00:11 PM.  
 For internetTR, OfficialFromDateTime = 2/8/2006 3:30:11 PM; record retrieved at 2/14/2006 4:00:11 PM.  
 For FitPlusDB.mdb, OfficialFromDateTime = 2/13/2006 3:30:11 PM; record retrieved at 2/14/2006 4:00:11 PM.  
 For offlineTRDB, OfficialFromDateTime = 2/13/2006 3:30:11 PM; record retrieved at 2/14/2006 4:00:11 PM.  
 Successfully read data stream from internetTR at 2/14/2006 4:00:11 PM.  
 Parsed 2 completion records from internetTR data stream at 2/14/2006 4:00:11 PM.  
 Completed writing 2 new rows to internetTRIncoming table at 2/14/2006 4:00:11 PM.  
 Successfully retrieved 2 internetTR records at 2/14/2006 4:00:11 PM.  
 Completed updating 2 current internetTR Training records at 2/14/2006 4:00:11 PM.

## WBT Records Process Reports

Successfully updated 2 internetTRTraining records at 2/14/2006 4:00:11 PM.  
Retrieved 217 completion records from wbtddb1 at 2/14/2006 4:00:11 PM.  
Retrieved 0 completion records from wbtddb2 at 2/14/2006 4:00:12 PM.  
Retrieved 0 completion records from internetTR at 2/14/2006 4:00:12 PM.  
Retrieved 18 unique completion records from FitPlusDB.mdb at 2/14/2006 4:00:26 PM.  
Retrieved 48 completion records from offlineTRDB at 2/14/2006 4:00:26 PM.  
Total number of completion records retrieved = 283; retrieval completed at 2/14/2006 4:00:26 PM.  
Found 283 unique records from 283 submitted completion records; filtering completed at 2/14/2006 4:00:26 PM.  
Matched 262 of 283 person records in HR database; filtering completed at 2/14/2006 4:00:30 PM.  
Found valid charge codes for 283 of 283 writeable records; filtering completed at 2/14/2006 4:00:30 PM.  
Successfully wrote 283 new records to Training's Incoming Table at 2/14/2006 4:00:34 PM.  
Completed archiving current records report file(s) at 2/14/2006 4:00:34 PM  
Successfully wrote reports\web\_completions.txt at 2/14/2006 4:00:35 PM.  
Successfully updated the LastCompDateTime to 2/14/2006 3:30:11 PM for wbtddb1.  
LastCompDateTime was not updated for wbtddb2.  
LastCompDateTime was not updated for internetTR.  
Successfully updated the LastCompDateTime to 2/14/2006 3:30:11 PM for FitPlusDB.mdb.  
Successfully updated the LastCompDateTime to 2/14/2006 3:30:11 PM for offlineTRDB.  
Process completed at 2/14/2006 4:00:35 PM.

Report Date/Time: 2/15/2006 4:00:10 PM. HAMMERCMS Records Processing Report

Executing wbtrecs.exe, version 2/3/2006 9:19:06 AM.  
Process initiated at 2/15/2006 4:00:10 PM.  
For wbtddb1, OfficialFromDateTime = 2/14/2006 3:30:11 PM; record retrieved at 2/15/2006 4:00:10 PM.  
For wbtddb2, OfficialFromDateTime = 2/13/2006 3:30:11 PM; record retrieved at 2/15/2006 4:00:10 PM.  
For internetTR, OfficialFromDateTime = 2/8/2006 3:30:11 PM; record retrieved at 2/15/2006 4:00:10 PM.  
For FitPlusDB.mdb, OfficialFromDateTime = 2/14/2006 3:30:11 PM; record retrieved at 2/15/2006 4:00:11 PM.  
For offlineTRDB, OfficialFromDateTime = 2/14/2006 3:30:11 PM; record retrieved at 2/15/2006 4:00:11 PM.  
Successfully read data stream from internetTR at 2/15/2006 4:00:11 PM.

WBT Records  
Process Reports

Parsed 5 completion records from internetTR data stream at 2/15/2006 4:00:11 PM.  
Completed writing 5 new rows to internetTRIncoming table at 2/15/2006 4:00:11 PM.  
Successfully retrieved 5 internetTR records at 2/15/2006 4:00:11 PM.  
Completed updating 5 current internetTR Training records at 2/15/2006 4:00:11 PM.  
Successfully updated 5 internetTRTraining records at 2/15/2006 4:00:11 PM.  
Retrieved 168 completion records from wbtodb1 at 2/15/2006 4:00:11 PM.  
Retrieved 0 completion records from wbtodb2 at 2/15/2006 4:00:11 PM.  
Retrieved 3 completion records from internetTR at 2/15/2006 4:00:11 PM.  
Retrieved 24 unique completion records from FitPlusDB.mdb at 2/15/2006 4:00:27 PM.  
Retrieved 100 completion records from offlineTRDB at 2/15/2006 4:00:27 PM.  
Total number of completion records retrieved = 295; retrieval completed at 2/15/2006 4:00:27 PM.  
Found 295 unique records from 295 submitted completion records; filtering completed at 2/15/2006 4:00:28 PM.  
Matched 293 of 295 person records in HR database; filtering completed at 2/15/2006 4:00:32 PM.  
Found valid charge codes for 295 of 295 writeable records; filtering completed at 2/15/2006 4:00:32 PM.  
Successfully wrote 295 new records to Training's Incoming Table at 2/15/2006 4:00:36 PM.  
Completed archiving current records report file(s) at 2/15/2006 4:00:36 PM  
Successfully wrote reports\web\_completions.txt at 2/15/2006 4:00:37 PM.  
Successfully updated the LastCompDateTime to 2/15/2006 3:30:11 PM for wbtodb1.  
LastCompDateTime was not updated for wbtodb2.  
Successfully updated the LastCompDateTime to 2/15/2006 3:30:11 PM for internetTR.  
Successfully updated the LastCompDateTime to 2/15/2006 3:30:11 PM for FitPlusDB.mdb.  
Successfully updated the LastCompDateTime to 2/15/2006 3:30:11 PM for offlineTRDB.  
Process completed at 2/15/2006 4:00:37 PM.

# **HAMMER Courseware Management System**

## **Attachment 4.**

**wbtrecs**

**Problem: 2/8/2006**

Problem Report: PersonA completed the course 100099 on the internet but no record went to Training Records.

1. Report from internet training database for 2/8/2006

### Training Report

Course_Name	Student_Id	Last_Name	First_Name	Middle_Initial	Event_Date	Event_Time
100099 - Hanford Site Orientation	PersonX	X	Person		2/8/2006	7:08:38 PM
100099 - Hanford Site Orientation	PersonY	Y	Person		2/8/2006	12:38:18 PM
100099 - Hanford Site Orientation	PersonZ	Z	Person		2/8/2006	10:20:09 AM
100099 - Hanford Site Orientation	PersonA	A	Person		2/8/2006	8:40:33 PM

2. No records found from internet training database for 2/9/2006, 2/10/2006, 2/11/2006, 2/12/2006

3. Report from internet training database for 2/13/2006

### Training Report

Course_Name	Student_Id	Last_Name	First_Name	Middle_Initial	Event_Date	Event_Time
HAMMER Facility Orientation	PersonM	M	Person		2/13/2006	10:39:11 AM
HAMMER Facility Orientation	PersonN	N	Person	M	2/13/2006	11:13:07 AM

wbtrecs  
 Problem: 2/8/2006

4. Data rows in externalIncoming training table

eHarfordID	CourseNumber	CompletionDate	CompletionTime	SystemInfo	WriteTime
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/9/2006 4:00:11 PM
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/10/2006 4:00:11 PM
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/13/2006 4:00:11 PM
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/14/2006 4:00:11 PM
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/15/2006 4:00:11 PM
PERSONA	100099	2/8/2006	8:40:33 PM	Good	2/16/2006 8:03:06 AM

5. Data row in externalTraining training table

eHarfordID	CourseNumber	CourseTitle	CompDate Time	SystemInfo	WriteTime
PERSONA	100099	Harford Site Orient	2/8/2006 8:40:33 / Inserted CompDate	2/16/2006 8:03:06	

6. Data row in incomingRecords table for PersonA data  
 WHERE (CompDate Time BETWEEN CONVERT(DATETIME, '2006-02-08 00:00:00', 102) AND CONVERT(DATETIME, '2006-02-16 00:00:00', 102))

StudentID	CourseNumber	Course Title	Last Name	First Name	Good	Write Status	SystemInfo	Write Time	CompDate Time

## **HAMMER Courseware Management System**

### **Attachment 5.**

**wbtrecs**

**Fix: 2/8/2006**



wbtrecs  
Fix: 2/8/2006

(Expanded) data row in externalTraining table

PersonID	CourseNumber	CourseTitle	CompDateTime	SystemInfo	WriteTime
PersonA	100099	Hanford Site Orient	2/8/2006 8:40:33 AM	Inserted CompDate	2/16/2006 8:03:06

Faulty code

```
                thisCompDatetime =  
CStr(Format(CDate(IncominginternetTRRecords(recordNumber).CompletionDate), "MM/dd/yyyy") & " " &  
Format(CDate(IncominginternetTRRecords(recordNumber).CompletionTime), "hh:mm:ss"))
```

defaulted to AM !

Corrected code

```
                thisCompDatetime =  
CStr(Format(CDate(IncominginternetTRRecords(recordNumber).CompletionDate), "MM/dd/yyyy") & " " &  
Format(CDate(IncominginternetTRRecords(recordNumber).CompletionTime), "hh:mm:ss tt"))
```

# **HAMMER Courseware Management System**

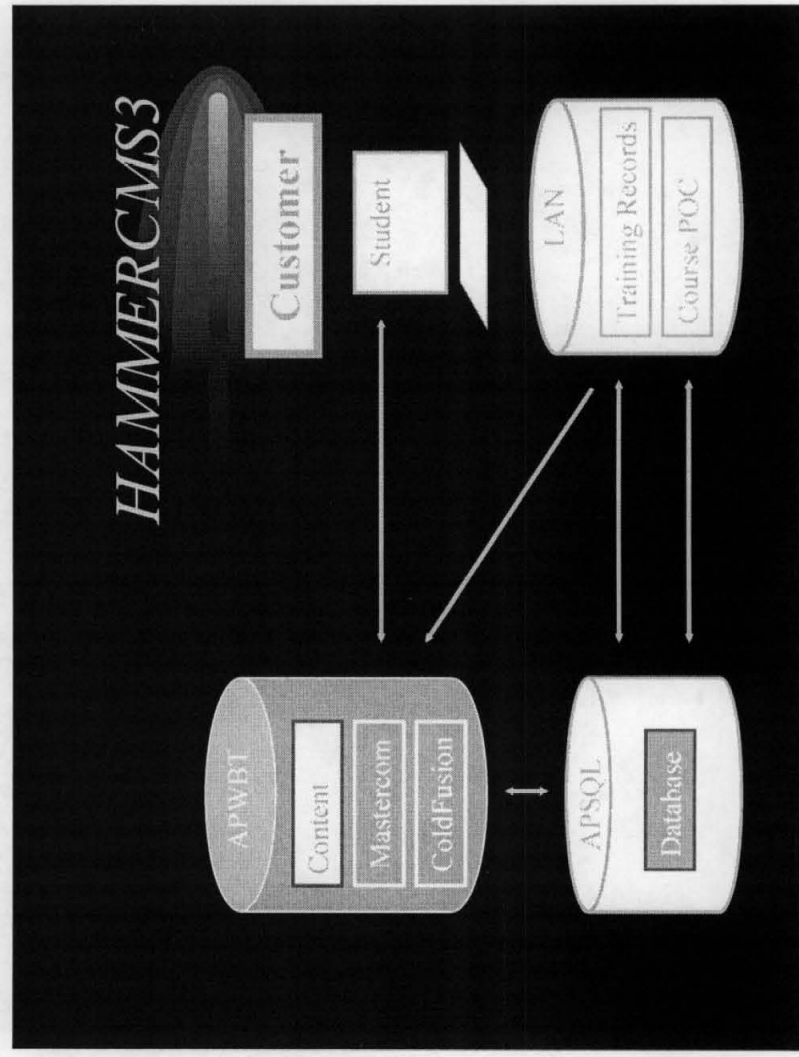
**Powerpoint Slides.**

# HAMMER Courseware Management System

System Development  
and Implementation

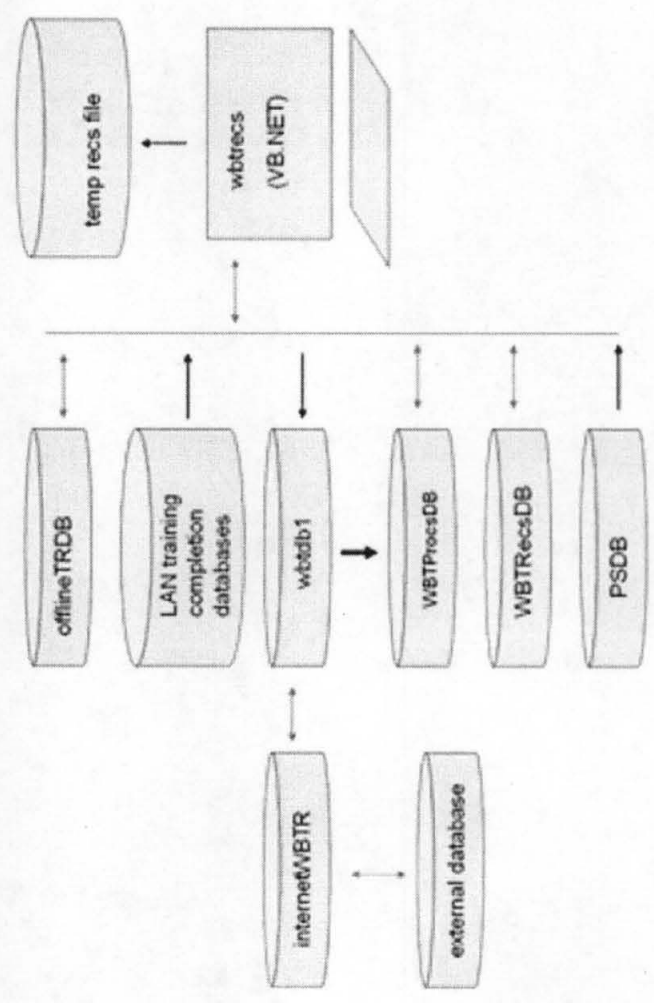


# Early Picture: 5/9/2001



# Recent Picture: 1/6/2006

WBT Records Information Flow



# Documentation!

- Approved Statement of Work
- Project Management Plan
- Requirements Specification
- System Design Description (Table of Contents attached)
- Requirements Traceability Matrix
- Internal/External Interface Agreements
- Test Plan
- Test Cases
- User Acceptance
- Version Design Description
- Y2K Certification
- Findings Resolution
- Configuration Management Plan
- System Security Plan
- Application Risk Assessment

# Most Useful Documentation

Utility depends on context. For this application in context of owner organization developing the system for itself, most useful are:

- System Design Description
- Test Plan

# Coding Considerations

Static Web Pages	HTML
Flashy Web Pages	HTML and JavaScript; more and more in Flash
Question Processing	Javascript to check for input validity; ColdFusion to record response in a SQL database.
Information Display	For screen display: SQL, ColdFusion for data retrieval; HTML, some JavaScript. For text reports for training administrators: SQL, VB6, VB.Net.
Official Retrieval	VB.Net.
Web Developers Interface	HTML, Javascript, Flash templates for content; HTML, JavaScript, SQL, ColdFusion for database interface