ADAPTING THE PERCEPTION ASSESSMENT SERVER FOR MULTIMEDIA STREAMING DELIVERY

Xabier J. Merino Martínez

Promoters: **Dr. Ir. Kris STEENHAUT (VUB-EHB)**

Lic. Peter EBRAERT (EHB)

Brussels, July 2002

Thesis work developed in the context of a Socrates exchange program in partial fulfillment of the requirements for the degree of civil engineering in Telecommunication of Universidad Pública de Navarra. Academic year 2001-2002.

ADAPTING THE PERCEPTION ASSESSMENT SERVER FOR MULTIMEDIA STREAMING DELIVERY

Xabier J. Merino Martínez

Promoters: Dr. Ir. Kris STEENHAUT (VUB-EHB)

Lic. Peter EBRAERT (EHB)

Brussels, July 2002

Thesis work developed in the context of a Socrates exchange program in partial fulfillment of the requirements for the degree of civil engineering in Telecommunication of Universidad Pública de Navarra.

Academic year 2001-2002

ACKNOWLEDGMENTS

I would like to express my most grateful thanks to the following people who have helped me throughout this thesis: Dr. Ir. Kris Steenhaut for accepting being my promoter and getting me in touch with ICTO. My assistant Peter Ebraert for answering all the questions and for his help day after day during all this time. Dr. Sonia Porta for accepting being the Spanish responsible of my thesis and for so many good advises. Jean Pierre Roose for accepting my request of being a Socrates student at his institution. And the rest of members of ICTO: Jan, Arnout, Tim, Lieven and Jess for making me a place at the department and for their help and answers in so many moments.

But this not only the result of my work here in Brussels but the result of seven hard years of work in which I have enjoyed the help of many people I would also like to thank.

Thanks very much to my close family and above all of them thanks to my parents and my brother for their economical and emotional support during every second of these years. Without their help this would not have been possible.

Thanks very much to my girlfriend Yolanda for her help and understanding in every moment throughout these years. The end of this thesis is also the beginning of a new life that we are going to live together.

Thanks to Edurne, Ioseba, Yosune, Miren and Maria, my friends in the university, for all those moments spent. We will forget the bad moments but the good ones will be always kept in our minds.

And finally thanks to the rest of my friends here in Belgium and in Spain for their unfailing support.

ABSTRACT

QuestionMark Perception Server is being used in the *Erasmushogeschool Brussel* for electronic evaluation and examination since several years. That way, electronic facilities can be added to the exams and tests. While conventional paper-made examinations are very limited, with Perception we gain on flexibility and interactivity.

One of these facilities is the possibility to add multimedia content to questions.

This thesis has been developed in order to improve the option of adding multimedia to questions. For doing that, the *Perception Server* and a media server equipped with *Windows Media Services* were coordinated to work together. This coordination allows changing from the standard option of downloading the media files to the new *streaming* technology making the process much more efficient.

The participant of the test will benefit from a reduction in waiting time, and at the same time the traffic on the network will be made less bursty. The author of the test will only have to carry out one simple additional step in the authoring process if multimedia streaming is requested and all this within the same secure environment of *Perception* applications.

SINOPSIS

QuestionMark Perception Server se utiliza desde hace varios años en la Erasmushogeschool Brussel para realizar electrónicamente la escritura y distribución de los exámenes y evaluaciones. De esta forma se pueden añadir a los tests y exámenes diversas facilidades electrónicas que son imposibles de incluir en exámenes convencionales hechos en papel y que dan mayor flexibilidad e interactividad. Una de estas facilidades es la posibilidad de incluir contenido multimedia en las preguntas.

Este proyecto fin de carrera tiene el objetivo de mejorar la opción de añadir multimedia a las preguntas. Para ello, el servidor *Perception* y un servidor multimedia como *Windows Media Services* han sido coordinados para trabajar conjuntamente. La coordinación entre ellos nos permitirá cambiar de la opción estándar que consiste en bajar los archivos multimedia al ordenador local a un sistema que utilice tecnología de *streaming* con lo cual se ganará en eficiencia y rapidez.

Los participantes de los exámenes notarán la reducción en tiempo de espera mientras que al mismo tiempo el tráfico en la red se hace más homogéneo. El autor del examen solo tendrá que ejecutar una acción más en el proceso de realización del examen si quiere adaptarlo al nuevo sistema de *streaming* y todo ello dentro del mismo entorno de seguridad utilizado por *Perception*.

SAMENVATTING

QuestionMark Perception Server wordt in de *Erasmus Hogeschool Brussel* reeds enkel jaren gebruikt voor elektronische evaluatie en examinatie. Elektronische uitbreidingen kunnen aan de proeven en examens toegevoegd worden waardoor we flexibelere testen creëren. Daar waar de klassieke examens op papier beperkt zijn, winnen we nu aan flexibiliteit en interactiviteit.

Eén van die toepassingen is het toevoegen van multimedia aan de vragen. Dat is het onderwerp van dit eindwerk. Om dit mogelijk te maken, moet de *Perception Server samenwerken met* een media server uitgerust met *Windows Media Services*. Deze coördinatie laat ons toe te veranderen van het standaard downloaden van media naar een nieuwe streaming technologie die het proces doeltreffender maakt.

De geëxamineerde bemerkt een verkorte wachttijd, terwijl burstiness van de bezetting van het netwerk sterk verlaagd wordt. De auteur van de test moet slechts één enkele stap toevoegen aan het schrijfproces als multimedia *streaming* gewenst is. Bovendien werkt de toepassing in dezelfde beveiligde omgeving van de *Perception* toepassingen.

TABLE OF CONTENTS

TABLE OF FIGURES

1. INTRODUCTION

QuestionMark Perception [1] is commercial software for electronic writing and online organizing of tests and questionnaires. With Perception you create and deliver tests, quizzes and surveys on intranets or over the Internet.

Perception software provides multiple facilities both for the *author* who writes the test and for the *participant* who takes it. Those facilities range from reporting statistics on the results of a test to the option of including any kind of resource in the questions or providing feedback to each question or at the end of the assessments. Within the functionalities of Perception, there is also an option of adding multimedia content to questions. That is one of the most novel and interesting elements included in the application.

By adding multimedia content you can make questions "come to life" and the participant will really enjoy the difference between conventional paper examinations and Perception based questionnaires.

When using the multimedia facility in the questions together with the web based delivery service of Perception we come across two performance limitations:

- Multimedia files are usually several megabits large so the size of the test will rise significantly when those files are included.

- Traffic and bandwidth availability on the Internet are not guaranteed and fluctuate frequently.

Both characteristics result in the fact that the more multimedia content you add to the test, the longer it will take to download the test. This will make the whole process of taking a test a tedious long waiting process and all the interactivity will be lost.

This thesis proposes coordinating the *Perception Server* and a media server equipped with *Windows Media Services* [2] in order to modify the default multimedia capabilities included in Perception changing from the present file-downloading system to a new multimedia system that takes advantage of the *streaming* technology. That will reduce the waiting time and will also cope with the problem of fluctuating bandwidth over the Internet.

In opposition to the usual "download and play later" systems, the Multimedia Streaming technology offers the advantage of not having to wait until the file has been completely downloaded. The participant only has to wait for a buffer to be filled and from that moment on, it is possible to start playing the media.

The participant of a Perception *assessment* [1] will not notice any difference whether the assessment has multimedia content or not. At the moment of the request of taking the test, only the text content will be downloaded. The multimedia content will start being streamed when the user attempts to play it.

To accomplish this task, part of the Perception Server files will have to be modified and a client-server application will have to be made for both the Perception Server side and the Windows Media server side. That new software will adapt the necessary links and will copy the multimedia content from its initial location to the correct streaming directory.

Only appropriate persons (administrators, teachers) will be able to add the streaming facility to an assessment. Assessments will still be developed through the standard method. That way, the process of adapting the questions to the new streaming delivery method remains completely independent of the process of making them. The new functionality can be implicated to each assessment individually. That way, each instructor can decide whether to add streaming characteristics to his test or not.

[1] The word *assessment* is used in Perception to describe a test, survey, exam or other group of questions developed to be taken by a participant.

2. EXISTING ARCHITECTURE

In this chapter we will take a look at the departmental configuration and network resources of the Erasmushogeschool Brussel as a first step to see the different possibilities to configure the Perception streaming system developed in this thesis.

2.1 Network Configuration

The Erasmushogeschool Brussel is divided in several educational departments spread in different buildings and locations all over the city of Brussels. This special configuration of the institute of high education results in a network architecture composed of several different Local Area Networks connected through the Internet.

The quality of service obtained with this configuration is much lower than a hypothetical WAN configuration obtained interconnecting the different LANs of the departments by means of leased lines. However the prize for the dedicated lines would be so high that it is not worth configure such a system for the only traffic between departments of the institute.

The present configuration results in the fact that the traffic between departments will compete for bandwidth in the Internet in the same conditions than the rest of the traffic from outside the Erasmushogeschool. If the required quality of service is not reached we must develop other lateral actions to improve the service like the one of multimedia streaming proposed in this thesis.

2.2 Perception Server Configuration

The cheapest and most obvious architecture is that only one Perception server will be configured located at ICTO, the department for innovation in education using information technologies. The Perception server and the media server will be member of the ICTO LAN.

There are a high number of departments outside the ICTO LAN [2], so most of the clients that request a Perception assessment will be remote clients. Besides, Perception can also be accessed from remote clients completely outside of the institute LANs. We can consider the following two types of clients:

i) Clients that are inside the same Local Area Network where Perception server is running. They will be called the *Local Clients*.

[2] Actually computers at ICTO form their own LAN. This LAN is connected to a bigger one (referred to here as ICTO LAN) which connects several independent networks in the same campus where ICTO is located and with a unique common Internet connection

ii) Clients that are connecting to the Perception Server from any point of the Internet, by means of an Internet Services Provider or directly from remote LANs. Those will be called *Remote Clients*.

The main difference between them is the bandwidth availability. While the *Local Clients* may have a lot of bandwidth available, the bandwidth on the Internet is not guaranteed and fluctuates continually. Even the type of connection a remote client gets, differs for each client. The objective is to make the difference in making the tests as small as possible for both the local and the remote clients.

When Perception assessments are composed of simple questions (plain HTML [3]), the assessment size is so small that it is not worth to make any extra effort to improve the transfer. Both remote and local clients will probably not notice any difference if they are remotely or locally accessing the server.

The real problem stands when large files (for example multimedia) are included in the questions. In this case the differences will become more apparent. Remote clients will have to wait until the file downloading has completed. If the file has to pass through the Internet, the waiting time can be so long that the use of the system becomes annoying. On the other hand, local clients won't probably have that problem, because the files will only have to be moved between two stations of the same local network where the bandwidth is much higher.

To reduce the differences between local and remote clients this thesis proposes the coordination between the Perception server and a media server to change the delivering technology into a more efficient one (streaming technology). This way, the Perception server will not be responsible anymore for serving media files so the capacity of serving the "text" part of the questions will be also increased.

2.3 Optimal configuration

The ideal and optimum solution would be to place one media server in each institute LAN. All of them would have to be coordinated with the Perception Server. That way, from the media server point of view, all clients from the institute will be local, because they are always requesting media files to the media server placed inside their networks. There could be also a different media server to serve files to clients connecting from the Internet, that is, for the real remote clients. An overview of this configuration can be seen on figure 2.1.

As it can be seen in the figure, the LAN configurations of routers and other network elements are not defined specifically due to the fact that each department of the institute of high education has its own LAN configured in a different way, and the networks represented are only a generalization of those.

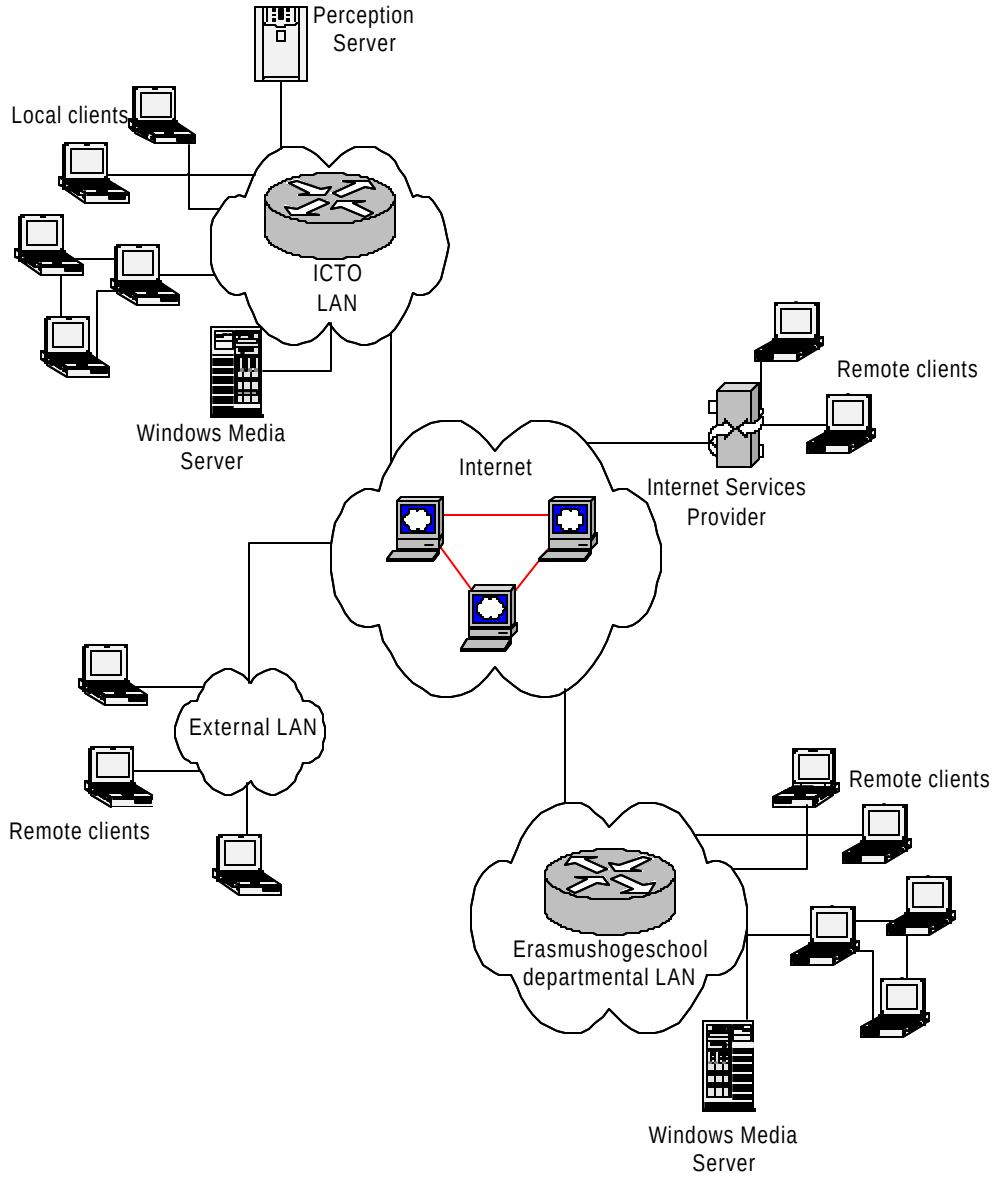

Figure 2.1 Optimal Configuration

To carry out this configuration several, points have to be considered:

- The media servers will have to be synchronized with the Perception server at one given moment.

- The references (links) to the media files have to be made at the time the assessment is requested so that they point to the correct media server depending on the location of the client, or we can assume that each assessment belongs to a different department (normally each exam or exercise is made for a determined class or group of one department) so the references can be made to the correct media server from the beginning.

- Not all the multimedia files are going to be used by all the media servers, so it would be more efficient if they are copied as they are being requested. This way each media server only has the media files that have been requested at least once and are likely to be requested more.

- Maybe it is not necessary to place a Media Server in every location of the Erasmushogeschool Brussel because in some of the locations the traffic is going to be quite low. The requests from that locations will have to be directed to one of the others media servers.

2.4 Proposed configuration

The aim of this thesis is to perform the coordination between the media server and the Perception server independently from the final number of media servers or the final architecture adopted. We will focus on the cooperation between one Perception server and one media server. The media server will be placed in the ICTO LAN together with the Perception server.

In order to perform the coordination let us first take a look at the Perception server and Windows Media Services characteristics.

3. PERCEPTION STANDARD CONFIGURATION

3.1 General Structure of Perception

Using Perception involves the following basic stages:

- *Authoring:* This is the process of composing a structured bank of questions and then selecting appropriate ones for assessments that are given to participants.

- *Publishing:* When assessments have been constructed from questions, they must be copied to a database (together with any associated files) from which they can either be delivered by a web server, or used by participants who are not web enabled.

- *Delivery:* Participants can now take the assessments, either by accessing a URL (which points to a server program) from within their web browser (if they are web enabled), or by using a special Windows program (if they are not).

- *Reporting:* When the participants have taken their assessments, Perception offers two tools for reporting and analysis of the results: Enterprise Reporter and Windows reporter.

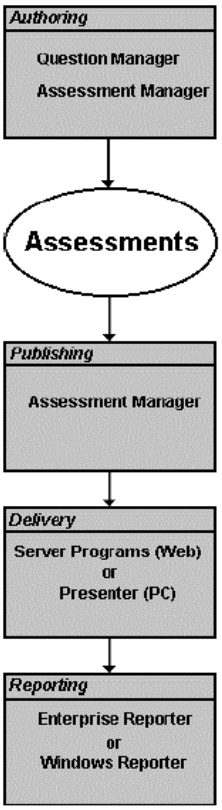

Figure 3.1 Perception System

The authoring stage consists of the following activities:

- Composing Questions: Using the *Question Manager* (Windows Application) it is possible to create questions and organize them as a tree structure of topics containing questions, subtopics or both. The questions are stored in a question database that is used later as a library for building assessments.

- Building Assessments: When the questions are ready, *Assessment Manager* enables them to be selected for inclusion into assessments in a variety of ways. You can also define which questions the participant sees, and in what order.

Question Manager and Assessment Manager are Windows applications, and can be installed in any computer independently of the server. But it is also possible to use special versions of them from within a web browser. These versions are referred to as *Browser Based Question Manager* and *Browser Based Assessment Manager*; this method of authoring is called *Browser Based Authoring*.

When the author has constructed the assessments, the next stage is publishing them with the Assessment Manager. Authors specify a production database [3] into which the assessments from their development database [4] are to be copied, together with other resources such as graphics and templates.

Perception includes the browser based Enterprise Manager which provides an integrated environment for security, administration and reporting. Within Enterprise Manager you can access and use all the administrator functionalities provided by Perception in a secure way: Browser Based Question Manager, Browser Based Assessment Manager, Reporting Manager and System Administration. Which topics one is able to access, depends on the kind of administration privileges that person has.

3.2 Adding multimedia to questions

It is not the purpose of this thesis to explain all the Perception authoring options, but it is essential to explain the adding of multimedia to questions facility. Up to now, the only possible way to add multimedia to questions is through the desktop Question Manager. The browser-based model of this tool still does not include that option, but it is supposed to be included in later editions.

Adding multimedia to a question is a quite straightforward process. Once you have written the text part of the question, from the Question Editor window, select Question – Add Content – Multimedia and the screen depicted in figure 3.2 will be shown.

[3] Production databases are the databases defined in the server side of the system.

[4] Development databases are local databases used to store questions and assessments during the authoring process. Further information on these topics can be found in paragraph 3.4.

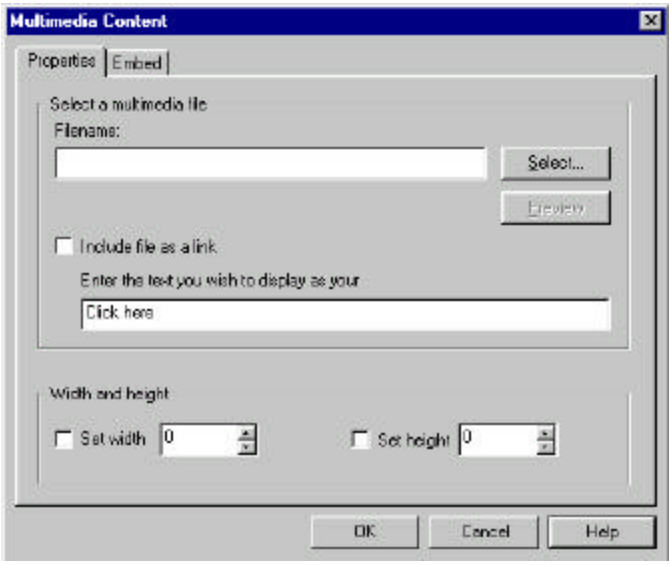

Figure 3.2 Adding Multimedia Content

By clicking **Select** you can choose the multimedia file you want to insert.

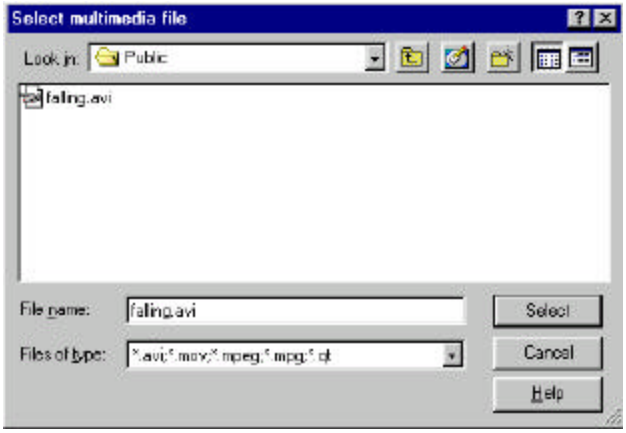

Figure 3.3 Selecting a multimedia file for content

It is possible to set the height and width in pixels of the selected multimedia file using **Set width** and **Set height.**

There are two ways of inserting a multimedia object, one of them is to have a text linked to the object and the other is to embed the object so that it is part of the web page.

If you select **Include file as a link** then you can also enter the text for the link, “click here” is the default text.

If Include file as a link is deselected then the controls on the Embed tab are active. Using this tab the multimedia object is physically embedded into the web page. It also allows you to control several options like to play repeatedly the object, to automatically start playing the object or if a sound file is used to hide the multimedia object.

3.3 Building and publishing the assessments

Both the browser based Assessment Manager and the desktop versions allow us to build the assessments. Building an assessment consists in selecting questions from the question databases and including them inside one questionnaire.

When using the desktop version of the Assessment Manager, we work with local databases. After building the assessment it must be copied to the server databases in order to enable the web access to it. That step is known as the publishing. If we are using the browser based versions of Question Manager and Assessment Manager, the publishing step is avoided because we directly work with the server databases.

There are two possible ways to publish the assessments:

Publish to *Microsoft Access* databases.
Publish to other databases via the ODBC [4] mechanism.

Your License file determines whether you can use only the first option or both.

If you are only allowed to publish to *Microsoft Access* databases, then you have to work directly on the server machine selecting the correct database:

Question Database: …./perception3/server/databases/masterq.question
Assessment Database: …./perception3/server/databases/mastera.assessment

These are the default databases defined but can be changed using the Enterprise Manager.

If you are allowed to use other database engines, then it is possible to publish remotely using the ODBC mechanism. For ODBC publishing, you must select DSN names instead of database file names. DSN names are defined using the Enterprise Manager – Server Settings option. In figure 3.4 we can see the DSN names for the default server databases.

The multimedia files must be published to the server's multimedia directory. The default multimedia directory is:

Multimedia directory: …./perception3/server/resources/multimedia

That default folder can also be changed using the Enterprise Manager.

However Perception doesn't provide a solution for copying the multimedia files in the multimedia folder from a remote location. Therefore, they must be copied manually to the server, that is, an administrator has to be given the media files included in the assessments and copy them in the resources-multimedia folder working directly on the server machine.

It is possible to automate this task configuring for example an FTP [5] server or writing a specific application.

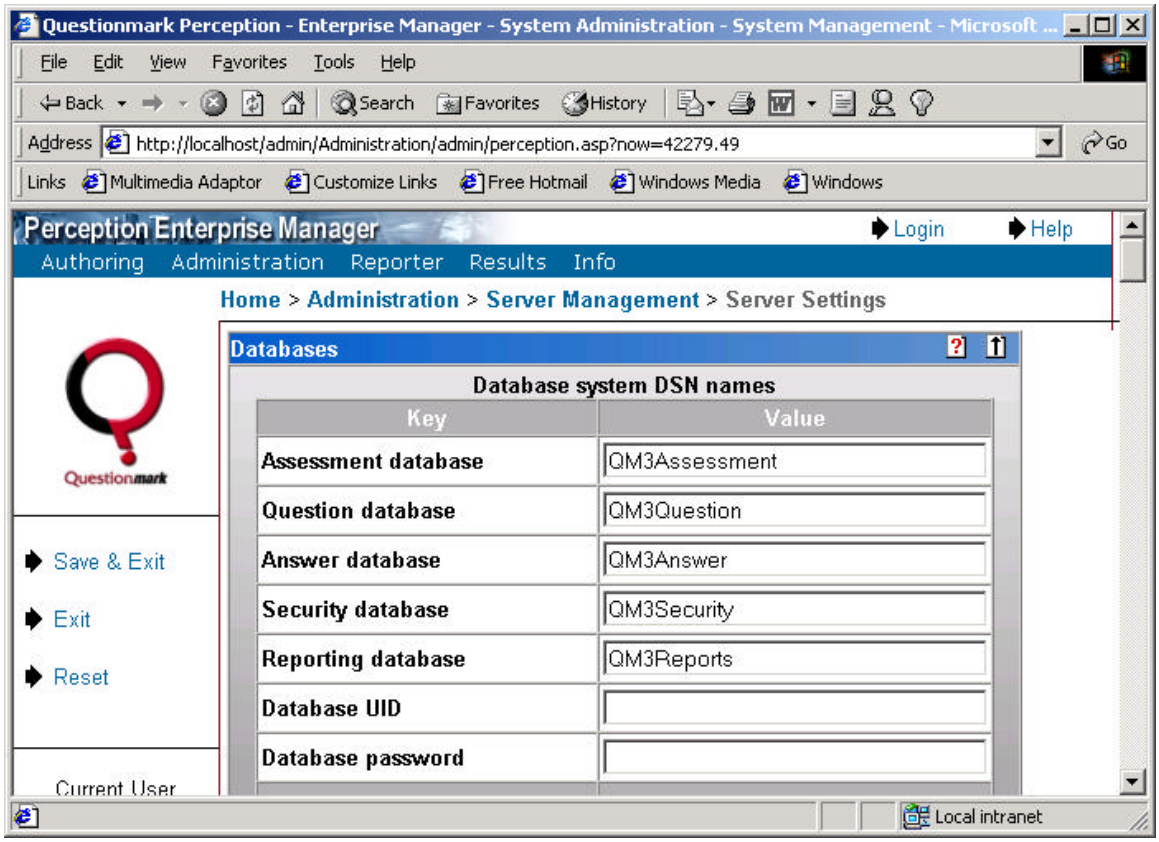

Figure 3.4 DSN names for the default Perception server databases

Figure 3.5 explains all the possibilities for authoring and publishing provided by Perception.

3.4 How information is stored in Perception

It is possible to differentiate between two different groups of databases in the Perception system: databases used by the server system and databases used by the authoring system.

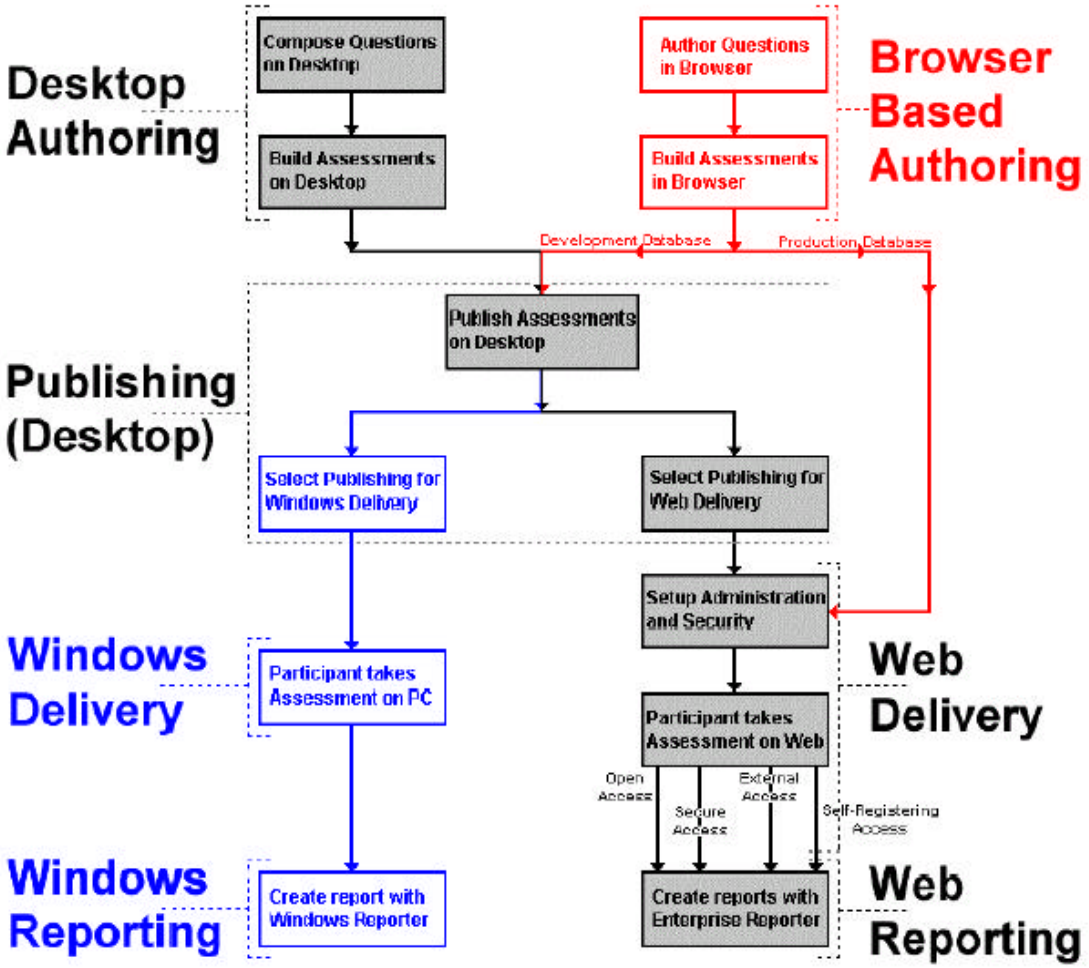

Figure 3.5 Authoring and publishing possibilities with Perception

In the authoring system two kinds of databases are used: assessment databases and question databases. These databases have respectively the extensions .assessment and .question. You can define in your authoring process all the question or assessments databases you want in order to organize your work in the best way. Those databases are referred to as *development databases*.

The server system uses only one question database and one assessment database. Besides them, three other databases are used. One is for administration, another is for reporting the results and the last one is for storing the answers given by the participants to the assessments.

Question and assessment databases in the server system are the ones accessed when publishing your assessment for web delivery. They are referred to as *production databases*.

Perception accesses its databases via the ODBC [4] technology, so both the server and the authoring machines must have the correct ODBC drivers installed. Depending on which database format you are using with Perception you will need to ensure that you have the appropriate ODBC drivers[5].

We are interested in the structure of question and assessment databases because they are the ones that are going to be consulted to obtain information and to modify the appropriate entrances when necessary. In the next three paragraphs we will focus in the interesting records of those databases and we will see how given an assessment name it is possible to find the questions included on it, procedure that will be used later in this thesis.

3.4.1 Question Databases

Question databases contain seven different tables to store all the different information about questions needed by the system. The most important table is the *Q_QML* table where the text of the question is stored in the column field known as *QML string*.

Questions are not stored in HTML format but in a special language developed by Perception called QML[6] [6].

When a perception assessment is requested through the Internet, HTML pages are constructed on the fly. The system looks for the QML String in the server question database and then the page is constructed translating the QML language into HTML. For example when a participant requests an assessment, the system will check first if the participant is allowed to access that assessment. Then it will search for the questions included in the assessment. At that time, the QML string of each question will be translated to HTML and then will be sent to the client's browser.

In the same table we can also find two numbers stored in each row. These are called *Question_MID* and *Question_LID*. These numbers are respectively the top eight numerals and the bottom eight numerals of the *Question ID*, a randomly generated number that uniquely identifies each question in all the Perception system. Also the name of the editor, the last modification date and some other less interesting fields are stored in the table.

A view on the Q_QML table can be seen in figure 3.6.

[5] MS Access Version 4.00.6019.00 of odbcjt32.dll or later
SQL Server ODBC Driver Version 3.70 or later
Oracle Microsoft ODBC for Oracle Driver Version 2.573 or later
(Do not use the Oracle Corporation Driver)

[6] QML stands for Question Markup Language.

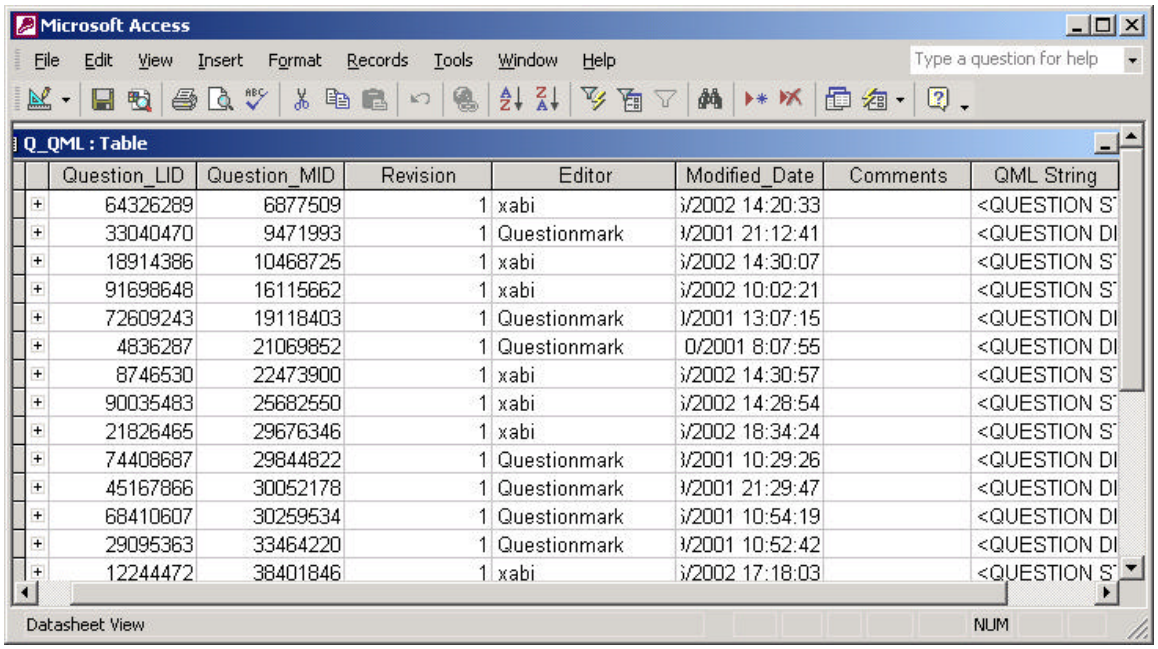

Figure 3.6 Q_QML table of a question database.

We are also interested in the *Q_Question* table where all the Question_ID numbers are related with the corresponding topic to which the question comes belongs. The topics are defined by their Topic_ID number which is also a random number given by Perception to uniquely identify the topic throughout the system. Figure 3.7 shows an example of this table.

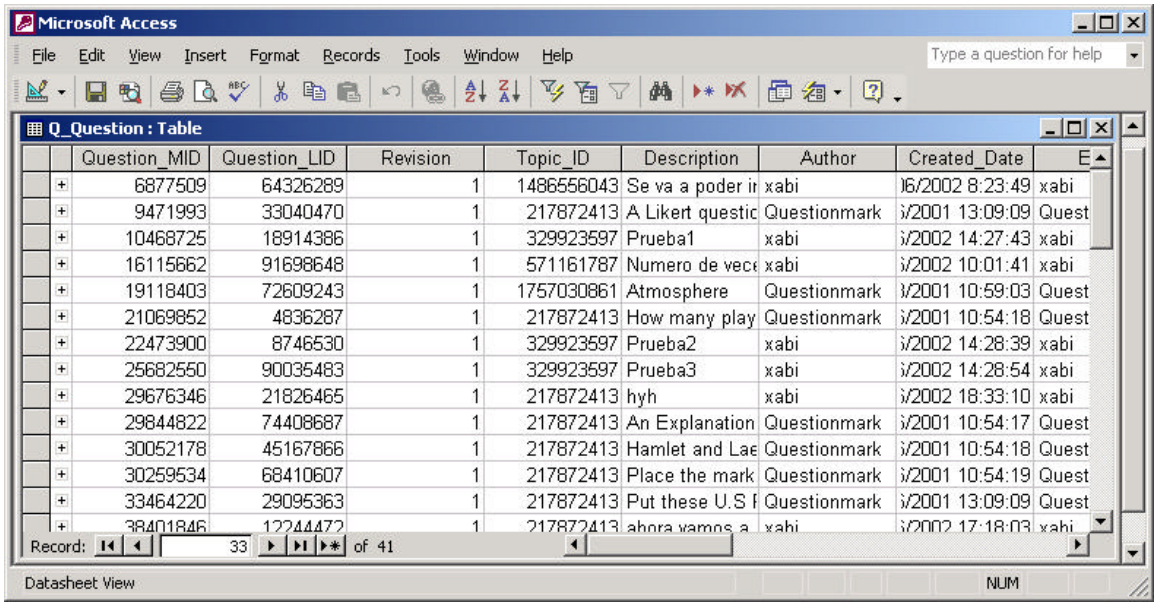

Figure 3.7 Q_Question table from a question database

3.4.2 Assessment Databases

Each assessment database consists of nine tables. For our purposes only the *S_Header*, *S_Index* and *S_Item* tables are interesting.

In the *S_Header* table, the name of the assessment [7] and the *Session_LID* and *Session_MID* numbers are stored. As their correspondents for questions, *Session_LID* and *Session_MID* are the top and bottom parts of the *Session_ID*. It is a random number which uniquely identifies the assessment in the system. An example of this table is presented in figure 3.8.

Figure 3.9 shows the *S_Index* table. We can see how several numbers are assigned to each *Session_MID* and *Session_LID* defined in the preceding table. These numbers are called *Block_ID*, *Block_Order* and *Block_Type*. This table indexes the *Session_ID*s to question blocks defined by the system.

We are only interested in the *Block_ID*s of each *Session_ID* whose correspondent *Block_Type* has the value 2 because is the one which refers to the questions that are included in the assessments as we will see in the S_Item table.

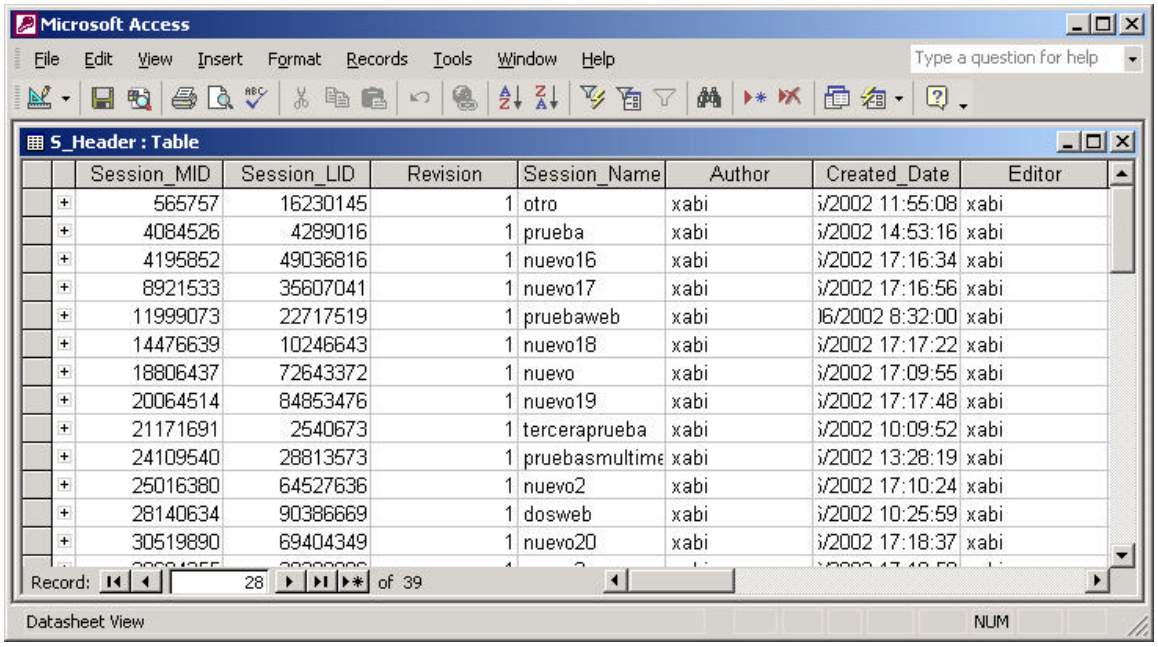

Figure 3.8 S_Header table of an assessment database

[7] Each assessment is given a name by the author when built. Throughout the Perception system this name is generically known as the Session name.

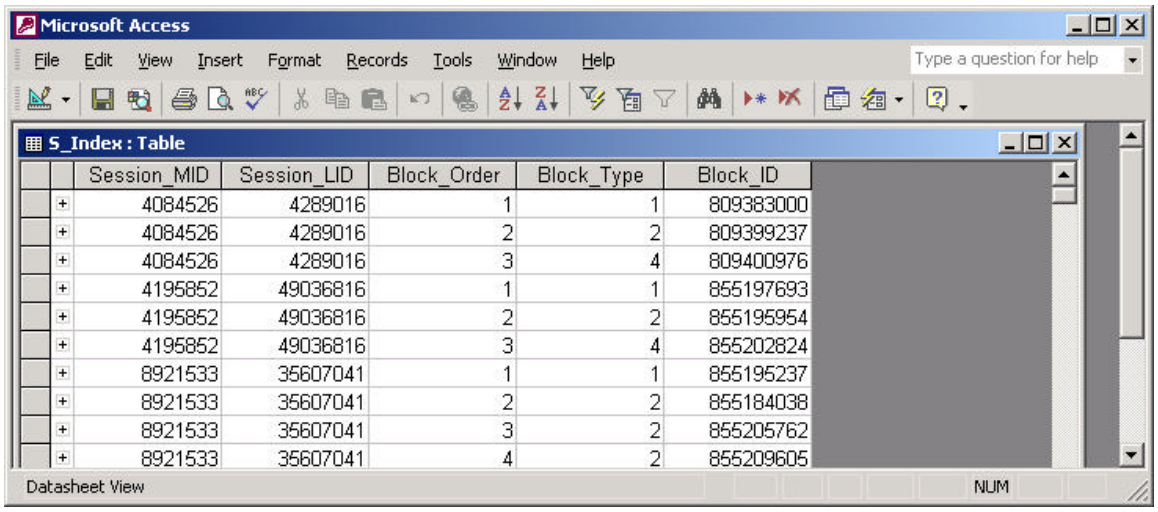

Figure 3.9 S_Index table of an assessment database

Finally, in the *S_Item* table we can see that the Block IDs are linked to other values:

- Method: Column with three possible values. 0 if the assessment is composed of several random questions from one topic. 1 if the assessment includes all the questions from one topic and 2 if only selected questions from one determined topic have been included in the assessment. 3 is also possible, but will not be considered because is hardly ever used.

- Include Subtopics: Selected if questions from subtopics of the main topic are included in the assessments.

- Topic_ID: Indicates the ID of the topic from which questions in the assessment come from. If the assessment contains questions from several topics, that information is stored in a different row of this table.

- Number of Questions: It contains the number of random questions included in the assessment if the method value is 0. Its value will be 0 in case that the method value is 1 or 2.

- Question_MID and Question_LID: Contains the question numbers in case the method value is 2 what means that only selected questions have been included in the assessments. On the other hand, the value of these columns will be 0 in case the method value is 0 or 1, that is, in case that a random group of questions or all the questions from a topic have been included.

An example of the S_Item table can be seen on figure 3.10.

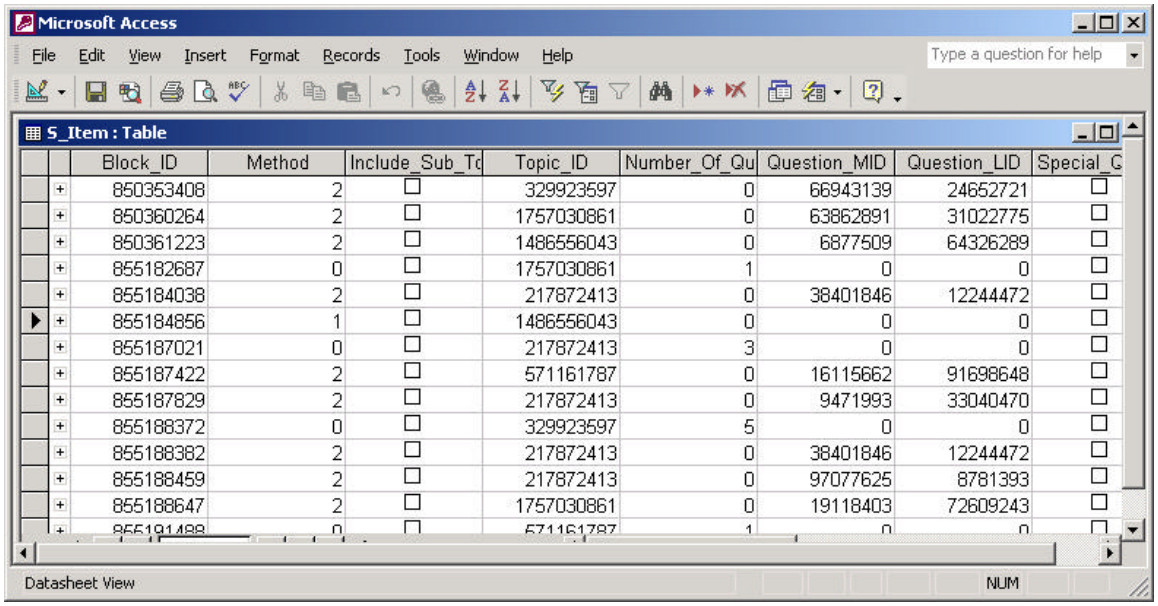

Figure 3.10 S_Item table of an assessment database

3.4.3 Finding the Questions

Later in this thesis we will be interested in finding the questions included in an assessment given the assessment name. Using the information provided in paragraphs 3.4.1 and 3.4.2 we can achieve this aim.

Let us suppose that we want to obtain the questions included in an assessment called *assessment_name*.

1. The first step is to find the *Session_ID* (*Session_MID* and *Session_LID*) corresponding to that assessment_name. This information can be easily obtained from the *S_Header* table in the assessment database. In that table, the assessment names are referred to as *Session_Name*.

2. Once the *Session_ID* has been obtained, we go to the *S_Index* table. From that table we have to get all the *Block_ID* records which correspond to the previously obtained *Session_ID* and only those whose *Block_Type* value is 2.

3. With the values obtained we go to the *S_Item* table and we search for all the records in that table that contain those values of *Block_ID*. Then, it depends on the value of the *method* column which actions we will further undertake:

4.1 If the method column value is 0: In that case, we have to take the value of the *Topic_ID* and the value of the *Number_Of_Questions* columns, because the assessment consists in *Number_Of_Questions* random questions from that topic.

4.2 If the method column value is 1: Then we have to take the value of the Topic_ID again because all the questions included in that topic have been included in the assessment.

4.3 If the method column value is 2: Then we can take the *Question_ID* (*Question_LID* and *Question_MID*) because that is the number of one of the questions included in the assessment.

5.1 Now, if we have obtained *Question_ID* numbers from step 4, we can directly go to the Q_QML table in the question database. There we will find the corresponding QML string for that identification number.

5.2 If we have obtained a *Topic_ID* from step 4 we must first check the Question_IDs included in that topic in the Q_Question table of the question database.

We can see a block diagram explaining the above procedure in figure 3.11.

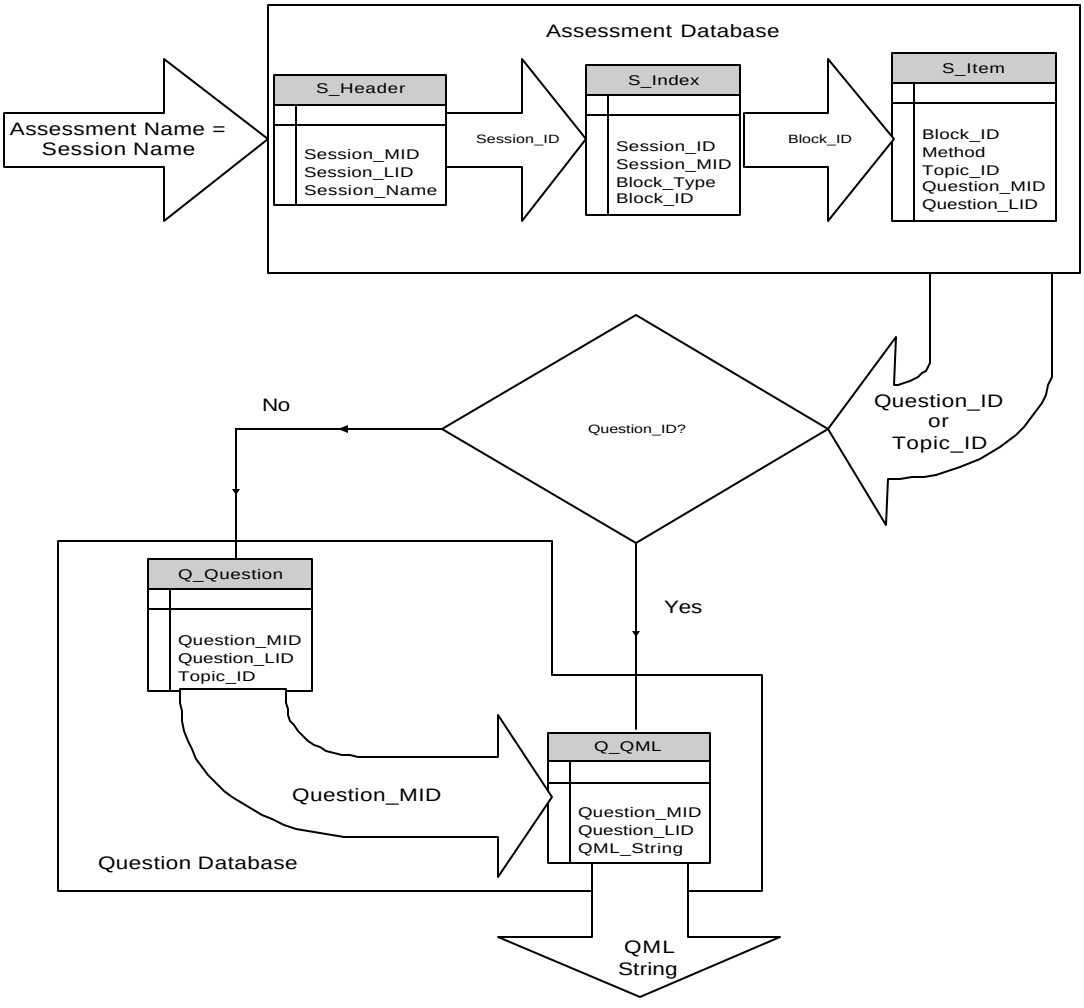

Figure 3.11 Obtaining the QML string from the assessment name

3.5 Question Markup Language

The QML language is a platform independent language to describe questions for use in tests and assessments. QML is defined by Question Mark Computing Ltd, but is designed as an extensible language that the whole computer assisted assessment community can use. QML is used as the native language for describing questions in Perception.

QML works on the same general principle as HTML in terms of a file containing text with tags to define specific meanings. The general structure of a tag is:

<TAGNAME ATTRIBUTE>TAG CONTENTS</TAGNAME>

Attributes are defined in the form NAME=NUMBER, NAME=WORD or NAME="STRING".
Tags must be explicitly closed with a matching closing tag, e.g. <TAGNAME>…</TAGNAME>, in order to allow a wider variety of characters within the tags.

QML is extensible, able to incorporate content and question types not yet envisaged, by means of the principles:

- All elements will contain information about their type to enable their interpretation by different implementations.
- All elements can contain used defined attribute tags which will be ignored by implementations which do not understand them.

The extensibility characteristic of QML will be used later in this thesis.

3.6 What does Perception do when multimedia is included?

In the Options Settings dialog of the Question Manager it is possible to define the multimedia directory that will be used for the author system. The default option is ../perception3/author/resources/multimedia.

When a multimedia file is selected to be included in a question (as explained in paragraph 3.2), is automatically copied to the multimedia directory and a reference to that file is included in the QML string of that question. That reference consists of a tag indicating that the content type is a multimedia file and the name of the referenced file.

When publishing the file to the server databases the used multimedia resources must also be copied to the server resources folder, because when the server will translate the question from QML into HTML the link inserted to the multimedia file will be a link to the server resources directory and of course not to the author one. The server resource directory is by default …./perception3/server/resources/multimedia but can also be changed using the Enterprise Manager.

4. WINDOWS MEDIA SERVICES

4.1 Streaming Options

Windows Media Services is Microsoft's solution for delivering media content over a network in one of two possible ways:

- Unicast: Is a point to point connection between the client and the server. Each client receives a distinct stream from the server. Unicast streams can be distributed in one of two ways:
  - ♦ On-demand: Active connection where the user initiates the client connection to the server by selecting the content item. An example of an on-demand unicast is when a user requests a stored file. The client connects to the server to receive a specific piece of content and the content is streamed only to a single client. The client uses the name of the server and the name of the file to identify the URL it has to connect to.
  - ♦ Broadcast: Passive experience of receiving a stream. The client receives the stream but does not control it. It is similar in some way to a TV broadcast where you can turn on or turn off the receiver but you cannot control the content being broadcasted.
- Multicast: A multicast streaming is a one-to-many connection in which multiple clients can receive the same stream from a server. In multicast the client receives the stream by monitoring a specific IP address. Multicast preserves network bandwidth and can be extremely useful for low bandwidth local area networks. To perform a multicast delivering we need a multicast-enabled network, that is, a network that has routers that can interpret class D IP [7] addresses.

In unicast streaming a separate copy of the data is sent from the source to each client building a different connection. When the same data needs to be sent to only a portion of the clients this method wastes a lot of network bandwidth by sending multiple copies of the data and building particular connections for each client.

In contrast, multicast sends a single copy of the data to those clients who request it, no multiple copies are sent. We can see the clear difference in network load per client when unicasting an 8-Kbps PCM audio stream and multicasting the stream in the following graph.

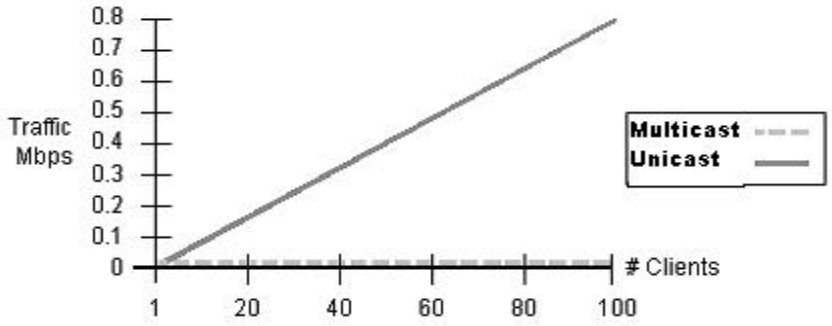

Figure 4.1 Unicast vs. Multicast traffic comparison

When using multicast streaming, several clients must want to access the same data at the same time. This could be our case when the students are having an exam, they will probably access the multimedia file at the same time or at least they will access it in groups. If we configure the system in that way, we will only have one connection for all the students instead of one connection for each student. In terms of network load and efficiency that would mean a great step forward.

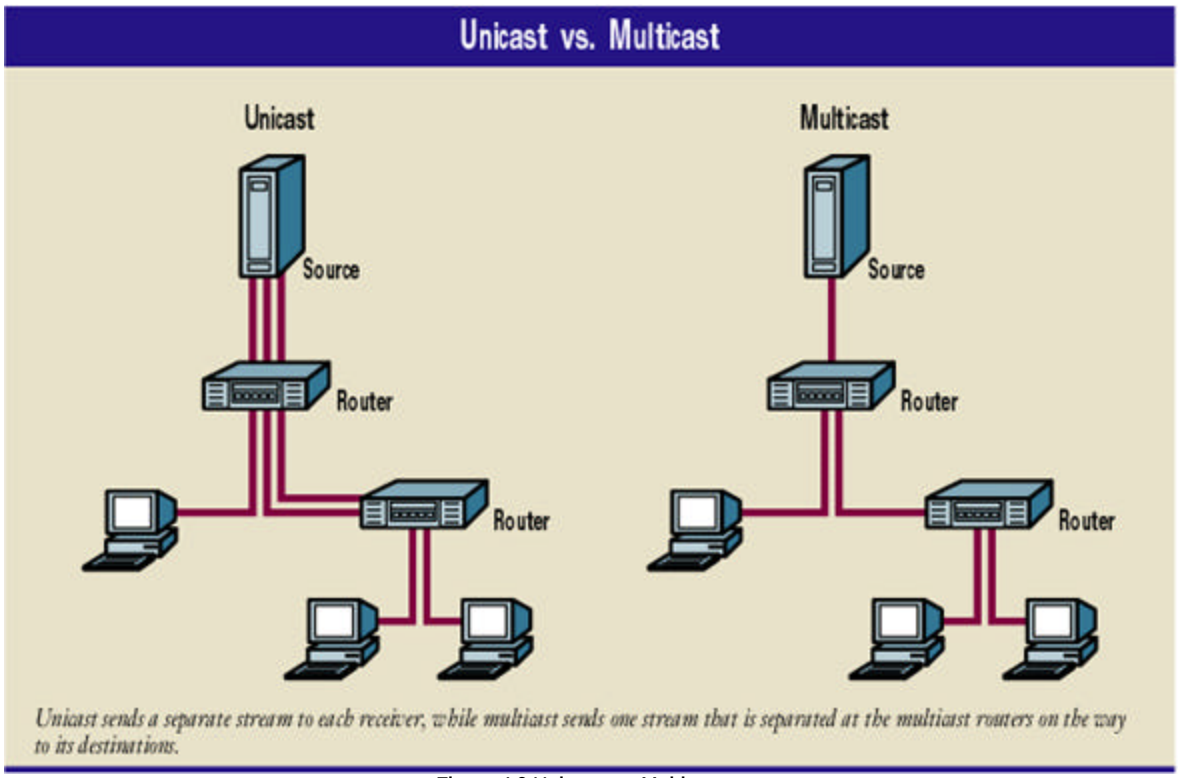

Figure 4.2 Unicast vs. Multicast

4.2 Multimedia formats

Windows Media Services offers the ability to provide multimedia content to a large number of clients using ASF, .wma, and .MP3 formats.

ASF (the Advanced Streaming Format) is the suggested format for streaming. If one chooses to stream .wav or .MP3 files, the server performance can be significantly affected. ASF is an open standard used to arrange, organize and synchronize multimedia data for streaming over networks. While ASF is optimized for sending multimedia streams over networks it is equally suited for local playback as well. Any compression-decompression algorithm (codec) can be used to encode ASF streams. Information stored in the ASF stream can be used to aid the client in determining which codec is needed to decompress the stream. In addition, ASF streams can be carried over any underlying network transmission protocol.

Windows Media Audio (.wma) is an audio-only ASF file compressed with the Windows Media Audio codec. This type of media differs from .asf files only in its extension.

It is possible to transcode all the other types of multimedia files to .asf files using the Windows Media Encoder.

4.3 Providing content to a client

Windows Media Services uses the terms publishing points and stations when describing how the server provides content to a client.

- Publishing point: Is used to access unicast content. A publishing point is a virtual directory that stores content you have made available to clients on your server. There are two possible types of publishing points: on-demand unicast publishing points used to provide .asf files and broadcast unicast publishing points used to provide live ASF streams. It is possible to set limits on the number of clients that can access the publishing point at a given time and on the amount of bandwidth that can be used by that publishing point. These limits are useful in managing the amount of bandwidth that the publishing point consumes.

- Station: Is used to access multicast content. Because stations are multicast, the stream is shared among all clients that want to access it. The number of clients has no impact on the network or the stream. Thus, there is no need to limit the number of clients that can access the streams.

4.4 Windows Media Services protocols

To better understand the Windows Media Services protocols let us take a closer look at figure 4.3. It depicts how protocols are used to communicate between the components of a Windows Media Services system.

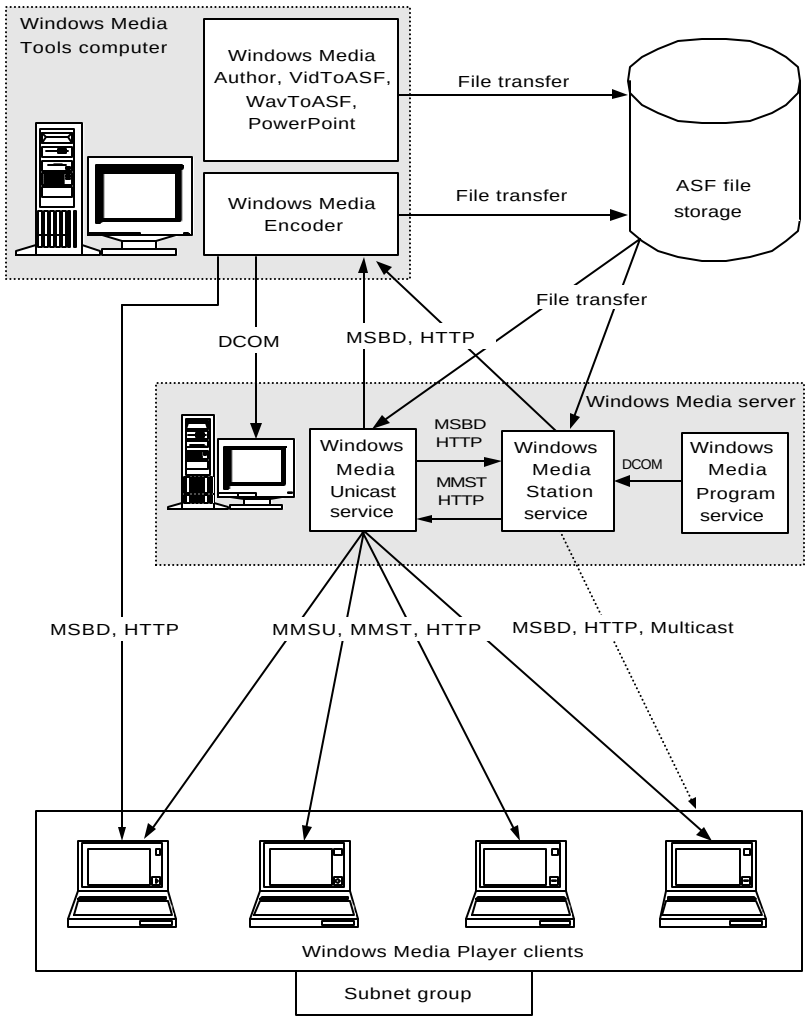

Figure 4.3 Windows Media Services protocols

The first important feature is that it is possible to connect all of the components via HTTP, allowing them to communicate even if they are separated by a firewall.

Clients that connect to a multicast do not use a protocol; they receive the data as it streams over the multicast IP address and do not need to negotiate a connection.

The MMS protocol is used to access unicast content from a Windows Media publishing point. MMS is the default method of connecting to the Windows Media Unicast service. When connecting using MMS, protocol rollover is used to get the best connection. Protocol rollover starts by trying to connect the client to the server via MMSU. MMSU is the MMS protocol combined with UDP data transport. If the MMSU connection is unsuccessful, then the server attempts to use MMST. MMST is the MMS protocol combined with TCP data transport. If we are connecting to an indexed .asf file and want to be able to fast-forward, rewind, pause, start, and stop the stream, you must use MMS.

When you connect to a publishing point from a stand-alone Windows Media Player, you must specify the URL to the unicast content. If the content is being published on-demand over the Home publishing point, the URL is composed of the server name and .asf file name, for example:

mms://windows_media_server/file.asf

The MSBD protocol is used to distribute streams between Windows Media Encoder and the Windows Media server components, and to transfer streams between servers. MSBD is a connection-oriented protocol optimized for use with streaming media.

It is also possible to use the HTTP [8] protocol to stream content. Using HTTP streaming helps to overcome firewalls as obstacles because most firewalls allow HTTP to pass through[8]. HTTP streaming can also be used to stream content from Windows Media Encoder through a firewall to the Windows Media server and can be used to connect Windows Media servers that are separated by a firewall. We have to be sure there is no conflict on port 80 in case we are running Windows Media Services on the same computer as a Web server.

If an .asf file is published using the MMS protocol, protocol rollover is automatic from MMSU to MMST and finally to HTTP. Windows Media Player tries each protocol in turn until a connection to the source is made.

Windows Media Player performs this type of rollover automatically when MMS protocol is specified as the unicast publishing point is created.

URL rollover can also be used to specify different Windows Media servers that contain the same content. For example, if the first REF tag specifies an .asf file on a server called server 1 and the second REF tag specifies a copy of the file on server 2, Windows Media Player can reach the file using either server.

4.5 Client software

Windows Media Player is the client software provided by Windows Media Services and is used to receive and render streams from a Windows Media Server.

[8] On the other hand, firewalls are not usually configured to allow MMS to pass through.

Windows Media Services uses Windows Media Player to render .asf streams that can include video content, audio content and images. For a client to receive multicast file transfers, Nsfile.ocx must be configured and installed on the client computer. Nsfile.ocx is not installed as part of Windows Media Player.

Clients are available for the Microsoft Windows, Apple Macintosh and UNIX operating systems.

4.6 Windows media server and a web server on the same computer

Windows Media server and a web server can coexist on a computer when you use their default values that is if the protocol used for streaming is the MMS which uses port 1755.

If you want to stream content via HTTP (for clients outside a firewall) then the Windows Media server must have an IP address available on port 80.

To use HTTP streaming when Windows Media server components and IIS[9] are installed on the same computer, you require the following:

- At least two IP addresses bound to your network card.
- Unique Domain Name Server (DNS). A resource records for the Windows Media server IP address and the IIS server IP address.

[9] IIS = Internet Information Services

5. NEW PERCEPTION ARCHITECTURE

Once we have overviewed the characteristics of the Perception Server and Windows Media Services the next step is to decide which streaming option of the ones explained in previous chapter is finally chosen. After doing that let us start explaining how to adapt the system to include the streaming functionality, that is, how we can manage to perform the coordination between the Perception and Windows Media servers.

5.1 Selecting the streaming option

As explained, most part of the Internet is not multicast enabled yet. In chapter 2 we saw that most of the clients calling the Perception Server and the media server for assessments will connect to the system through the Internet. So it is nonsense to configure our system in that way if part of the clients are not going to be able to access its contents.

Even in the case we could use multicast streaming (because the network configuration would permit it) we still would have to solve the problem of synchronization between clients asking for the same media file. They are not going to ask for it exactly at the same time, so if the multicast has already started for another client, the new ones joining in the meanwhile would receive an incomplete stream.

Those reasons are enough to choose the on-demand unicast streaming as the streaming option for our system.

Although the network efficiency is not so high as in the multicast case it will create a different connection for each client so they can access the data at the time they want. Also working with optimized streaming files like .asf the results obtained with unicast streaming are quite good in most cases.

5.2 Creating a new On-demand Publishing Point

It is quite straightforward to define a publishing point in the Windows Media Server. By running the Windows Media Administrator we select Unicast Publishing Points and then a new window is presented with the Publishing Points already existing both On Demand and Broadcast. Clicking the On-demand button we select new and a wizard appears which will help us through the process. Following the different screens of the wizard you will be asked about the final directory where the media files are going to be stored and some other questions about aliases for the publishing point and similar. After answering that questions the publishing point is created.

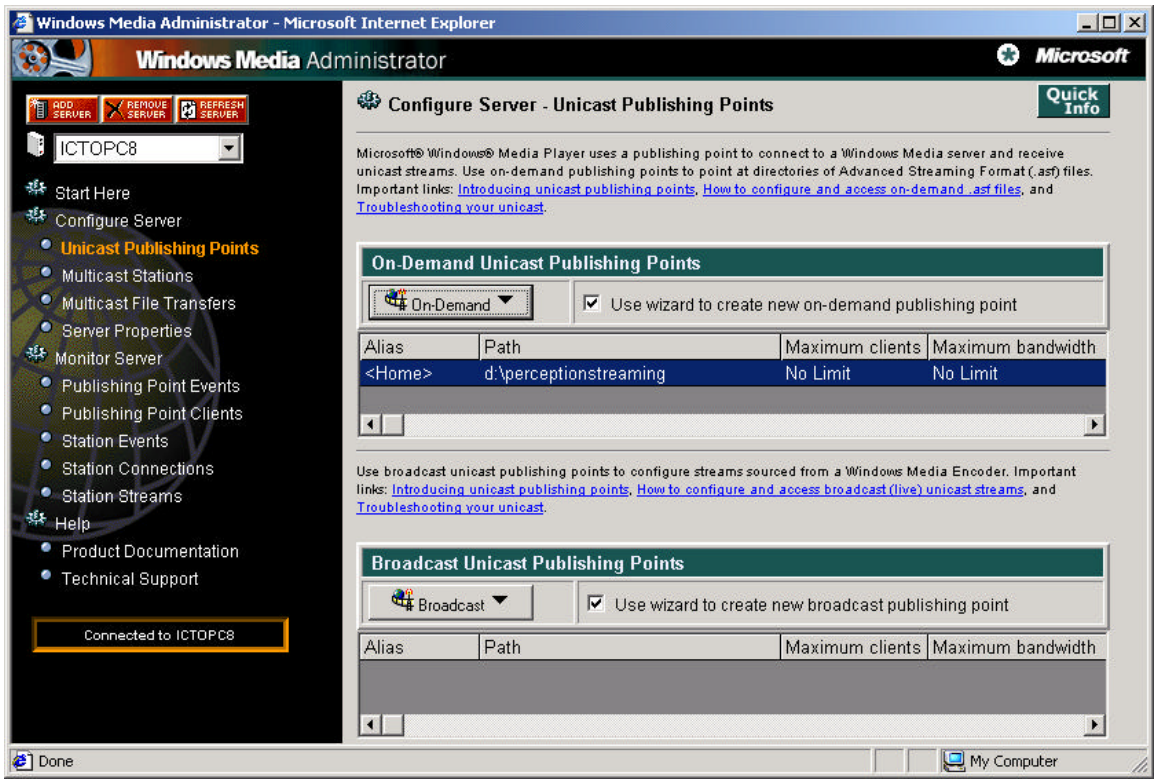

Figure 5.1 Windows Media Administrator

5.3 How to redirect the links to the Windows Media Server

Figures 5.2 and 5.3 are examples of the QML format of two questions one including an embedded multimedia file and the other one including a linked media file.

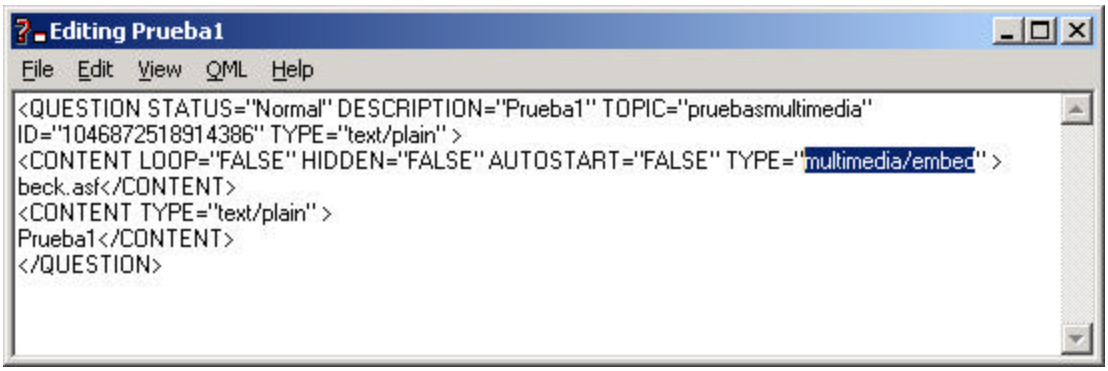

Figure 5.2 QML string of a question including an embedded file.

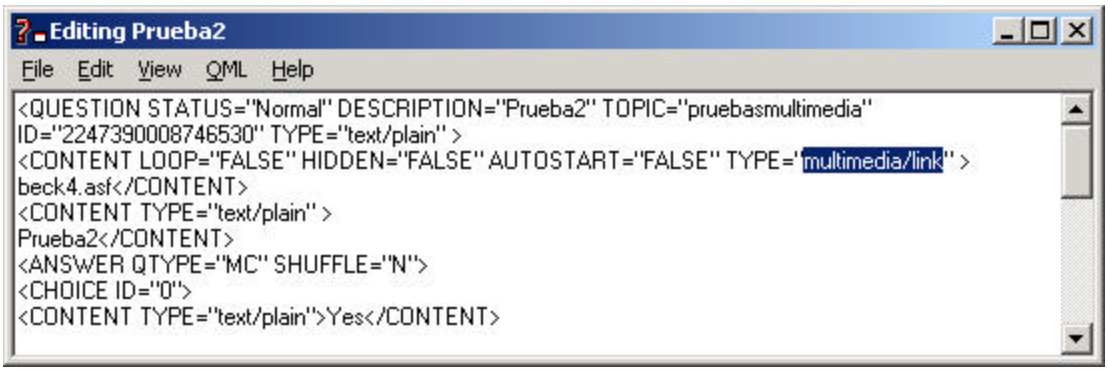

Figure 5.3 QML string of a question including a linked file.

Having a look at the figures it is easy to deduce that QML tags for including multimedia into questions are the following:

<CONTENT TYPE="multimedia/link">name of the file</CONTENT>

<CONTENT TYPE="multimedia/embed">name of the file</CONTENT>

As one can observe, only the name of the linked file and not the complete URL is stored in the QML string.

When translating the QML string to HTML, Perception uses a file named htmlcore.format. This format file defines all the correspondences between QML and HTML. At the end of that file, the translation of the multimedia link and embed are defined too.

MULTIMEDIA_EMBED=multimedia/embed
MULTIMEDIA_LINK=multimedia/link

MULTIMEDIA_EMBED = <EMBED SRC = "%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%" %CONTENT.ATTRIBUTES%>

MULTIMEDIA_LINK = <A HREF = "%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%"> %CONTENT.NAME%</A>

CONTENT.URL refers to the name of the media file referenced. It is stored in the QML string.
CONTENT.NAME refers to the text that will be shown by the link.
CONTENT.ATTRIBUTES refers to the possible attributes of the embed tag.
SERVER.GRAPHICS is the graphics directory defined. By default it is …./perception3/server/resources. In the resources directory all complementary files from questions are stored (multimedia, graphics, java files, etc).

We must modify that file in order to make those links point to the publishing point we have just defined instead of the graphics directory.

For example, if our publishing point is the “home” publishing point defined on the machine SERVER.URL (it is not necessary to specify the name of the publishing point because of being the home) and if we are using the mms protocol, the file will have to be modified in the following way:

MULTIMEDIA_EMBED= <EMBED SRC = "mms://SERVER.URL/%CONTENT.URL%"%CONTENT.ATTRIBUTES%>

MULTIMEDIA_LINK = <A HREF = "mms://SERVER.URL/%CONTENT.URL%">%CONTENT.NAME%</A>

After adapting those links in the htmlcore.format file the server dll’s have to be reset to include the new changes. Resetting the dll’s can be done via the Enterprise Manager by an administrator with Server Management privileges or by restarting the machine where the server is running.

5.4 Simple coordination

There is a way of performing easy first step coordination between de servers. It is possible to run the Media Server on the same machine where Perception Server is running. We can also define the publishing point in the multimedia directory of the Perception server (…./perception3/server/resources/multimedia).

Now we can redirect the links as explained above in the htmlcore.format file to point to that publishing point (mms://IPaddress_PerceptionServer/aliaspublishingpoint).

Just by doing this, when clicking a link or when clicking play on an embed player, the multimedia file will be streamed by the media server.

Although very simple and easy to build, with this system coordination we will not reach the results expected for two main reasons:

- In chapter number 6 we have explained that only fixed types of files can be used to be streamed, .MP3, .asf, .wma, are the format files that Windows Media Server supports. Using this simple coordination configuration only those types of files will be suitable to be used, because all the multimedia files will be served by the Media Server through the publishing point defined. The player will try to connect the server only through that publishing point, so if the file is not suitable to be streamed, the player won’t be able to connect to the server and an error message will be thrown instead.
  So, we are limited to use only those three types. If another file extension is used by error, the participant will not be able to visualize it.

- By installing the media server in the same machine where Perception server was already intalled, the server performance can be seriously reduced and affected.
  If the server is busy attending media requests through the publishing point defined, it won’t be able to attend perception requests without reducing the efficiency it had.

It's not worth to build this system in order to improve the efficiency of Perception when using multimedia files, if afterwards the performance of Perception itself will be reduced.

5.5 Adjusting the requirements

Now that we know more about the Perception and Windows Media Server possibilities we are about to define more exactly the characteristics required for the final system.

- First of all, none of the present possibilities of Perception will have to be reduced or limited. If up to now, we were able to include all kind of multimedia files in the questions, the same possibility would have to be included in the final configuration, even if some files cannot be streamed and have to be downloaded.

- Secondly, the Server Performance does not have to be affected or reduced. If afterwards it is finally decided to install the Media Server in the same machine where the Perception Server is running, this is up to the administrator of the system but there must be at least another possibility.

- Finally, the new system management must keep the same structure than the present system. That way administrative, authoring, or publishing tasks of Perception must remain the same.

5.6 Solutions that meet with the requirements

We still have two possible solutions which meet the requirements we have imposed to ourselves:

a) If the system administrator wants to relax a little or even ignore the second condition we can achieve an easy solution to our problem: We can select the Perception server multimedia directory also as the publishing point directory, but in that case we have to find a way to make a distinction between the files that must be streamed and those which must be kept in the default download structure.

b) The other possible solution is to configure the Windows Media Server in a different server machine and arrange the Perception server to redirect the multimedia links to that new location.

Solution *a* is simpler and more economical; we don't need different server hardware to add the new functionality. Instead, all the Perception system and the media services can be configured in the same machine. The disadvantage of solution *a* is that the server performance can be considerably affected and becomes worst as the quantity of requested multimedia files grows.

On the other hand, solution *b* manages with all the precedent requirements but has the disadvantage that new server hardware has to be dedicated for streaming purposes, with the extra cost implied.

Solution *b* is the recommended one when the number of participants is large or if there are going to be many Participants taking assessments at the same time.

When the number of participants is low enough or the number of requested multimedia files at the same time is going to be low, then the recommended solution for this case is solution a).

5.7 Necessary system changes

To configure Perception to work in the way explained in solutions *a* and *b* several changes have to be made to the system:

- First: we need to define new tags in QML that will indicate the difference if the multimedia files are suitable to be streamed or just downloaded.
- Second: we need to change the tags in the QML strings of the questions stored in the server question database, writing the new ones when streaming will be available.
- Third: we have to look for the referenced multimedia files in the Perception server multimedia folder and store the ones which are going to be streamed in the machine and directory defined in the publishing point.
- Fourth: an easy to use means to perform these changes to the questions has to be provided to the authoring administrators in order to make the new functionality as transparent as possible for them.

5.7.1 QML tags for streaming multimedia

We have to differentiate between the files that are going to be streamed and those which will be downloaded with the default option.

When the assessment is taken by a participant, as explained, the QML string of each question is translated into HTML using the definitions in htmlcore.format file.

Now we need to extend those definitions in order to support two types of multimedia references:

- The default one creates a link to the multimedia files in the Perception multimedia directory, and downloads the file to the client machine when it is attempted to be played.

- A new one will create a link to the publishing point define in the Windows Media Server, this link will have to be created with all the files with suitable format to be streamed (.wma, .asf, .MP3).

For example, these new tags could be (represented here together with the old ones to see the differences):

Download link: <CONTENT TYPE="multimedia/link>name of the file</CONTENT>

Streaming link: <CONTENT TYPE="streaming/link">name of the file</CONTENT>

Download embed: <CONTENT TYPE="multimedia/embed">name of the file</CONTENT>

Streaming embed: <CONTENT TYPE="streaming/embed">name of the file</CONTENT>

Now, let us define the relations between those QML tags and real HTML tags in the htmlcore.format file in the following way:

```
MULTIMEDIA_EMBED=multimedia/embed
MULTIMEDIA_LINK=multimedia/link
STREAMING_EMBED=streaming/embed
STREAMING_LINK=streaming/link
```

```
MULTIMEDIA_EMBED = <EMBED SRC = "%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%" %CONTENT.ATTRIBUTES%>

MULTIMEDIA_LINK = <A HREF = "%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%"> %CONTENT.NAME% </A>

STREAMING_EMBED = <EMBED SRC = ""mms://ICTOPC10/%CONTENT.URL%" %CONTENT.ATTRIBUTES%>

STREAMING_LINK = <A HREF = "mms://ICTOPC10/%CONTENT.URL%">%CONTENT.NAME%</A>
```

5.7.2 Changing the question tags

The next thing to do is to change the multimedia references in the QML strings of the questions.

As explained before, Perception questions are stored in QML strings in the question databases. Once the assessments are published, the questions are stored in the server question database defined. By default, that question database is …./perception3/server/databases/masterq.question.

This database is divided in several tables; the table named Q_QML is the table containing the question text, more concretely the column QML String (whose real name is QML_Data).

For doing this, a Java [9] application (PerceptionStreaming.java) has been created which connects to the database using JDBC (=Java Database Connectivity) [10]. It gets the QML string of the corresponding questions and looks for the multimedia tags. If they exist, the application looks for the extension of the referenced file. If the extension of the file is .wma, .asf or .MP3, then it modifies the standard tag with one of the new tags defined in the previous point. If the replaced tag was of the embed type, the new one will be also of the embed type and the same with the link type.

PerceptionStreaming.java will be explained further in chapter 6.

5.7.3 Storing the multimedia files in the correct folder

Once the reference has been adapted to point to the chosen publishing point defined, it is also necessary to copy the referenced multimedia files in the correct directory to be streamed.

PerceptionStreaming.java will also realize this task.

When including a multimedia link in the Question Manager, the multimedia file is copied in the author resources directory. Afterwards, when the assessment is published, the files will need to be copied to the server resources directory.

When the QML tag has been changed because the multimedia file was suitable to be streamed, the java application will have to search for the multimedia file in the server resources manager and store it again in the directory defined by the publishing point.

If the Windows Media Server is configured in the same machine where the Perception Server is and the directory defined by the publishing point is the same as the server resources folder then this process will not be necessary.

Otherwise, both if the machine is different or the directory is different, we will have to run a second java application (ServerPerceptionStreaming.java) in the local machine where the Windows Media server will be configured that will act as a server and will be always waiting for connections from the first java application that now will act as a client.

The client (PerceptionStreaming.java) will connect to the server (ServerPerceptionStreaming.java) when needed and will send the multimedia file to it. The server will receive the file and will store it in the correct directory defined as the publishing point.

5.7.4 User Interface

The characteristics of the user interface must be transparency and similarity to the rest of Perception applications.

Running a java program is not at all a comfortable task if you are not used to work with java. It is even worse if you have to run it each time you want to adapt your assessments for streaming delivery.

Moreover, apart from the comfort and simplicity, we are also looking for a secure interface, at least as secure as the Perception system is.

Enterprise Manager provides a secure way of administering the system remotely through the Internet.

We can use that interface provided by Perception to include the new functionality of adapting the assessments for streaming delivery. Using Enterprise Manager, we will work in the same secure environment of Perception and with the same user interface. That way, administrators won't see any big difference in the way they work when including the new streaming functionality to the system.

6. FILE MODIFICATIONS AND NEW APPLICATIONS

In this chapter, all the needed modifications to the existing Perception files will be explained as well as the function and configuration of the new java applications will be set clear.

6.1 Action overview

First of all let us summarize all the different modifications that have to be made to the system:

- New QML tags have to be defined in the htmlcore.format file. This task has to be accomplished manually. In appendix A we can find the code of this file. The added lines in bold.

- Reference tags to multimedia files in the questions have to be changed when streaming is possible, that is when the extension of the files is .asf, .wma or .MP3.

- If streaming is possible and the tags have been modified, the referenced multimedia file has to be copied to the directory defined by the publishing point.

- The Enterprise Manager has to be modified in the way that it includes the possibility of adapting the assessments to multimedia streaming. Only administrators with authoring privileges should be able to see this new function.

6.2 Servlets

Enterprise Manager is a web application to manage and administer the Perception system remotely.

As explained, the idea is to include the administration and running of the new streaming facility inside Enterprise Manager, that way, the interface is already known by the administrator and we will be working in the same secure environment as Perception itself.

Besides the applications which facilitate the translation from the download system to the new streaming system are java applications and will be explained below.

When an administrator logs on to the Enterprise Manager for adding streaming functionality to one of his assessments, how is it possible to run the java applications from the web page? The answer to that question is with the use of servlets [11].

Java Servlet technology provides a simple consistent mechanism for extending the functionality of a web server. This technology was created as a portable way to provide dynamic, user-oriented web content. A servlet can be thought as an applet that runs on the server side.

The java applications developed are in fact servlets running on a servlet engine which must be installed in the same machine where Perception is running.

With the use of the <form> tag within the web pages we can call the servlet, or in other words run the java classes of what is composed.

6.3 Multimedia Adaptor

Multimedia Adaptor is the selected name for the whole application developed as the result of this thesis. The name stands for the fact that it adapts the assessments containing questions with multimedia files for streaming.

Multimedia Adaptor must be run after publishing the assessment to the server databases, including the multimedia files.

6.3.1 Availability of the assessments and the media files

It is possible to check if the assessment has been correctly published logging into the Enterprise Manager with authoring privileges and selecting **assessment manager** in the **authoring menu**, you will enter what is known as *Browser Based Assessment Manager*. A screen with all the available assessments will be presented (see figure 6.1). If the name of your assessment is in the list then it means that it has been correctly published.

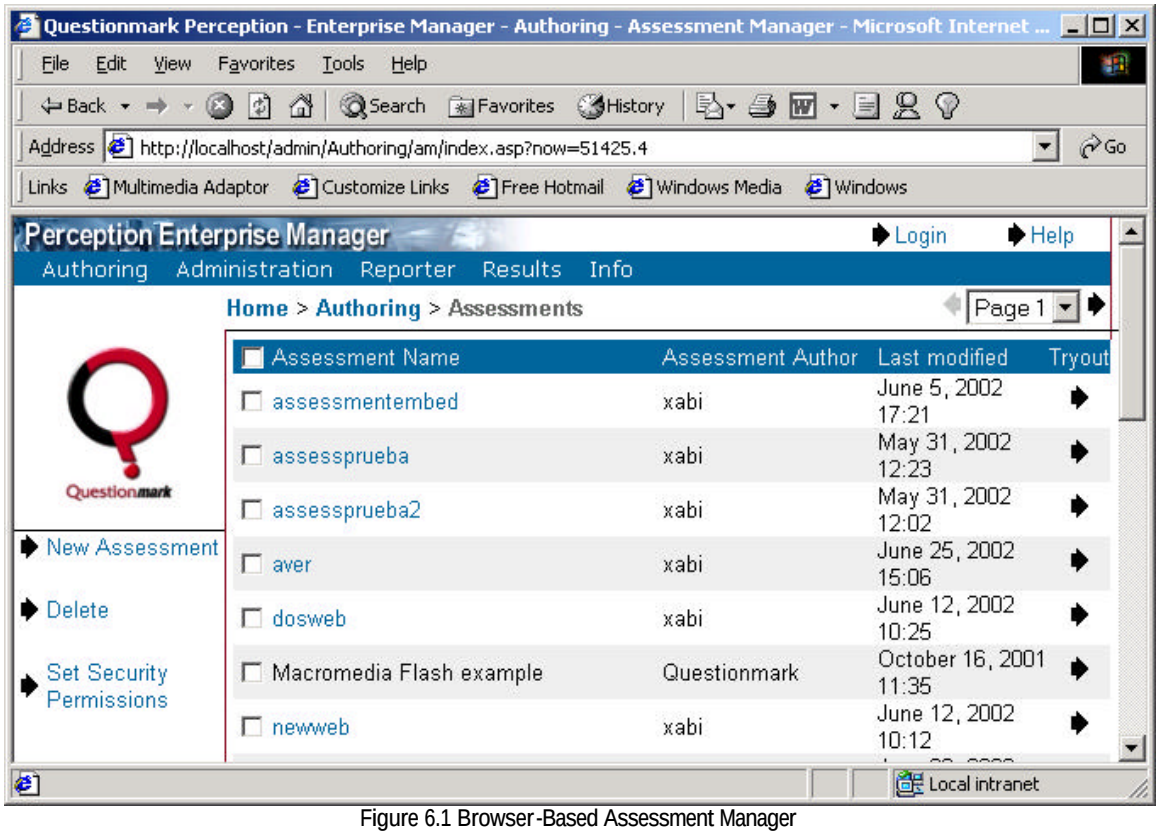

Figure 6.1 Browser-Based Assessment Manager

Also the availability of the multimedia files referenced in your questions can be checked from within the Enterprise Manager. Now you have to select **Resource Manager** and then **Multimedia Folder** and again a list with all the multimedia files available in the server is shown (Figure 6.2).

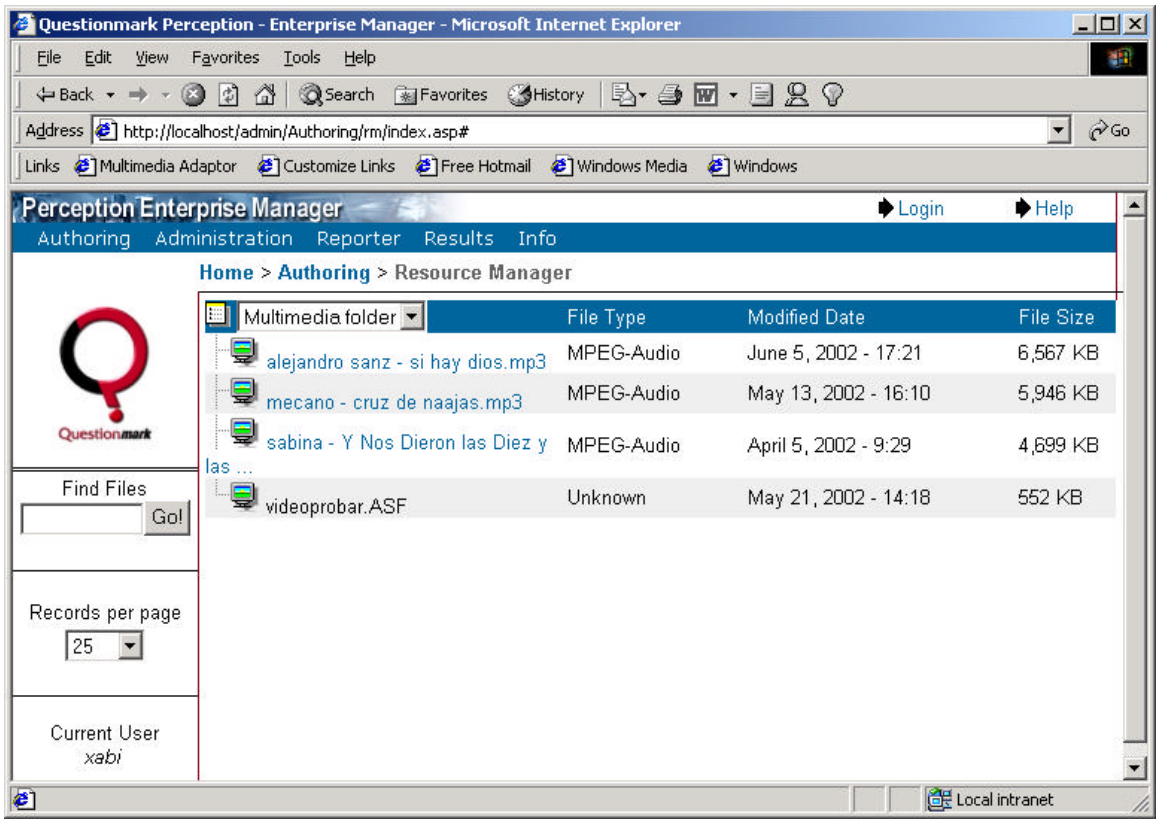

Figure 6.2 Resource Manager

6.3.2 Running the Adaptor

Once the availability of the assessment and of the media files has been checked, the next step is running the Multimedia Adaptor. For doing that you have to come back to the home page of the Enterprise Manager. A new entrance has been added to the home page, the one linking the Enterprise Manager home page to the Multimedia Adaptor.

In figures 6.3 and 6.4 we can see the difference between the aspect of the standard Enterprise Manager Home page and the one when the Multimedia Adaptor has been configured.

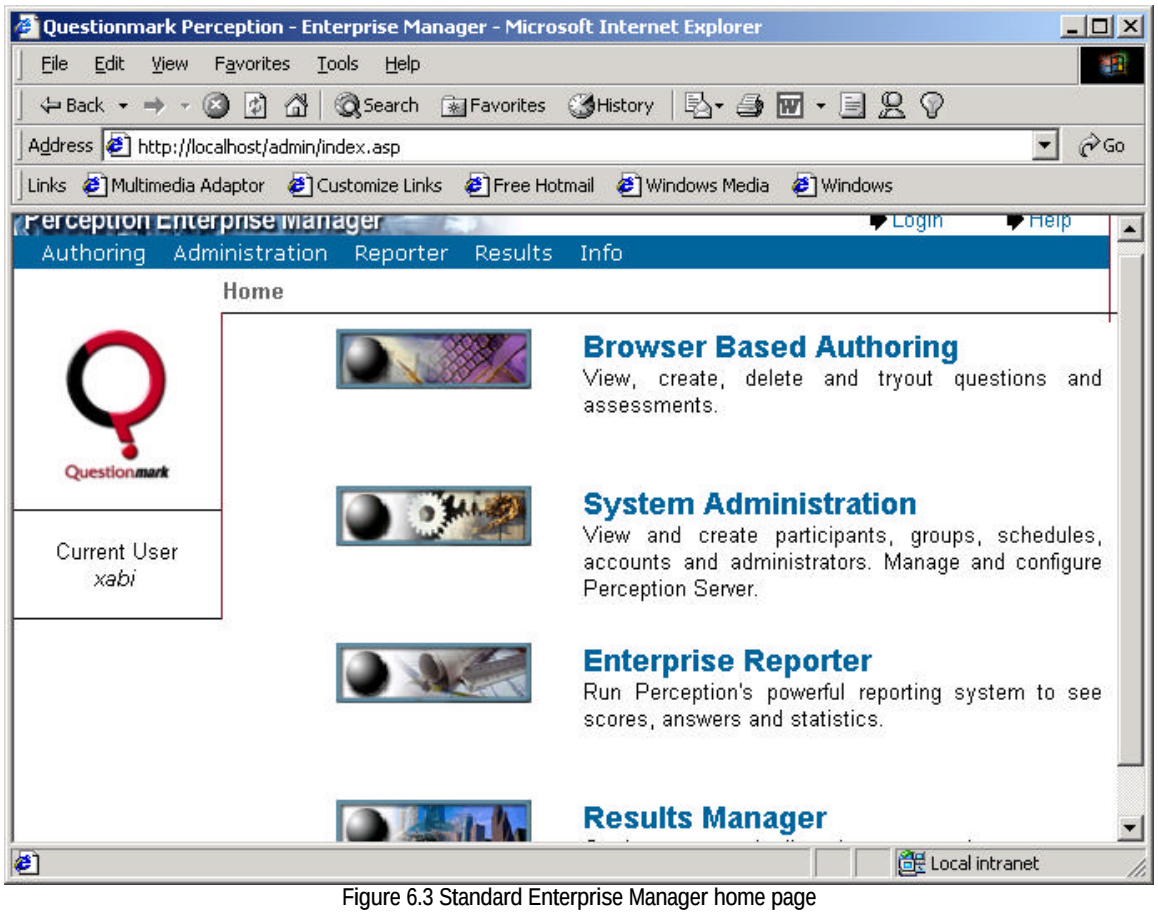

Figure 6.3 Standard Enterprise Manager home page

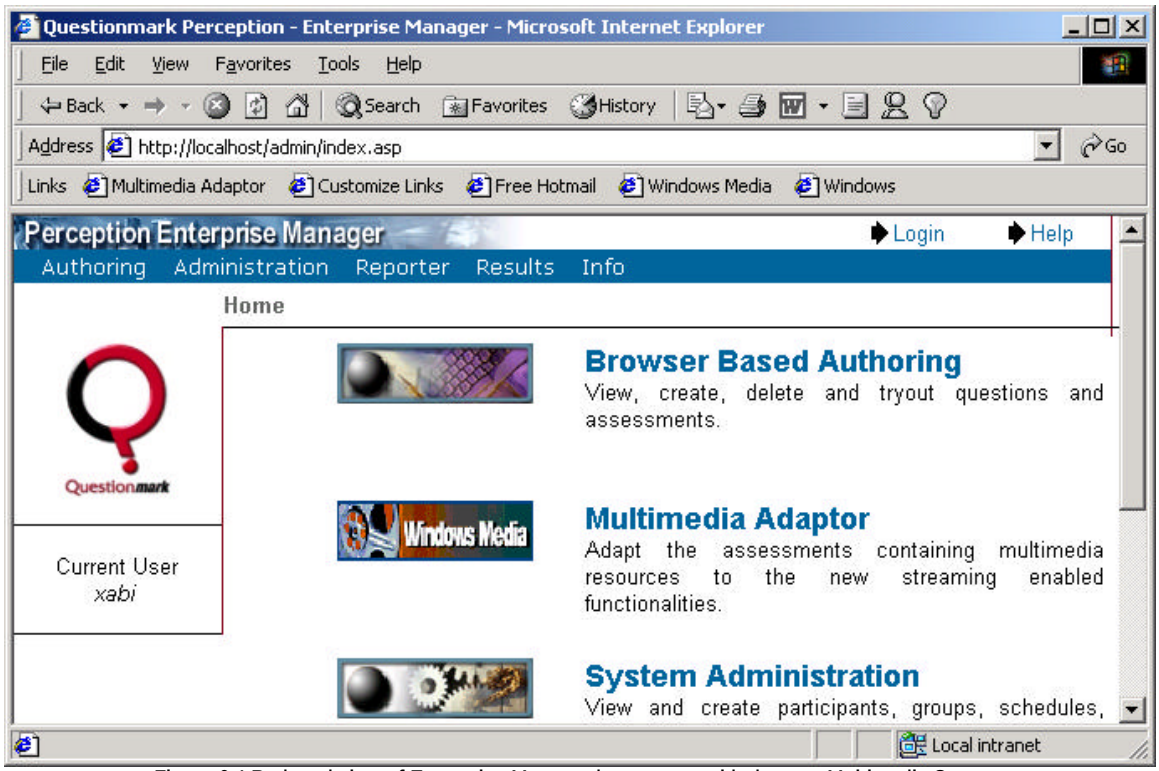

Figure 6.4 Reduced view of Enterprise Manager home page with the new Multimedia System

When one clicks on the new link, he enters the Multimedia Adaptor Home page which can be seen in figure 6.5. This home page contains a little explanation of the Multimedia Adaptor functionality and a text box to write the name of the assessment that is going to be adapted. It also contains a button to access a list of all possible assessments in case you do not remember its name.

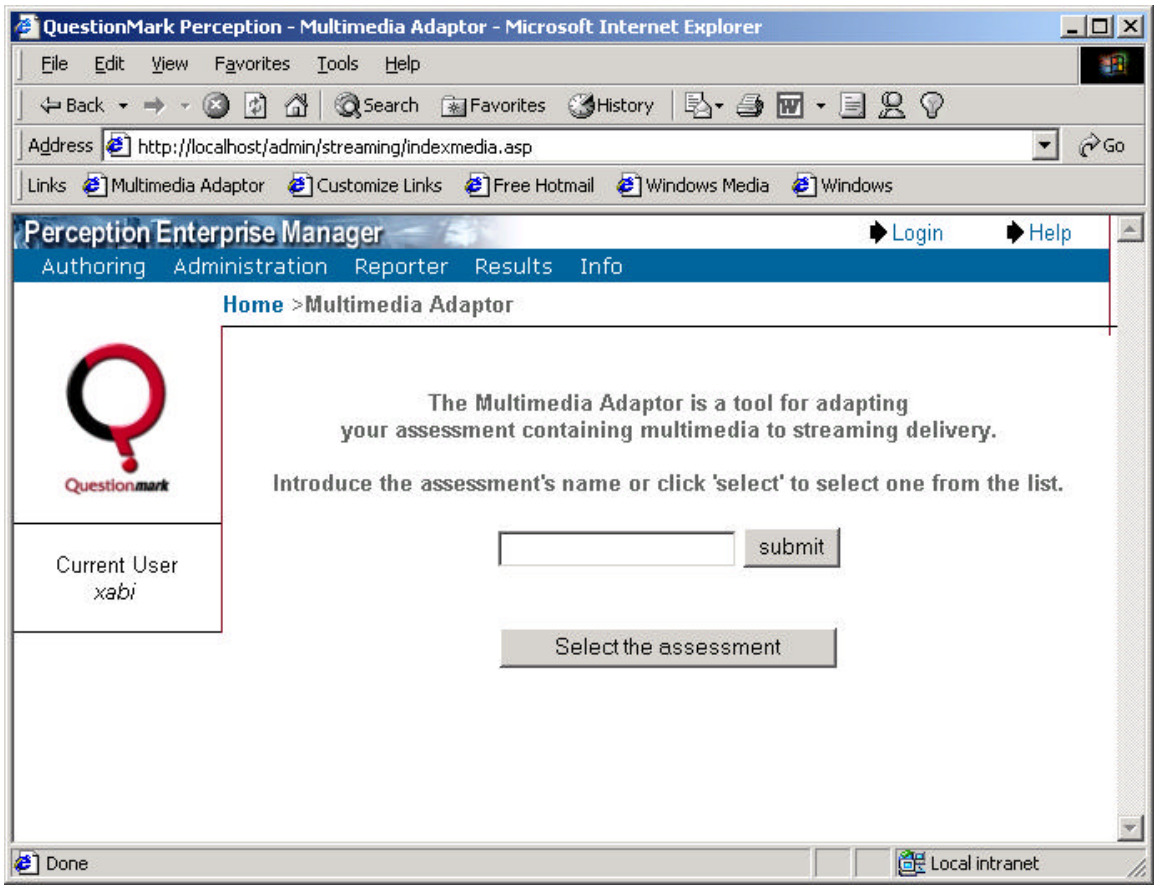

Figure 6.5 Multimedia Adaptor home page.

6.4 File modifications

Apart from the modifications explained in the htmlcore.format file (paragraph 5.7.1), another file must be modified to configure the Multimedia Adaptor system. That is the Index.asp file located in the em (enterprise manager) folder of the Perception server directory.

6.4.1 Index.asp

The file …./perception3/server/em/index.asp is the file corresponding to the Enterprise Manager home page.

In the appendix A we can see a copy of this file with the modifications shown in bold. The modification is just to add a new row to the HTML table containing the links to each of the submenus of the Enterprise Manager. This row has to be added in the exact place where it is, in order to only let administrators with authoring privileges see and use the functionality.

6.5 New applications

Besides of the modifications in the corresponding files four new applications have to be run in the server. In the step by step guide (chapter 7) we will see where and how to run them.

6.5.1 Indexmedia.asp

Indexmedia.asp is a new ASP [12] file created following the Perception security system and which also uses the same interface as Perception. A copy of Indexmedia.asp can also be seen in the appendix.

Apart from the security functions, this program contains two HTML forms:

- The first one contains four input fields:

  1. A text box to introduce the name of the assessment.
  2. A hidden type which contains the name of the current user.
  3. Another hidden type which contains the checksum, that is, the coded password number of the user.
  4. A submit button.

- The second one contains only three input fields

  1. A hidden type which contains the name of the current user.
  2. Second hidden type with the checksum.
  3. A submit button which value is ‘select the assessment’.

The first form will connect to the servlet PerceptionStreaming.java when the submit button is clicked.

The second form will connect to the servlet LookforAssessments.java when the ‘select the assessment’ button is clicked.

Both the user name and the checksum are used for security reasons. In the servlet side it is not possible to call the security functions of Perception. Instead it is possible to access the Perception Databases because they are running on the same server where Perception is installed.

The checksum is a number that represents the coded password of the user. This number is stored in the admin Perception database (in the G_User table) so it is only accessible locally. When the servlet obtains that number it verifies in the database if it is a valid checksum number and then if it corresponds to the current user. Figure 6.6 presents a partial sight of that table.

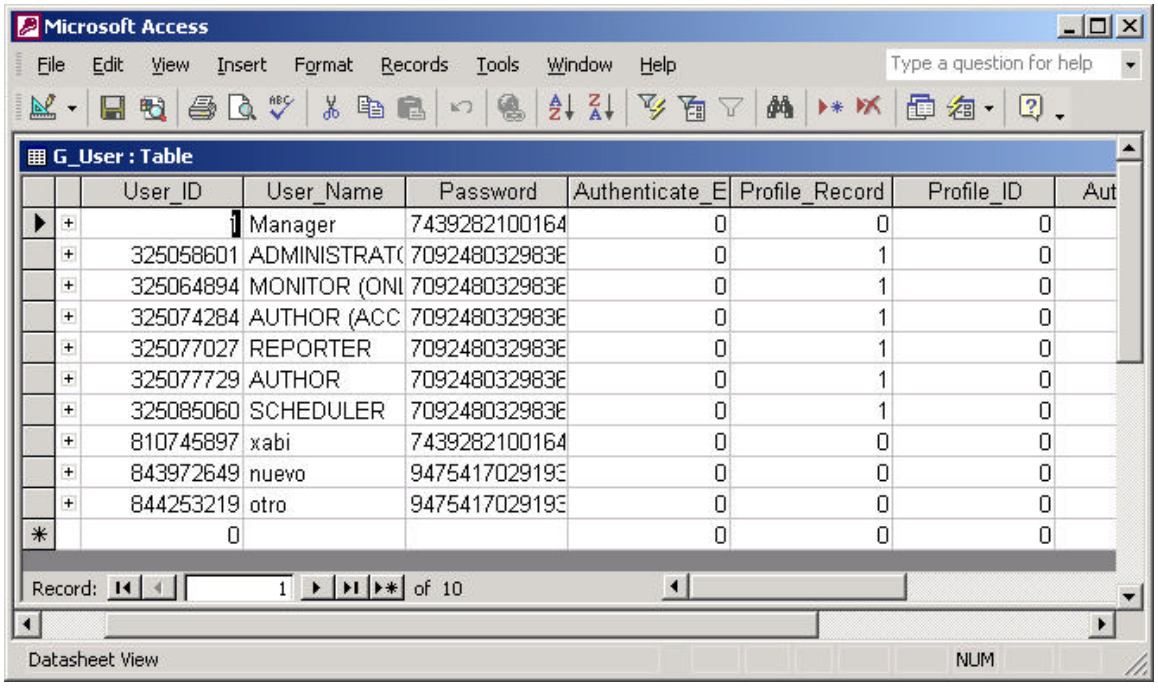

Figure 6.6 G_User table of the Perception Admin database

6.5.2 LookforAssessments.java

LookforAssessments.java is a servlet used for the Multimedia Adaptor to obtain a list of all the assessments that are currently published.

LookforAssessments.java can be called either from Indexmedia.asp or from PerceptionStreaming.java.

LookforAssessments.java needs two parameters to run. Those parameters are obtained with the HTTP [8] POST method.

The parameters are the ones explained in the preceding paragraph: checksum and current user.

The first action of this servlet is to check in the Perception admin database if the parameters passed are correct. Otherwise a screen of not authorization for running the program will be shown.

Then it accesses the Perception server assessment database (mastera.assessment) and gets all the assessment names of the published assessments (in the Session_Name column).

Then output of this servlet is a web page written in HTML with the same interface that the Enterprise Manager showing a table with all the assessment names obtained from the database. The output also contains an HTML form with four fields: checksum, username, one "radio" field and a submit button. Checksum and username are used for the same security purposes they had.

The radio field is the assessment we have previously selected from the table. By clicking the submit button PerceptionStreaming.java will be called and the checksum, username and assessment name selected will be passed to it again with the HTTP POST method.

The purpose of this program is to make it easier for the user to execute the Multimedia Adaptor, selecting the assessment name from the list instead of remembering and writing the name.

The list of names presented as the output of the program is the complete list of all published assessments, not only the assessments containing multimedia. Even the assessments that have been previously adapted are not excluded from the list. We have decided to do this in that way because although it is relatively simple to keep a list with all the assessments that have been already adapted, it is quite complicated to keep that list up to date. Assessments can be modified at any moment, add or delete questions from them and save them with the same name. Besides, the assessments don't contain the multimedia but the questions do. The same question can be included in different assessments. So, maintaining that hypothetical list is quite a complex process that is not worth to develop just to show the names later in the table.

A flow chart can be seen on figure 6.7 representing the operation of this application.

6.5.3 PerceptionStreaming.java

PerceptionStreaming.java is the most important application of the entire Multimedia Adaptor system.

There are three parameters in the application: Checksum and username, with the same security function than in the previous application and the name of the assessment we are going to adapt which has been either selected from the list with LookforAssessments.java or written in the text box in the Multimedia Adaptor home page.

The tasks performed by this application are the following:

- The application checks the admin database for the checksum and username. If they are not correct a screen with a no authorization message is shown.

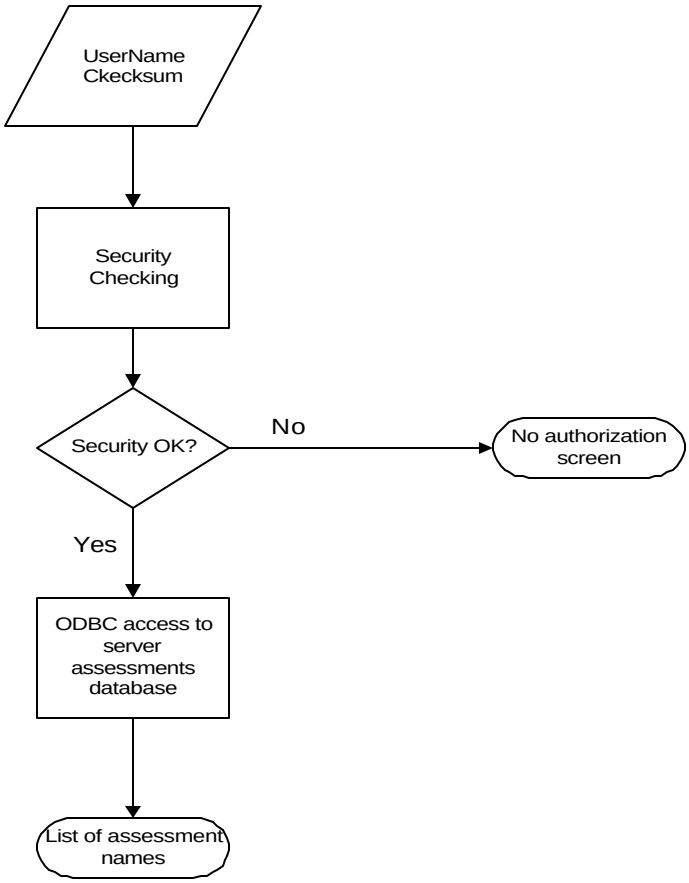

Figure 6.7 Operation of LookforAssessments.java

- If the assessment name has been passed to the application by writting the name in the text box presented in the Multimedia Adaptor home page (through Indexmedia.asp), it checks if that assessment name exists in the S_Header table in the assessment database. That step is performed because a spelling error can happen writing the name. If the name provided is not a valid assessment name, a screen with that message and with the option to select the name from the list (calling LookforAssessments.java) is shown.

- If the assessment name has been checked and is correct or if it has been selected from the list then the procedure for obtaining the QML string of the questions depicted in paragraph 3.4.3 is executed.

- Once we have found the QML string of a question, the application looks for the QML tags referencing multimedia files. If they exist, it checks if the multimedia file referenced is of the .MP3, .wma or .asf types.

- If the media file referenced is suitable to be streamed, the application will perform three different actions:

  - The QML tag will be changed with the new one defined for streaming as explained in paragraph 5.7.1.
  - The Question_ID number will be added to the real name of the multimedia file to obtain a new name of the referenced file as follows:

    Question_IDRealName.extensionofthefile

  - The media file referenced will be copied in the directory defined as the publishing point with the new name given in the preceding point.

- The output of the application will be a web page with a message explaining that the indicated assessment has been correctly adapted for multimedia streaming. If any problem is encountered during the adaptation then a message with the java exception will be shown in the web page and with a recommendation to contact the administrator.

6.5.3.1 Why does the application change the name of the media file?

There are two reasons to change the name of the media file referenced.

The first one and most important is a security reason. A publishing point is an open point accessible from anywhere in the Internet. By writing the URL of the publishing point and a valid name of a media file stored on it then it is possible to play the file from any Internet connection without any security requirements.

Perception provides the functionality to schedule the assessments in order to prevent the access to them before a determined day and hour or after certain period of time, but the publishing point doesn't have that functionality.

Besides, an author may follow certain procedure to name the multimedia files included in his/her questions, for example: multimediafile1, multimediafile2, etc. In that case it would be quite easy for anyone knowing that procedure to access the content once the Multimedia Adaptor has been run.

If the real name of the file is preceded by the Question_ID (a random name given by Perception to identify the question throughout the system), it is almost impossible to guess the new name given to the file and so it is impossible the internet access to it.

Besides, it is possible to use the same name for different multimedia files (several author may use the same procedure explained above) because afterwards a unique name (Question_ID + real name) is given to each file.

6.5.3.2 How does the application copy the file to the new location?

To copy the file to the new location server-client application is used.

The client will be the PerceptionStreaming.java application and the server will be a new application called ServerPerceptionStreaming.java.

ServerPerceptionStreaming.java will be running in the same machine where the Windows Media Server is running and will be always waiting for connection from PerceptionStreaming.java.

The connections are managed using the java Socket [13] technology.

When a multimedia file has to be copied to a new location, two sockets are opened connecting the client and the server. In the first one, the name of the file (that now will be Question_ID + real name) is sent and the second socket is used for sending the multimedia file itself.

A flow chart can be seen on figure 6.8 representing the operation of PerceptionStreaming.java.

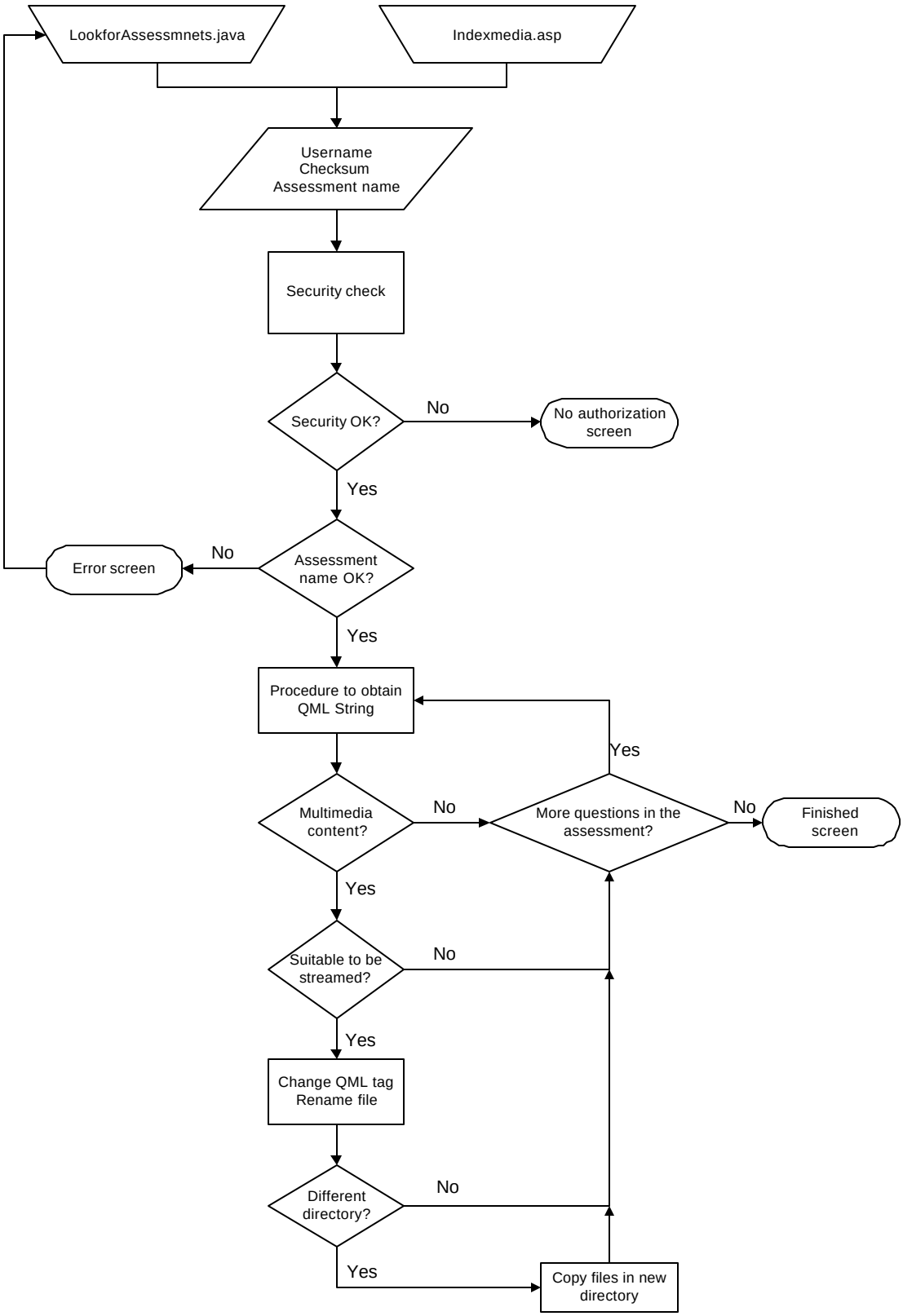

Figure 6.8 Operation of PerceptionStreaming.java

7. STEP BY STEP CONFIGURATION GUIDE

A summary of the steps to configure the Perception system to include the Multimedia Adaptor is presented in this chapter.

This is a technical chapter so if you are not really interested in configure your Perception system with the Multimedia Adaptor you can skip it.

7.1 Prerequisites

Perception Server version 3 is supposed to be running with the standard configuration.

The names for folders and databases used in this chapter are the standard ones defined by Perception, but can be adapted to your particular configuration as will be explained.

Windows Media Services has to be running either on a dedicated server machine or in the same computer where the Perception server is installed. Follow the instructions indicated in the Windows Media Services web site [2] to avoid port conflicts in case they share the same computer.

A java development kit[10] must also be run on both the computers where Perception and Windows Media servers are installed. It can be downloaded for free from the java official web site [14].

A servlet engine has also to be running on the same machine where Perception is running. The simple servlet included in the JavaServer Web development kit (jswdk) [15] is enough for our needs and can also be downloaded for free form the java web site.

7.2 Publishing Point

Configure the publishing point from which the media files are going to be streamed following the instructions in paragraph 5.2.

If the Windows Media Server shares the computer with the Perception server you can place the Publishing Point in the same directory where multimedia files are stored by Perception (…/perception3/server/resources/multimedia), but the recommendation from Microsoft is to place it in a dedicated NTFS partition even situated in a physically separated hard disk. This kind of partition has been proved to have a better performance with streaming than FAT partitions.

[10] The recommended one is j2sdk1.4.0 [12] because is the one used to develop the applications which will be run

7.3 Servlet engine

There are two possible ways to configure the servlet part of the Multimedia Adaptor.

First option: All calls to the servlets from the asp applications are of the form:

http://perception_server_name:8080/multimediaadaptor/name_of_the_servlet

So you must configure the port in your sevlet engine to be the 8080 and one of the web application names must be “multimediaadaptor”.

Second option: You can change all the references to the servlets in the different applications to adapt them to your web application name and port in the servlet engine.

7.4 File modifications

7.4.1 Existing files

Make the necessary changes in the files:

htmlcore.format

…./perception3/server/em/index.asp

Those changes can be seen in the appendix A and are explained in the previous chapters of this thesis. Be careful to modify the URLs of the servlets.

7.4.2 New applications

The new applications (appendix B) must be adapted for each system where they are going to be runned.

The points to adapt are the following:

- Modify in PerceptionStreaming.java the URL to the publishing point defined. The place to change this URL is indicated in the first lines of the code.
- Modify the URL to the Perception Server in PerceptionStreaming.java and in LookforAssessments.java.
- If the DSN names of the Perception Server databases are changed make also the necessary modifications on both applications, the lines to change are also clearly indicated in the code.

7.5 Compile and run the application

Once the .java files has been changed compile the applications with the command:

javac –classpath path_to_the_servlet.jar name_of_the_application.java

If you are using the *java server web development kit* the servlet.jar file is located in the lib directory.

To run javac you must either add the full path of the JDK[11] bin directory to the PATH[12] variable or change the current working directory to that directory.

Once the application has been compile, copy the obtained *.class* files to the web application directory defined in your servlet engine and start the engine.

If you are using the *java server web development kit,* double clicking in the startserver.exe icon will start the servlet engine.

7.6 Streaming directory

Make a new folder named *streaming* inside the …./perception3/server/em directory.

Place the indexmedia.file and the images multimedia.gif and noauth.gif inside the new directory.

7.7 Socket server

If the Windows Media player is running in a different machine than the Perception server or if it is running in the same but the publishing point directory is different from the multimedia folder in the resources directory of the server, then, run the ServerPerceptionStreaming.java application by executing in a command prompt the following order:

java ServerPerceptionStreaming

As in paragraph 7.5 you will have to set the PATH variable or change to the JDK bin directory.

[11] JDK = Java Development Kit

[12] To set the path permanently: In Windows NT: select Start -> Control Panel. Select System, select Environment, and look for “Path” in the User Variables and System Variables.

8. CONCLUSIONS AND FURTHER WORK

The obtained result of the coordination between the Perception and the Windows Media Servers is an improvement with respect to the standard download system. The waiting time for downloading assessments with multimedia included, has been reduced with the new system and also the internal traffic has been made less bursty when participants request assessments with multimedia.

However, the performance of the system still depends on the traffic and bandwidth availability of the Internet. The streaming technology improves the results obtained with multimedia comparing to download systems with the same quality of service. But the dependency on the load of the Internet is so high that the results cannot be appreciated clearly.

The next step would be to try to make this system as independent as possible from the Internet. For doing that, it is possible to configure one Windows Media server in each local area network belonging to the Erasmushogeschool Brussel. That way, the requests for multimedia included in the assessments could be attended locally where the bandwidth availability is always much bigger.

9. REFERENCES

[1] Question Mark Perception. URL: www.questionmark.com/perception

[2] Windows Media Services: URL: http://www.microsoft.com/windows/windowsmedia/technologies/services.asp

[3] World Wide Web Consortium, “Hypertext Markup Language”, January 2000.

[4] Microsoft ODBC. URL: http://www.microsoft.com/data/odbc/

[5] File Transfer Protocol specification. URL: http://www.w3.org/Protocols/rfc959/Overview.html

[6] Question Markup Language. URL: http://www.questionmark.com/us/qml/index.htm

[7] IP Addressing Architecture URL: http://www.rad.com/networks/1994/ip_addr/tcpip2.htm

[8] Hypertext Transfer Protocol Overview URL: http://www.w3.org/Protocols/

[9] The java tutorial URL: http://java.sun.com/docs/books/tutorial/

[10] Java Database Connectivity URL: http://java.sun.com/products/jdbc/

[11] Java Servlet Technology URL: http://java.sun.com/products/servlet/

[12] Getting started with active server pages (Q297943) URL: http://support.microsoft.com/default.aspx?id=kb;en-us;q297943

[13] Class socket URL: http://java.sun.com/products/jdk/1.2/docs/api/java/net/Socket.html

[14] Java 2 platform standard edition URL: http://java.sun.com/j2se/1.4/download.html

[15] Java Servlet Technology downloads URL: http://java.sun.com/products/servlet/download.html

APPENDIX

A. MODIFIED FILES

A.1 .../PERCEPTION3/SERVER/EM/INDEX.ASP

```
<%
' ************************************************************************
'		Questionmark Perception Enterprise Manager
'		Copyright (c) Question Mark Computing Ltd. 2001-2002
'		index.asp
'		last modified April 2002
'		version 3.2.0
'
'		Enterprise Manager Main Index
' ************************************************************************

ON ERROR RESUME NEXT
%>
<!-- #include file="functions.asp" -->
<%

' Set up any local variables to use throughout this page here.
Dim sUserName ' The Current User's validated login name

' Define the local Subs used in this page
Sub WriteLocalStyles()
%>
		a.heading { font-size:16pt; }
<%
End Sub

Sub WriteLocalScript()
%>
		function help() {
			pageHelp("./help/em3_help_main.html");
		}

		strAuthoring = "<%= TX_INDEX_AUTHORING_STATUS %>";
		strSecurity = "<%= TX_INDEX_SECURITY_STATUS %>";
		strGrading = "<%= TX_INDEX_GRADING_STATUS %>";
		strReporting = "<%= TX_INDEX_REPORTING_STATUS %>";
		imgAuthoring = new Image();
		imgAuthoring.src = "<%= GX_INDEX_AUTHORING %>";
		imgSecurity = new Image();
		imgSecurity.src = "<%= GX_INDEX_SECURITY %>";
```

```
        imgGrading = new Image();
        imgGrading.src = "<%= GX_INDEX_GRADING %>";
        imgReporting = new Image();
        imgReporting.src = "<%= GX_INDEX_REPORTING %>";
        imgAuthoring_glo = new Image();
        imgAuthoring_glo.src = "<%= GX_INDEX_AUTHORING_GLO %>";
        imgSecurity_glo = new Image();
        imgSecurity_glo.src = "<%= GX_INDEX_SECURITY_GLO %>";
        imgGrading_glo = new Image();
        imgGrading_glo.src = "<%= GX_INDEX_GRADING_GLO %>";
        imgReporting_glo = new Image();
        imgReporting_glo.src = "<%= GX_INDEX_REPORTING_GLO %>";

        strHlpSupport = "<%= TX_INDEX_HELP_SUPPORT_STATUS %>";
        strHlpFind = "<%= TX_INDEX_HELP_FIND_STATUS %>";
        strHlpUpgrade = "<%= TX_INDEX_HELP_UPGRADE_STATUS %>";
        imgHlpSupport = new Image();
        imgHlpSupport.src = "<%= GX_INDEX_HELP_ICON %>";
        imgHlpSupport_glo = new Image();
        imgHlpSupport_glo.src = "<%= GX_INDEX_HELP_ICON_GLO %>";
        imgHlpFind = new Image();
        imgHlpFind.src = "<%= GX_INDEX_HELP_ICON %>";
        imgHlpFind_glo = new Image();
        imgHlpFind_glo.src = "<%= GX_INDEX_HELP_ICON_GLO %>";
        imgHlpUpgrade = new Image();
        imgHlpUpgrade.src = "<%= GX_INDEX_HELP_ICON %>";
        imgHlpUpgrade_glo = new Image();
        imgHlpUpgrade_glo.src = "<%= GX_INDEX_HELP_ICON_GLO %>";

        // Changes images and status text
        function hilight(item)
        {
                sTmp = "img" + item + "_glo.src";
                document[item].src = eval(sTmp);
                sTmp = "window.status = str" + item;
                eval(sTmp);
                return true;
        }
        function lolight(item)
        {
                sTmp = "img" + item + ".src";
                document[item].src = eval(sTmp);
                window.status = '';
                return true;
        }
<%
End Sub

Sub WriteNavString()
%><font size=2 color="<%= CX_NAVTEXT %>"><b><%= TX_HOME %></b></font><%
End Sub

Sub WriteSideMenu(nCell)
%>
                                                <p  align="center"> <br><%=  TX_CURRENT_USER
%><br><i><%= sUserName %></i><br> </p>
```

```
<%
End Sub

Sub WritePageContent()
	Dim iColWidth
	Dim iColImgWidth
	Dim iColSpace
	Dim iCountWritten

	iColImgWidth="150"
	iColWidth="400"
	iColSpace="60"
	iCountWritten=0
%>
				<table cellpadding=0 cellspacing=0 border=0 width="100%"><tr>
				<td width="<%=iColSpace%>"> </td>
				<td align=left>
					<table cellpadding=4 cellspacing=0 border=0>
<%
					If arrSubMenus(0)>0 Then
%>

						<tr valign="top" align="left">
							<td width="<%=iColImgWidth%>"><a href="Authoring/index.asp" onMouseOver="return hilight('Authoring');" onMouseOut="return lolight('Authoring');" title="<%= TX_INDEX_AUTHORING_TITLE %>"><img src="<%= GX_INDEX_AUTHORING %>" width="117" height="36" border="0" name="Authoring"></a><br><img src="<%= GX_SPACER %>" width="120" height="50"></td>
							<td width="<%=iColWidth%>">
								<p align="justify">
<font size="4"><b><a href="Authoring/index.asp" onMouseOver="return hilight('Authoring');" onMouseOut="return lolight('Authoring');" title="<%= TX_INDEX_AUTHORING_TITLE %>"><%= TX_INDEX_AUTHORING %></a></b></font><br>
<%= TX_INDEX_AUTHORING_TEXT
%>
								</p>
							</td>
						</tr>
						<tr valign="top" align="left">
							<td width="<%=iColImgWidth%>"><a href="streaming/indexmedia.asp" title="Launch the Multimedia Streaming Adaptor"><img src="streaming/multimedia.gif" width="117" height="36" border="0" name="Adapting"></a><br><img src="<%= GX_SPACER %>" width="120" height="50"></td>
							<td width="<%=iColWidth%>">
								<p align="justify">
<font size="4"><b><a href="streaming/indexmedia.asp" title="Launch the Multimedia Streaming Adaptor" name="Adapting">Multimedia Adaptor</a></b></font><br>
Adapt the assessments containing multimedia resources to the
new streaming enabled functionalities.
								</p>
							</td>
						</tr>

<%
```

```
                                        iCountWritten=iCountWritten+1
                                    End If
                                    If arrSubMenus(1)>0 Then
%>
                                        <% If QSLCheck(sMenuUID, QSL_BIT_S_SM) OR QSLCheck(sMenuUID, QSL_BIT_S_EB) OR QSLCheck(sMenuUID, QSL_BIT_S_SERVER) OR QSLCheck(sMenuUID, QSL_BIT_S_RESET) Then %>
                                            <tr valign="top" align="left">
                                                <td width="<%=iColImgWidth%>"><a href="Administration/index.asp" onMouseOver="return hilight('Security');" onMouseOut="return lolight('Security');" title="<%= TX_INDEX_SECURITY_TITLE %>"><img src="<%= GX_INDEX_SECURITY %>" width="117" height="36" border="0" name="Security"></a><br><img src="<%= GX_SPACER %>" width="120" height="50"></td>
                                                <td width="<%=iColWidth%>">
                                                    <p align="justify">
                                                        <font size="4"><b><a href="Administration/index.asp" onMouseOver="return hilight('Security');" onMouseOut="return lolight('Security');" title="<%= TX_INDEX_SECURITY_TITLE %>"><%= TX_INDEX_SECURITY %></a></b></font><br>
                                                        <%= TX_INDEX_SECURITY_TEXT %>
                                                    </p>
                                                </td>
                                            </tr>

<%
                                            iCountWritten=iCountWritten+1

                                        End If
                                    End If
                                    If arrSubMenus(2)>0 Then
%>
                                        <tr valign="top" align="left">
                                            <td width="<%=iColImgWidth%>"><a href="Reporting/index.asp" onMouseOver="return hilight('Reporting');" onMouseOut="return lolight('Reporting');" title="<%= TX_INDEX_REPORTING_TITLE %>"><img src="<%= GX_INDEX_REPORTING %>" width="117" height="36" border="0" name="Reporting"></a><br><img src="<%= GX_SPACER %>" width="120" height="50"></td>
                                            <td width="<%=iColWidth%>">
                                                <p align="justify">
                                                    <font size="4"><b><a href="Reporting/index.asp" onMouseOver="return hilight('Reporting');" onMouseOut="return lolight('Reporting');" title="<%= TX_INDEX_REPORTING_TITLE %>"><%= TX_INDEX_REPORTING %></a></b></font><br>
                                                    <%= TX_INDEX_REPORTING_TEXT %>
                                                </p>
                                            </td>
                                        </tr>
<%
                                        iCountWritten=iCountWritten+1
                                    End If
                                    If arrSubMenus(3)>0 Then
%>
                                        <tr valign="top" align="left">
                                            <td width="<%=iColImgWidth%>"><a href="Grading/index.asp" onMouseOver='return hilight('Grading');" onMouseOut="return lolight('Grading');" title="<%= TX_INDEX_GRADING_TITLE %>"><img src="<%= GX_INDEX_GRADING %>" width="117" height="36" border="0" name="Grading"></a><br><img src="<%= GX_SPACER %>" width="120" height="20"></td>
                                            <td width="<%=iColWidth%>">
```

```
                                                                <p align="justify">
                                                                    <font size="4"><b><a
href="Grading/index.asp" onMouseOver="return hilight('Grading');" onMouseOut="return lolight('Grading');" title="<%=
TX_INDEX_GRADING_TITLE %>"><%= TX_INDEX_GRADING %></a></b></font><br>
                                                                    <%= TX_INDEX_GRADING_TEXT %>
                                                                </p>
                                                            </td>
                                                        </tr>


<%
                                                        iCountWritten=iCountWritten+1
                                                    End If
                                                    If iCountWritten=0 Then
%>
                                                        <tr valign="top" align="center">
                                                            <td colspan="2" align="center"
width="<%=iColWidth+iImgWidth%>">
                                                                <br> <br>
                                                                <b><%=
TX_NO_PERMISSION_TO_RUN_ENTERPRISE_MANAGER %></b>
                                                            </td>
                                                        </tr>
<%
                                                    End If
%>
                                                    </table>
                                                </td></tr></table>
<%
End Sub

Sub WriteLocalFooter()
End Sub

' Do any initialisation and security checks here.
bValidLicense = False
bValidLicense = ValidLicense(LicenseFilePath)
If (bValidLicense = True) Then
        ' Have we logged on properly.
        sUserName = Request.Cookies("Security")("Name")
        If (ConfirmLoginDetails(sUserName, Request.Cookies("Security")("Checksum")) = True) Then
                Call BuildPage("", TX_INDEX_TITLE, GX_V3_LOGO, True, True, True, 1, "")
        Else
                Call InvalidLogin("")
        End If
Else
        Call InvalidLicense(LX_QMARK_LOGO_LINK)
End If
%>
```

A.2 …./PERCEPTION3/SERVER/FORMAT/HTMLCORE.FORMAT

```
# ##############################################################################
# Question Mark Computing Perception
# Copyright (c) Question Mark Computing Ltd. 1998 - 2001
#
# HTMLCORE.FORMAT version 3.1.0
# Main definitions for converting QML Questions to HTML with Perception Server
#
#                       ***** WARNING *******
#
# These definitions play an important part in how
# Perception Server works when producing HTML documents
# Editing below this line is at your own risk, unless advised otherwise
# by Question Mark technical support.
#
# Always make sure you have a safe backup of this file before changing it
#
# ##############################################################################
# Contains the following sections :
#
#  [DOCUMENT] - User definable text and settings
#           - top level document definitions
#
#  [QUESTION] - define how questions are formatted
#
#  [QUESTION TYPE] - map question type codes to text
#
#  [ANSWER] - define answer format
#
#  [CHOICE] - define format of individual choices within a question
#
#  [OPTION] - define format of individual choices within a question
#
#  [FEEDBACK] - define how feedback to questions is formatted
#
#  [TYPE] - map MIME type to styles
#
#  [STYLE] - formatting for mime types
#
# ##############################################################################

# ##############################################################################
# top level document definitions used for conversion of questions and feedback

# start of document section
[DOCUMENT] # do not change

TARGET=TARGET="_top"

# main format definition for question documents
DOCUMENT=
```

```
_IF %USE_QXQ% = 1
%qxqDOCUMENT%
_ELSE
%TOP%
<FORM NAME="QUESTION" METHOD="POST" ACTION="%SESSION.URL%" AUTOCOMPLETE="off"
%DO_SUBMIT% %TARGET%>
%SESSION.TIMED%
%DOCUMENT_HEADER%
%START_DOCUMENT%
%NO_CLICK%
%SESSION.CONTENT%
%SEPARATOR%
%SUBMIT% %SHOW_RESET_DEF%  %SHOW_QUIT_DEF%
</FORM>
%END_DOCUMENT%
%TAIL%

# ##########################################
# feedback to questions before the end of the assessment
FEEDBACK=
_IF %USE_QXQ% = 1
%qxqFEEDBACK%
_ELSE
%TOP%
<FORM NAME="FEEDBACK" METHOD="POST" ACTION="%SESSION.URL%" AUTOCOMPLETE="off"
%DO_FEEDBACK_SUBMIT% %TARGET%>
%SESSION.TIMED%
%FEEDBACK_HEADER%
%START_DOCUMENT%
%NO_CLICK%
%SESSION.FEEDBACK%
%SEPARATOR%
%CONTINUE%<BR>
</FORM>
%END_DOCUMENT%
%TAIL%

# ##########################################
# feedback to questions to be saved for presentation at the end of the assessment
END_FEEDBACK=
_IF %USE_QXQ% = 1
%qxqEND_FEEDBACK%
_ELSE
<FORM NAME="END" AUTOCOMPLETE="off" %DO_END_SUBMIT% %TARGET%>
%END_FEEDBACK_HEADER%
%SESSION.FEEDBACK%
</FORM>
%SEPARATOR%

# ##########################################
# final page - can include feedback saved using END_FEEDBACK definition
END=
_IF %USE_QXQ% = 1
%qxqEND%
_ELSE
%TOP%
```

```
%END_HEADER%
%SEPARATOR%
%START_DOCUMENT%
%NO_CLICK%
%OUTCOME_DEF%
%SESSION.FEEDBACK%
%SESSION.NEXT%
%END_DOCUMENT%
%TAIL%


# ####################################################
# page definitions for QxQ delivery

# ####################################################
# main format definition for question documents
qxqDOCUMENT=
%TOP%
%QXQ_SCRIPT%
%STYLE_SHEET%
%SPOILER%
<FORM NAME="QUESTION" METHOD="POST" AUTOCOMPLETE="off" ACTION="%SESSION.URL%"
%DO_SUBMIT% %TARGET%>
<DIV ID="list" CLASS="list">
</DIV>
%NAV_CONTROL%
<DIV ID="header" CLASS="header">
%DOCUMENT_HEADER%
</DIV>
%TIMER_CONTROL%
%SESSION.CONTENT%
%SEPARATOR%
%SUBMIT_CONTROL%
%QUIT_CONTROL%
</FORM>
%WRITE_LIST%
%TAIL%

# ###########################################
# feedback to questions before the end of the assessment
qxqFEEDBACK=
%TOP%
%QXQ_SCRIPT%
%IS_FEEDBACK%
%STYLE_SHEET%
%SPOILER%
<FORM NAME="FEEDBACK" METHOD="POST" ACTION="%SESSION.URL%" AUTOCOMPLETE="off"
%DO_FEEDBACK_SUBMIT% %TARGET%>
<DIV ID="list" CLASS="list">
</DIV>
%NAV_CONTROL%
<DIV ID="header" CLASS="header">
%DOCUMENT_HEADER%
</DIV>
%TIMER_CONTROL%
%SEPARATOR%
```

```
<DIV ID="B_%SESSION.NUMBER%" CLASS="question">
%FEEDBACK_HEADER%
</DIV>
<SCRIPT>
        oTemp = new Question(window, "B_%SESSION.NUMBER%", %SESSION.NUMBER%, iIndex, 'B', false, 0,
0, false);
        oTemp.setTitle('%BLOCK.TITLE% - %BLOCK.SCORE% %SCORE_OUT_OF_TEXT% %BLOCK.MAX%');
        aElements[iIndex] = oTemp;
        iIndex++;
</SCRIPT>
%SESSION.FEEDBACK%
%SEPARATOR%
%CONTINUE_CONTROL%
</FORM>
%WRITE_LIST%
%TAIL%

# ###########################################
# feedback to questions to be saved for presentation at the end of the assessment
qxqEND_FEEDBACK=
_IF %SHOW_BLOCK_FEEDBACK% = 1
 <FORM NAME="END" AUTOCOMPLETE="off" %DO_END_SUBMIT% %TARGET%>
 <DIV ID="B_%SESSION.NUMBER%" CLASS="question">
 %END_FEEDBACK_HEADER%
 </DIV>
 <SCRIPT>
        if (isQxQ) {
                oTemp = new Question(window, "B_%SESSION.NUMBER%", %SESSION.NUMBER%, iIndex, 'B',
false, 0, 0, false);
                oTemp.setTitle('%BLOCK.TITLE%  -  %BLOCK.SCORE%  %SCORE_OUT_OF_TEXT%
%BLOCK.MAX%');
                aElements[iIndex] = oTemp;
                iIndex++;
        }
 </SCRIPT>
 %SESSION.FEEDBACK%
 </FORM>
 %SEPARATOR%
_ELSE
<FORM NAME="END" AUTOCOMPLETE="off" %DO_END_SUBMIT% %TARGET%>
 %SESSION.FEEDBACK%
 </FORM>
 %SEPARATOR%


# ###########################################
# final page - can include feedback saved using END_FEEDBACK definition
qxqEND=
%TOP%
%QXQ_SCRIPT%
%IS_FEEDBACK%
%IS_END%
%NO_TIMER_STYLE_SHEET%
%SPOILER%
<DIV ID="list" CLASS="list">
</DIV>
```

```
%NAV_CONTROL%
<DIV ID="header" CLASS="header">
%QXQ_END_HEADER%
</DIV>
%SEPARATOR%
<DIV ID="Outcome" CLASS="question">
%QXQ_END_BODY%
%OUTCOME_DEF%
</DIV>
<SCRIPT>
        oTemp = new Question(window, "Outcome", 0, iIndex, 'O', false, 0, 0, false);
        oTemp.setTitle('%SESSION.TITLE% - %SESSION.SCORE% %SCORE_OUT_OF_TEXT%
%SESSION.MAX%');
        aElements[iIndex] = oTemp;
        iIndex++;
</SCRIPT>
%SESSION.FEEDBACK%
%NEXT_CONTROL%
%WRITE_LIST%
%TAIL%


# ########################################
# header definitions

# show some information about the block
DOCUMENT_HEADER=
%OPEN_DOCUMENT_HEADER%
%SHOW_TITLE_DEF%
%SHOW_BLOCK_TITLE_DEF%
%SHOW_INTRO_DEF%
%CLOSE_DOCUMENT_HEADER%

FEEDBACK_HEADER=
%OPEN_FEEDBACK_HEADER%
%SHOW_BLOCK_TITLE_DEF%
%SHOW_RUNNING_SCORE_DEF%
%SHOW_BLOCK_SCORE_DEF%
%CLOSE_FEEDBACK_HEADER%

END_FEEDBACK_HEADER=
%SHOW_BLOCK_SCORE_DEF%

END_HEADER=
%OPEN_END_HEADER%
%SHOW_END_TITLE_DEF%
%SHOW_FINAL_STATUS_DEF%
%SESSION.FINAL_SCORE%
%CLOSE_END_HEADER%

QXQ_END_HEADER=
%OPEN_END_HEADER%
%SHOW_END_TITLE_DEF%
%SHOW_FINAL_STATUS_DEF%
%CLOSE_END_HEADER%
```

```
QXQ_END_BODY=
%SESSION.FINAL_SCORE%


# show some information about the question
QUESTION_HEADER=
%SHOW_TOPIC_DEF%
%SHOW_NUMBER_DEF%
%SHOW_DESCRIPTION_DEF%

# ################
# sub definitions to turn display of information in headers on or off

# reset button
SHOW_RESET_DEF=
_IF %SHOW_RESET% = 1
 &amp;nbsp;&amp;nbsp;&amp;nbsp; %RESET%

# quit button
SHOW_QUIT_DEF=
_IF %SHOW_QUIT% = 1
 &amp;nbsp;&amp;nbsp;&amp;nbsp; %QUIT%

# assessment title
SHOW_TITLE_DEF=
_IF %SHOW_TITLE% = 1
 %START_TITLE%%SESSION.TITLE%%END_TITLE%

# assessment completed
SHOW_END_TITLE_DEF=
_IF %SHOW_TITLE% = 1
 %START_TITLE%%SESSION.TITLE% %SESSION_COMPLETED%%END_TITLE%

# assessment introduction
SHOW_INTRO_DEF=
_IF %SHOW_INTRO% = 1
 %START_INTRO%%SESSION.INTRO%%END_INTRO%

# block title
SHOW_BLOCK_TITLE_DEF=
_IF %SHOW_BLOCK_TITLE% = 1
 %START_TITLE%%BLOCK.TITLE%%END_TITLE%

SHOW_BLOCK_INTRO_DEF=
_IF %SHOW_BLOCK_INTRO% = 1
 %START_INTRO%%BLOCK.INTRO%%END_INTRO%

# score for each block
SHOW_BLOCK_SCORE_DEF=
_IF %SHOW_BLOCK_SCORE% = 1
 %START_BLOCK_SCORE% %BLOCK_SCORE%%END_BLOCK_SCORE%

# running (cumulative) score for assessment
SHOW_RUNNING_SCORE_DEF=
_IF %SHOW_RUNNING_SCORE% = 1
 %START_RUNNING_SCORE%%RUNNING_SCORE%%END_RUNNING_SCORE%
```

```
# final score for assessment
SHOW_FINAL_SCORE_DEF=
_IF %SHOW_FINAL_SCORE% = 1
 %START_FINAL_SCORE%%FINAL_SCORE%%END_FINAL_SCORE%

# show status at end of assessment
SHOW_FINAL_STATUS_DEF=
_IF %SHOW_FINAL_STATUS% = 1
 %START_FINAL_STATUS%%SESSION.STATUS%%END_FINAL_STATUS%

# topic name question comes from
SHOW_TOPIC_DEF=
_IF %SHOW_TOPIC% = 1
 %START_TOPIC%%QUESTION.TOPIC%%END_TOPIC%

# show number of questions in question numbering
SHOW_NUMBER_OF_DEF=
_IF %SHOW_NUMBER_OF% = 1
 %NUMBER_OF_TEXT% %BLOCK.NUMQUESTIONS%

# question numbering
SHOW_NUMBER_DEF=
_IF %SHOW_NUMBER% = 1
 %START_NUMBER%%QUESTION.NUMBER% %SHOW_NUMBER_OF_DEF% %END_NUMBER%

# question description
SHOW_DESCRIPTION_DEF=
_IF %SHOW_DESCRIPTION% = 1
 %START_DESCRIPTION%%QUESTION.DESCRIPTION%%END_DESCRIPTION%

# ##########################################
# scoring definitions

# block score shown after each block
BLOCK_SCORE=
%SCOREFONT%
%BLOCK_SCORE_TEXT%  %BLOCK.SCORE%  %SCORE_OUT_OF_TEXT%  %BLOCK.MAX%,
%BLOCK.PERCENT%%%
<BR></FONT>

# cumulative score shown at end of each block
RUNNING_SCORE=
%SCOREFONT%
%SCORE_SO_FAR_TEXT%  %SESSION.SCORE%  %SCORE_OUT_OF_TEXT%  %SESSION.MAX%,
%SESSION.PERCENT%%%
</FONT>

# final score shown at end of assessment
FINAL_SCORE=
%SCOREFONT%
%TOTAL_SCORE_TEXT%  %SESSION.SCORE%  %SCORE_OUT_OF_TEXT%  %SESSION.MAX%,
%SESSION.PERCENT%%%
<BR></FONT>

# assessment outcome and topic scoring
```

```
OUTCOME_DEF=
%START_ASSESSMENT_OUTCOME%
%SESSION.OUTCOME%
%END_ASSESSMENT_OUTCOME%
%SESSION.TOPICS%

# ##########################################
# topic scoring definitions
# for each topic, TOPIC_SCORE then TOPIC_FEEDBACK are written

# topic scoring for a topic
TOPIC_NAME=
%START_TOPIC_NAME%
%TOPIC.DESCRIPTION%
%END_TOPIC_NAME%

# topic scoring for a topic
TOPIC_SCORE=
%START_TOPIC_SCORE%
%TOPIC.PERCENT%%%
%END_TOPIC_SCORE%
%START_TOPIC_SCORE%
%TOPIC.OUTCOME%
%END_TOPIC_SCORE%

# topic feedback for a topic
TOPIC_FEEDBACK=
%START_TOPIC_FEEDBACK%
%TOPIC.FEEDBACK%
%END_TOPIC_FEEDBACK%

# ##########################################
# used to present information to user
MESSAGE=
%MESSAGETOP%
<H2>%SESSION.MESSAGE%</H2>
%MESSAGETAIL%


# ##########################################
#used to present error messages to user
ERROR=
%MESSAGETOP%
%START_BOX%
<H2>%SESSION.ERROR%</H2>
%SESSION.DETAIL%
%END_BOX%
%MESSAGETAIL%


# ##########################################
#used to present error message with retry option to user
RETRY_ERROR=
%MESSAGETOP%
%START_BOX%
%ERROR_RETRY_MESSAGE%
```

```
<FORM NAME="Retry" METHOD="POST" ACTION="%SESSION.URL%" %TARGET%>
%SESSION.MESSAGE% # contains all data submitted to be sent again
%RETRY_BUTTON%
</FORM><br>
%END_BOX%
%MESSAGETAIL%


# ##############################################
# page shown when perception.dll or open.dll redirect the user to session.dll
REDIRECT=
<HTML><HEAD><TITLE>%SESSION.TITLE%</TITLE>
<META HTTP-EQUIV="Refresh" CONTENT="0; URL=%SESSION.URL%">
</HEAD>
<BODY %BACKC% %BACKI% %TEXT% %LINK%>
%HEADER%
%REDIRECT_TEXT%
<P><A HREF="%SESSION.URL%">%REDIRECT_LINK_TEXT%</A>
%TAIL%

# ##############################################
# page shown when session.dll is offline
OFFLINE=
%MESSAGETOP%
%START_BOX%
%OFFLINE_MESSAGE%<br>
<FORM NAME="Offline" METHOD="POST" ACTION="%SESSION.URL%" %TARGET%>
%SESSION.MESSAGE% # contains all data submitted to be sent again
<INPUT TYPE=SUBMIT VALUE="%TRY_AGAIN_TEXT%">
</FORM><br>
%END_BOX%
%MESSAGETAIL%
%TAIL%

# ##############################################
# page for monitored assessments
# show enter button, and optional cancel button
MONITOR=
%MESSAGETOP%
<TABLE border="0" align=center cellPadding="0" cellSpacing="0">
        <TR>
        <TD>
        <TABLE          border="0"          cellPadding="5"          cellSpacing=0          width="100%"
background="%QMP_BACK_GRAPHIC%">
                <tr>
                <td colspan="4">
                <font size="3" color="%BOX_COLOR%"><b>%MONITOR_HEADING%</b></font>
                </td>
                </tr>
                <tr>
                <td colspan="4">
                <font size="3" color="%BOX_COLOR%">%MONITOR_INSTRUCTIONS%</font>
                </td>
                </tr>
                <tr>
                <td colspan="4">
```

```
                     
                    </td>
                    </tr>

                    <TR>
                    <FORM NAME="login" METHOD="POST" ACTION="%SESSION.URL%" >
                    %SESSION.MESSAGE% # contains all data in hidden form fields
                    <TD width=40> </TD>
                    <TD>
                    <font color="%BOX_COLOR%"><b>%MONITOR_NAME%</b></font></TD>
                    <TD><INPUT TYPE="TEXT" MAXLENGTH="50" NAME="MONITORNAME" VALUE=""></TD>
                    <TD width=40> </TD>
                    </TR>

                    <TR>
                    <TD width=40> </TD>
                    <TD>
                    <font color="%BOX_COLOR%"><b>%MONITOR_PASSWORD%</b></font>
                    </TD>
                    <TD><INPUT TYPE="PASSWORD" MAXLENGTH="50" NAME="MONITORPASS"
VALUE=""></TD>
                    <TD width=40> </TD>
                    </TR>

                    <TR>
                    <TD colspan=4>
                    <table border=0 width=100%%>
                    <tr>
                    %MONITOR_BUTTONS_DEF%
                    </tr>
                    </table>
                    </TD>
                    </TR>
          </TABLE>
          </TD>
          <td width="1" bgcolor="%BOX_BACKCOLOR2%"><img alt=" " src="%QMP_SPACE_GRAPHIC%" width="1"
height="1"></td>
          </TR>
          <TR>
          <td colspan="2" height="1" bgcolor="%BOX_BACKCOLOR2%"><img alt=" "
src="%QMP_SPACE_GRAPHIC%" width="1" height="1"></td>
          </TR>
</TABLE>

<script>
<!--
document.login.MONITORNAME.focus();
// -->
</script>
%MESSAGETAIL%
%TAIL%

MONITOR_BUTTONS_DEF=
_IF %SHOW_MONITOR_CANCEL% = 1
 <TD align=middle>
 %MONITOR_BUTTON%
```

```
 </TD>
 </FORM>
 <FORM NAME="END" METHOD="GET" ACTION="javascript:history.back()" %TARGET%>
 <TD align=middle>
 %BACK_BUTTON%
 </TD>
</FORM>
_ELSE
 <TD align=middle>
 %MONITOR_BUTTON%
 </TD>



# ##########################################
# placed at the top of all documents
# includes html tags to start document
TOP=
<HTML>
<HEAD>
<TITLE>%SESSION.TITLE%</TITLE>
%HTML_HEADER%
%SCRIPTS%
</HEAD>
<BODY %BACKC% %BACKI% %TEXT% %LINK% %DO_BODY% %BODY_ATTRIBUTES% topmargin="0">
%HEADER%

# ###################################

MESSAGETOP=
<html>
<head>
<title>%SESSION.TITLE%</title>
<style>
<!--
a.column { color:%BOX_BACKCOLOR3%; text-decoration:none; }
a.column:hover { color:##FF9900; text-decoration:underline; }
a { color:%BOX_COLOR%; text-decoration:none; }
a:hover { text-decoration:underline; color:##FF9900; }
td { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }
tr.list { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }
p.license { font-size:8pt; }
-->
</style>
</head>
<body bgcolor="%BOX_BACKCOLOR3%" text="%BOX_BACKCOLOR2%" link="%BOX_COLOR%"
alink="%BOX_COLOR%" vlink="%BOX_COLOR%" marginwidth="0" marginheight="0" leftmargin="0" topmargin="0" >
<table width=100% cellpadding="0" cellspacing="0" border=0>
        <tr>
        <td colspan=3><a name="top">
        <table cellpadding=0 cellspacing=0 border=0 width=100%>
                <tr>
                <td align="left">%HEADER%</td>
                </tr>
        </table>
        </td>
```

```
<td width="1"><img alt=" " src="%QMP_DOT1_GRAPHIC%" width="1" height="20"></td>
<td width="10"><img alt=" " src="%QMP_SPACE_GRAPHIC%" width="1" height="20"></td>
</tr>
<tr>
<td colspan=3 bgcolor="%BOX_COLOR%"> </td>
<td width="1"><img alt=" " src="%QMP_DOT1_GRAPHIC%" width="1" height="20"></td>
<td width="10"><img alt=" " src="%QMP_SPACE_GRAPHIC%" width="1" height="20"></td>
</tr>
<tr>
<td width=125> </td>
<td colspan=2>
<td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor="%BOX_BACKCOLOR%"
width=1><img alt=" " src="%QMP_DOT1_GRAPHIC%" width="1" height="26"></td></tr></table></td>
<td width="10"> </td>
</tr>
<tr>
<td width=125>
<table cellspacing=0 cellpadding=0 border=0 width=125>
        <tr>
        <td width=124><img alt=" " border="0" src="%QMP_DOT3_GRAPHIC%" width="124"
height="1"></td>
        <td bgcolor="%BOX_BACKCOLOR%" width="1"><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="1"></td>
        </tr>
</table>
</td>
<td colspan=4 bgcolor="%BOX_BACKCOLOR2%"><img alt=" " border="0"
src="%QMP_DOT2_GRAPHIC%" width="1" height="1" alt=""></td>
</tr>
<tr>
<td width=125 valign="bottom" bgcolor="%BOX_BACKCOLOR3%">
<table cellspacing=0 cellpadding=0 border=0 width=125>
        <tr>
        <td width=124><img alt=" " border="0" src="%QMP_DOT3_GRAPHIC%" width="124"
height="1"></td>
        <td bgcolor="%BOX_BACKCOLOR%" width="1"><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="5"></td>
        </tr>
</table>
</td>
<td colspan=2><img alt=" " border="0" src="%QMP_DOT3_GRAPHIC%" width="1" height="1"></td>
<td>
<table cellpadding=0 cellspacing=0 border=0><tr valign=top>
        <td bgcolor="%BOX_BACKCOLOR%" width=1><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="5"></td></tr>
</table></td>
<td><img alt=" " border="0" src="%QMP_DOT3_GRAPHIC%" width="1" height="1"></td>
</tr>
<tr>
<td valign="top" width=125>
<table cellpadding=0 cellspacing=0 border=0 width=125>
        <tr>
        <td width=124 align="center"><img border="0" src="%QMP_LOGO_GRAPHIC%"
alt="%QMP_LOGO_TEXT%" TITLE="%QMP_LOGO_TEXT%" width="%QMP_LOGO_WIDTH%"
height="%QMP_LOGO_HEIGHT%"><br><br></td>
```

```
                    <td bgcolor="%BOX_BACKCOLOR%" width=1><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="1"></td>
                    </tr>
                    <tr>
                    <td align="left">
                    </td>
                    <td bgcolor="%BOX_BACKCOLOR%" width=1><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="1"></td>
                    </tr>
                    <tr>
                    <td bgcolor="%BOX_BACKCOLOR2%"><img alt=" " src="%QMP_DOT2_GRAPHIC%" width="1"
height="1"></td>
                    <td bgcolor="%BOX_BACKCOLOR%" width=1><img alt=" " src="%QMP_DOT1_GRAPHIC%"
width="1" height="1"></td>
                    </tr>
          </table>
          </td>
          <td width="2"> </td>
          <td valign=top width=100%>
          <br>
          <TABLE border=0 cellPadding=0 cellSpacing=0 align=center width="100%">
                    <TR>
                    <td width="10%">
                     
                    </td>
                    <TD align=left>
                    <TABLE border=0 cellPadding=0 cellSpacing=0 width="100%">
                              <tr><td>

                              <!-- end of message top -->

MESSAGETAIL=
                              <!-- start of message tail -->
                              </td></tr>
                              <TR>
                              <TD colspan="2" align="right">
                              <br>
                              <p align="right">%QMP_TEXT%</p>
                              </TD>
                              </TR>
                    </TABLE>
                    </td>
                    <td width="10%">
                     
                    </td>
                    </tr>
          </TABLE>
</table>
</body>
</html>



# ################################################
START_BOX=
<CENTER>
```

```
<table width=70% border=2 bordercolor="%BOX_COLOR%" cellSpacing=0 cellPadding=5>
<tr><td>
<FONT size=+1 face="%TEXTFACE%" color=%TEXTCOLOR%>

END_BOX=
</FONT>
</TD>
</tr>
</table>
</CENTER>

# ##########################################
# placed at the bottom of all documents
# includes html tags to end document
TAIL=
%FOOTER%
</BODY>
</HTML>

# ##########################################
# put timer at top of all timed assessments
TIMED=
_IF %USE_TIMER% = 1
 # use the timer
 %VISIBLE_TIMER%
_ELSE
 # dont use the timer


VISIBLE_TIMER=
_IF %SHOW_TIMER% = 1
 %TEXT_TIMER%
_ELSE
 %STATUS_TIMER%

# timer is visible in text
TEXT_TIMER=
_IF %USE_QXQ% = 1
<DIV ID="timer" CLASS="timer">
%START_TIMER%
%TIME_REMAINING_MESSAGE% <input size=7 name=timespent value=" "><br>
%END_TIMER%
</DIV>
%INIT_TIMER%
%REQUIRE.DO_TIMER%
_ELSE
%TIME_REMAINING_MESSAGE% <input size=7 name=timespent value=" "><br>
%INIT_TIMER%
%REQUIRE.DO_TIMER%

STATUS_TIMER=
# timer will be displayed in status bar
%INIT_TIMER%
%REQUIRE.DO_TIMER%

# empty definition for untimed assessments
```

```
NOTTIMED=

# general definitions

# ##########################################
# access denied message
ACCESS_DENIED=<FONT COLOR="%ACCESS_COLOR%"
SIZE="%ACCESS_SIZE%">%ACCESS_DENIED_TEXT%</FONT><BR>

# assessment has expired
SESSION_EXPIRED=<FONT COLOR="%ACCESS_COLOR%" SIZE="%ACCESS_SIZE%">%SESSION.TITLE%
%ACCESS_EXPIRED_TEXT%</FONT><BR>

# license file limit of assessments completed has been exceeded
LIMIT_EXCEEDED=<FONT COLOR="%ACCESS_COLOR%"
SIZE="%ACCESS_SIZE%">%LIMIT_EXCEEDED_TEXT%</FONT><BR>

# ##########################################
# html syntax for submit button on a form
SUBMIT=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_SUBMIT_GRAPHIC%" border=0 NAME="%SUBMIT_TEXT%"
TITLE="%SUBMIT_TEXT%" CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%SUBMIT_TEXT%" CLASS="navbtn">

# html syntax for reset button on a form
RESET=
_IF %USE_GRAPHIC_BUTTONS% = 1
<A HREF="" border=0 NAME="%RESET_TEXT%" TITLE="%RESET_TEXT%" %DO_RESET_BUTTON%
CLASS="navbtn"><IMG SRC="%QMP_RESET_GRAPHIC%" border=0 ALT="%RESET_TEXT%"></A>
_ELSE
<INPUT TYPE=BUTTON VALUE="%RESET_TEXT%" %DO_RESET_BUTTON% CLASS="navbtn">

# html syntax for continue button on a feedback page
CONTINUE=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_CONTINUE_GRAPHIC%" border=0 NAME="%CONTINUE_TEXT%"
TITLE="%CONTINUE_TEXT%" %DO_CONTINUE_BUTTON% CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%CONTINUE_TEXT%" %DO_CONTINUE_BUTTON% CLASS="navbtn">

# html syntax for quit button on a form
QUIT=
_IF %USE_GRAPHIC_BUTTONS% = 1
<A HREF="" border=0 NAME="%QUIT_TEXT%" TITLE="%QUIT_TEXT%" %DO_QUIT_BUTTON%
CLASS="navbtn"><IMG SRC="%QMP_QUIT_GRAPHIC%" border=0 ALT="%QUIT_TEXT%"></A>
_ELSE
<INPUT TYPE=BUTTON VALUE="%QUIT_TEXT%" %DO_QUIT_BUTTON% CLASS="navbtn">

# home button of final page
HOME_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_HOME_GRAPHIC%" border=0 NAME="%LINK_HOME_TEXT%"
TITLE="%LINK_HOME_TEXT%" %DO_HOME_BUTTON% CLASS="navbtn">
_ELSE
```

```
<INPUT TYPE=SUBMIT VALUE="%LINK_HOME_TEXT%" CLASS="navbtn" %DO_HOME_BUTTON%>

# link to another assessment on final page
BRANCH_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_NEXT_GRAPHIC%" border=0 NAME="%LINK_SESSION_TEXT%"
TITLE="%LINK_SESSION_TEXT%" %DO_BRANCH_BUTTON% CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%LINK_SESSION_TEXT%" CLASS="navbtn" %DO_BRANCH_BUTTON%>

# button to retry for non-fatal errors
RETRY_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_RETRY_GRAPHIC%" border=0 NAME="%TRY_AGAIN_TEXT%"
TITLE="%TRY_AGAIN_TEXT%" %DO_RETRY_BUTTON% CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%TRY_AGAIN_TEXT%" CLASS="navbtn" %DO_RETRY_BUTTON%>

# button for monitor login screen
MONITOR_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_ENTER_GRAPHIC%" border=0 NAME="%MONITOR_START%"
TITLE="%MONITOR_START%" CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%MONITOR_START%" TITLE="%MONITOR_START%" CLASS="navbtn">

# back button to return to previous screen (used to cancel monitoring screen)
BACK_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QMP_GOBACK_GRAPHIC%" border=0 NAME="%MONITOR_BACK%"
TITLE="%MONITOR_BACK%" CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%MONITOR_BACK%" TITLE="%MONITOR_BACK%" CLASS="navbtn">


# the variable %QUESTION.ANSWERED% returns either %ANSWERED% or %NOTANSWERED%
# assign your own text to these here
ANSWERED=
_IF %USE_QXQ% = 1
 %ANSWERED_DEF%
 <SCRIPT>answered = true;</script>
_ELSE
 %ANSWERED_DEF%

ANSWERED_DEF=
_IF %SHOW_QUESTION_SCORE% = 1
 %SCOREFONT%%QUESTION.SCORE% %SCORE_OUT_OF_TEXT% %QUESTION.MAX%</FONT><BR>


NOTANSWERED=
_IF %USE_QXQ% = 1
 %SCOREFONT%%NOTANSWERED_TEXT%</FONT><BR><SCRIPT>answered = false;</script>
_ELSE
 %SCOREFONT%%NOTANSWERED_TEXT%</FONT><BR>

CACHE_TEXT=
```

```
_IF %USE_QXQ% = 1

_ELSE
 %CACHE_PREFIX_TEXT%

# the variable %CHOICE.CHECKED% returns either %CHECKED% or %NOTCHECKED%
# this is to allow selection of radio buttons in forms
CHECKED=CHECKED
NOTCHECKED=

# the variable %CHOICE.SELECTED% returns either %SELECTED% or %NOTSELECTED%
# this is to allow selection of options in forms
SELECTED=SELECTED
NOTSELECTED=

# the variable %SESSION.NEXT% returns
# %LINK_SESSION% if the assessment is branching to another assessment
# %LINK_URL% if the assessment is branching to an external URL
# %LINK_HOME% if the assessment doesn't branch
# use the definitions below to control the layout in these 3 circumstances

# html syntax for returning to where assessment started - must be one line with no ' characters
LINK_HOME=
_IF %USE_QXQ% = 1
<FORM NAME="END" METHOD="%LINK_METHOD%" ACTION="%HOME%"
%TARGET%>%QXQ_HOME_BUTTON%</FORM>
_ELSE
<FORM NAME="END" METHOD="%LINK_METHOD%" ACTION="%HOME%"
%TARGET%>%HOME_BUTTON%</FORM>

# html syntax for branching to another assessment - must be one line with no ' characters
LINK_SESSION=
_IF %USE_QXQ% = 1
<FORM NAME="END" METHOD="POST" ACTION="%SESSION.LINK%"
%TARGET%>%QXQ_BRANCH_BUTTON%</FORM>
_ELSE
<FORM NAME="END" METHOD="POST" ACTION="%SESSION.LINK%"
%TARGET%>%BRANCH_BUTTON%</FORM>

# html syntax for branching to an external URL - must be one line with no ' characters
LINK_URL=<A HREF="%SESSION.LINK%" %TARGET% %DO_LINK_HREF%>%LINK_URL_TEXT%</A>

# html syntax for assessment report link - must be one line with no ' characters
LINK_REPORT=
_IF %REPORT_NAME% =

%REPORT_URL%?di=1&iResult=%SESSION.RESULT%&Participant=%SESSION.USER%&Checksum=%SESSION.
CHECKSUM%
_ELSE

%REPORT_URL%?di=1&iResult=%SESSION.RESULT%&Report_Name=%REPORT_NAME%&Participant=%SESSI
ON.USER%&Checksum=%SESSION.CHECKSUM%

# background color html syntax
BACKI=
_IF %BACKIMAGE% =
```

```
_ELSE
 BACKGROUND="%SERVER.GRAPHICS%%BACKIMAGE%"

# background color syntax
BACKC=BGCOLOR=%BACKCOLOR%

# text color syntax
TEXT=TEXT=%TEXTCOLOR%

# link color syntax
LINK=LINK=%LINKCOLOR% VLINK=%VLINKCOLOR%

# feedback font formatting
FEEDBACKFONTFACE=
_IF %FEEDBACKFACE% =

_ELSE
 FACE="%FEEDBACKFACE%"

FEEDBACKFONT=<FONT COLOR=%FEEDBACKCOLOR% SIZE="%FEEDBACKSIZE%"
%FEEDBACKFONTFACE%>

# score font formatting
SCOREFONTFACE=
_IF %SCOREFACE% =

_ELSE
 FACE="%SCOREFACE%"

SCOREFONT=<FONT COLOR=%SCORECOLOR% SIZE="%SCORESIZE%" %SCOREFONTFACE%>

# for Java question type support - suppress %OPTION% string when no options
OPTION= # do not change

# suppress blank comment labels
COMMENTLABEL=

# ##########################################
# assorted QxQ definitions

STYLE_SHEET=
<SCRIPT>
WriteStyle(%SESSION.TIMELIMIT%, %SHOW_TIMER%);
</SCRIPT>

NO_TIMER_STYLE_SHEET=
<SCRIPT>
WriteStyle(0,0);
</SCRIPT>

WRITE_LIST=
<SCRIPT>
window.setTimeout('WriteList()',100);
</SCRIPT>
```

```
SUBMIT_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_FINISH_GRAPHIC%" width="%QXQ_FINISH_WIDTH%"
height="%QXQ_FINISH_HEIGHT%" border=0 NAME="%FINISH_TEXT%" TITLE="%FINISH_TOOLTIP%"
ID="qsubmit" CLASS="navbtn" onClick="return QxQSubmitNow();">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%FINISH_TEXT%" TITLE="%FINISH_TOOLTIP%" ID="qsubmit" CLASS="navbtn"
>

CONTINUE_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_CONTINUE_GRAPHIC%" width="%QXQ_CONTINUE_WIDTH%"
height="%QXQ_CONTINUE_HEIGHT%" border=0 NAME="%CONTINUE_TEXT%"
TITLE="%CONTINUE_TOOLTIP%" ID="qcontinue" CLASS="navbtn">
_ELSE
<INPUT TYPE=SUBMIT VALUE="%CONTINUE_TEXT%" ID="qcontinue" CLASS="navbtn"
TITLE="%CONTINUE_TOOLTIP%">

QUIT_CONTROL=
_IF %SHOW_QUIT% = 1
%QUIT_CONTROL_DEF%

QUIT_CONTROL_DEF=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_QUIT_GRAPHIC%" width="%QXQ_QUIT_WIDTH%"
height="%QXQ_QUIT_HEIGHT%" border=0 NAME="%QUIT_TEXT%" TITLE="%QUIT_TOOLTIP%" ID="qquit"
CLASS="navbtn" %DO_QUIT_BUTTON%>
_ELSE
 <INPUT TYPE=BUTTON VALUE="%QUIT_TEXT%" ID="qquit" CLASS="navbtn" TITLE="%QUIT_TOOLTIP%"
%DO_QUIT_BUTTON%>

NEXT_CONTROL=%SESSION.NEXT%

UP_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_UP_GRAPHIC%" width="%QXQ_UP_WIDTH%"
height="%QXQ_UP_HEIGHT%" border=0 NAME="%PREVIOUS_TEXT%" TITLE="%PREVIOUS_TOOLTIP%"
onClick="MoveUp();  return false;" ID="qup" CLASS="navbtn">
_ELSE
<INPUT TYPE="BUTTON" VALUE="%PREVIOUS_TEXT%" onClick="MoveUp();" ID="qup" CLASS="navbtn"
TITLE="%PREVIOUS_TOOLTIP%">

DOWN_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_DOWN_GRAPHIC%" width="%QXQ_DOWN_WIDTH%"
height="%QXQ_DOWN_HEIGHT%" border=0 NAME="%NEXT_TEXT%" TITLE="%NEXT_TOOLTIP%"
onClick="MoveDown(); return false;" ID="qdown" CLASS="navbtn">
_ELSE
<INPUT TYPE="BUTTON" VALUE="%NEXT_TEXT%" onClick="MoveDown();" ID="qdown" CLASS="navbtn"
TITLE="%NEXT_TOOLTIP%">

FLAG_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
```

```
<INPUT TYPE=IMAGE SRC="%QXQ_FLAG_GRAPHIC%" width="%QXQ_FLAG_WIDTH%" height="%QXQ_FLAG_HEIGHT%" border=0 NAME="FLAG_TEXT" TITLE="%FLAG_TOOLTIP%" onClick="Flag(); return false;" ID="qflag" CLASS="navbtn">
_ELSE
<INPUT TYPE="BUTTON" VALUE="%FLAG_TEXT%" onClick="Flag();" ID="qflag" CLASS="navbtn" TITLE="%FLAG_TOOLTIP%">

HELP_CONTROL=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_HELP_GRAPHIC%" width="%QXQ_HELP_WIDTH%" height="%QXQ_HELP_HEIGHT%" border=0 NAME="%HELP_TEXT%" TITLE="%HELP_TOOLTIP%" ID="qhelp" CLASS="navbtn" %DO_HELP_BUTTON%>
_ELSE
<INPUT TYPE=SUBMIT VALUE="%HELP_TEXT%" ID="qhelp" CLASS="navbtn" %DO_HELP_BUTTON% TITLE="%HELP_TOOLTIP%">

EMPTY_CONTROL=<br>

NAV_CONTROL=
<DIV CLASS="nav" ID="nav">
</DIV>

TIMER_CONTROL=%SESSION.TIMED%

QXQ_HOME_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_HOME_GRAPHIC%" width="%QXQ_HOME_WIDTH%" height="%QXQ_HOME_HEIGHT%" border=0 NAME="%LINK_HOME_TEXT%" TITLE="%HOME_TOOLTIP%" ID="qhome" CLASS="navbtn" %DO_HOME_BUTTON%>
_ELSE
<INPUT TYPE=SUBMIT VALUE="%LINK_HOME_TEXT%" ID="qhome" CLASS="navbtn" %DO_HOME_BUTTON% TITLE="%HOME_TOOLTIP%">

QXQ_BRANCH_BUTTON=
_IF %USE_GRAPHIC_BUTTONS% = 1
<INPUT TYPE=IMAGE SRC="%QXQ_BRANCH_GRAPHIC%" width="%QXQ_BRANCH_WIDTH%" height="%QXQ_BRANCH_HEIGHT%" border=0 NAME="%LINK_SESSION_TEXT%" TITLE="%BRANCH_TOOLTIP%" ID="qbranch" CLASS="navbtn" %DO_BRANCH_BUTTON%>
_ELSE
<INPUT TYPE=SUBMIT VALUE="%LINK_SESSION_TEXT%" ID="qbranch" CLASS="navbtn" %DO_BRANCH_BUTTON% TITLE="%BRANCH_TOOLTIP%">

# conditional defines to get right formating according to ALIGN setting

START_QXQ_PANEL_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_PANEL%
_ELSE
%START_QXQ_V_PANEL%

END_QXQ_PANEL_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_PANEL%
_ELSE
%END_QXQ_V_PANEL%
```

```
START_QXQ_LIST_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_LIST%
_ELSE
%START_QXQ_V_LIST%

END_QXQ_LIST_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_LIST%
_ELSE
%END_QXQ_V_LIST%

START_QXQ_ENTRY_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_ENTRY%
_ELSE
%START_QXQ_V_ENTRY%

END_QXQ_ENTRY_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_ENTRY%
_ELSE
%END_QXQ_V_ENTRY%

START_QXQ_PREV_BUTTON_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_PREV_BUTTON%
_ELSE
%START_QXQ_V_PREV_BUTTON%

END_QXQ_PREV_BUTTON_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_PREV_BUTTON%
_ELSE
%END_QXQ_V_PREV_BUTTON%

START_QXQ_NEXT_BUTTON_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_NEXT_BUTTON%
_ELSE
%START_QXQ_V_NEXT_BUTTON%

END_QXQ_NEXT_BUTTON_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_NEXT_BUTTON%
_ELSE
%END_QXQ_V_NEXT_BUTTON%

START_QXQ_FLAG_BUTTON_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_FLAG_BUTTON%
_ELSE
%START_QXQ_V_FLAG_BUTTON%

END_QXQ_FLAG_BUTTON_DEF=
_IF %ALIGN% = 2
```

```
%END_QXQ_H_FLAG_BUTTON%
_ELSE
%END_QXQ_V_FLAG_BUTTON%

START_QXQ_FINISH_BUTTON_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_FINISH_BUTTON%
_ELSE
%START_QXQ_V_FINISH_BUTTON%

END_QXQ_FINISH_BUTTON_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_FINISH_BUTTON%
_ELSE
%END_QXQ_V_FINISH_BUTTON%

START_QXQ_QUIT_BUTTON_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_QUIT_BUTTON%
_ELSE
%START_QXQ_V_QUIT_BUTTON%

END_QXQ_QUIT_BUTTON_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_QUIT_BUTTON%
_ELSE
%END_QXQ_V_QUIT_BUTTON%

START_QXQ_HELP_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_HELP_BUTTON%
_ELSE
%START_QXQ_V_HELP_BUTTON%

END_QXQ_HELP_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_HELP_BUTTON%
_ELSE
%END_QXQ_V_HELP_BUTTON%

START_QXQ_TIMER_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_TIMER%
_ELSE
%START_QXQ_V_TIMER%

END_QXQ_TIMER_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_TIMER%
_ELSE
%END_QXQ_V_TIMER%

START_QXQ_CONTINUE_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_CONTINUE%
_ELSE
```

```
%START_QXQ_V_CONTINUE%

END_QXQ_CONTINUE_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_CONTINUE%
_ELSE
%END_QXQ_V_CONTINUE%

START_QXQ_END_DEF=
_IF %ALIGN% = 2
%START_QXQ_H_END%
_ELSE
%START_QXQ_V_END%

END_QXQ_END_DEF=
_IF %ALIGN% = 2
%END_QXQ_H_END%
_ELSE
%END_QXQ_V_END%

IS_FEEDBACK=
<SCRIPT>
        bFeedback = true;
</SCRIPT>

IS_END=
<SCRIPT>
        bEnd = true;
</SCRIPT>


SPOILER=
<DIV CLASS="empty">
</DIV>

QUESTION_TIME=
<INPUT TYPE=HIDDEN NAME="TIME_%QUESTION.ID%" VALUE="">

# suppress right click in IE
NO_CLICK=
_IF %DISABLE_RIGHT_CLICK% = 1
 <SCRIPT LANGUAGE=vbscript>
 Function document_oncontextmenu
   document_oncontextmenu = false
 End Function
 </SCRIPT>
_ELSE
 <!-- -->


# ##############################################################################
# a range of redefined colors
# note the format is COLOR=#RGB, where R, G and B are hexadecimal pairs 00 - FF
# use double ## to 'escape' the # character so it's not treated as a comment
BLACK=##000000
RED=##FF0000
```

```
BLUE=##0000FF
GREEN=##008000
FUSCHIA=##FF00FF
YELLOW=##FFFF00
OLIVE=##808000
PURPLE=##800080
TEAL=##008080
WHITE=##FFFFFF
AQUA=##00FFFF
GRAY=##808080
MAROON=##800000
NAVY=##000080
SILVER=##C0C0C0
ORANGE=##FF8000
DRKBLUE=##004080
LIGHTGREEN=##00FF00

# ##########################################
# define how questions are formatted
[QUESTION]
# ##########################################

QUESTION=
_IF %USE_QXQ% = 1
%SEPARATOR%
<DIV ID="Q_%QUESTION.ID%" CLASS="question">
%QUESTION_HEADER%
%START_QUESTION%
%QUESTION.CONTENT%
%QUESTION_TIME%
%END_QUESTION%
</DIV>
<SCRIPT>
        oTemp = new Question(window, "Q_%QUESTION.ID%", %QUESTION.NUMBER%, iIndex, 'Q', false,
%QUESTION.SCORE%, %QUESTION.MAX%, answered);
        aElements[iIndex] = oTemp;
        aQuestions[%QUESTION.NUMBER%] = oTemp;
        iIndex++;
</SCRIPT>
_ELSE
%SEPARATOR%
%QUESTION_HEADER%
%START_QUESTION%
%QUESTION.CONTENT%<BR>
%END_QUESTION%


EXPLANATION=
_IF %USE_QXQ% = 1
%SEPARATOR%
<DIV ID="Q_%QUESTION.ID%" CLASS="question">
%START_EXPLANATION%
%QUESTION.CONTENT%
%END_EXPLANATION%
</DIV>
<SCRIPT>
```

```
        oTemp = new Question(window, "Q_%QUESTION.ID%", %QUESTION.NUMBER%, iIndex, 'E', false, 0, 0,
false);
        aElements[iIndex] = oTemp;
        iIndex++;
</SCRIPT>
_ELSE
%SEPARATOR%
%START_EXPLANATION%
%QUESTION.CONTENT%<BR>
%END_EXPLANATION%

# ##########################################
# format of comments, or 'write-in' questions
# %ANSWER.COMMENT% will return %COMMENT% if the question has a comment and blank if it has none
COMMENT=
%START_COMMENT%
%START_COMMENTLABEL%
%ANSWER.COMMENTLABEL%
% END_COMMENTLABEL%
<SPAN CLASS="text_area">
<TEXTAREA          NAME="COM_%QUESTION.ID%"          ROWS="%ANSWER.COMMENTHEIGHT%"
COLS="%ANSWER.COMMENTWIDTH%" WRAP=VIRTUAL>
</TEXTAREA>
</SPAN>
%END_COMMENT%


# ##########################################
# define how feedback to questions is formatted
[FEEDBACK]
# ##########################################

# as per the QUESTION definition in [QUESTION]
# but with added feedback
QUESTION=
_IF %USE_QXQ% = 1
%SEPARATOR%
<DIV ID="Q_%QUESTION.ID%" CLASS="question">
%QUESTION_HEADER%
%START_QUESTION_FEEDBACK%
%QUESTION.CONTENT%
%QUESTION.ANSWERED%
%FEEDBACKFONT%
%QUESTION.OUTCOME%
</FONT>
%END_QUESTION_FEEDBACK%
</DIV>
<SCRIPT>
        if (isQxQ) {
                oTemp = new Question(window, "Q_%QUESTION.ID%", %QUESTION.NUMBER%, iIndex, 'Q',
true, %QUESTION.SCORE%, %QUESTION.MAX%, answered);
                aElements[iIndex] = oTemp;
                iIndex++;
        }
</SCRIPT>
_ELSE
```

```
%SEPARATOR%
%QUESTION_HEADER%
%START_QUESTION_FEEDBACK%
%QUESTION.CONTENT%
%QUESTION.ANSWERED%
%FEEDBACKFONT%
%QUESTION.OUTCOME%
</FONT>
%END_QUESTION_FEEDBACK%

# format for each feedback item
FEEDBACK_ITEM=%START_FEEDBACK_ITEM% %OUTCOME.CONTENT% %END_FEEDBACK_ITEM%

# ##########################################
# format of feedback for comments, or 'write-in' questions
# %ANSWER.COMMENT% will return %COMMENT% if the question has a comment and blank if it has none
COMMENT=
%START_COMMENT%
<SPAN CLASS="text_area">
<TEXTAREA NAME="COM_%QUESTION.ID%" ROWS="%ANSWER.COMMENTHEIGHT%"
COLS="%ANSWER.COMMENTWIDTH%" WRAP=VIRTUAL READONLY="true">
%ANSWER.COMMENTVALUE%</TEXTAREA>
</SPAN>
%END_COMMENT%

# ##########################################
# map question type codes to text
[QUESTION TYPE]
# ##########################################

MC=Multiple Choice
MR=Multiple Response
NUM=Numeric
TM=Text Match
SEL=Pull-down list
FIB=Fill in Blanks
HOT=Drag and Drop
MAT=Matrix
ESSAY=Essay
JAVA=Java
MATCH=Matching
RANK=Ranking
SAB=Select a Blank
FLASH=Macromedia Flash

# ##########################################################

# ##########################################
# define answer format
[ANSWER]
# ##########################################

MC=
%START_QUESTION_mc%
%ANSWER.CONTENT%
%END_QUESTION_mc%
```

```
%ANSWER.COMMENT%

MR=
%START_QUESTION_mr%
%ANSWER.CONTENT%
%END_QUESTION_mr%
%ANSWER.COMMENT%

NUM=
%REQUIRE.VALID_NUMBER%
%START_QUESTION_num%
%ANSWER.CONTENT%
%END_QUESTION_num%
%ANSWER.COMMENT%

TM=
%START_QUESTION_tm%
%ANSWER.CONTENT%
%END_QUESTION_tm%
%ANSWER.COMMENT%

FIB=
%START_QUESTION_fib%
%ANSWER.CONTENT%
%END_QUESTION_fib%
%ANSWER.COMMENT%

SEL=
%START_QUESTION_sel%
%ANSWER.CONTENT%
%END_QUESTION_sel%
%ANSWER.COMMENT%

MAT=
%START_QUESTION_mat%
%ANSWER.CONTENT%
%END_QUESTION_mat%
%ANSWER.COMMENT%

HOT=
%START_QUESTION_hot%
%START_QXQ_QUESTION_hot%
%REQUIRE.CHECK_SCRIPT_VERSION%
%REQUIRE.CHECK_APPLETS%
<SCRIPT>
var bFirst = true;
var sChoices = "";
var sContent = "";
var iNum = 1;
</SCRIPT>
%ANSWER.CONTENT%
<br>
<SCRIPT>
document.writeln(sContent);
document.writeln('</applet>');
</SCRIPT>
```

```
</br>
<SCRIPT>
document.writeln(sChoices);
</SCRIPT>
<INPUT TYPE=HIDDEN NAME="QMApplet:QMApplet%QUESTION.NUMBER%" VALUE="%QUESTION.ID%">
%END_QXQ_QUESTION_hot%
%END_QUESTION_hot%
%ANSWER.COMMENT%

ESSAY=
%START_QUESTION_essay%
%ANSWER.CONTENT%
%END_QUESTION_essay%
%ANSWER.COMMENT%

JAVA=
%REQUIRE.CHECK_SCRIPT_VERSION%
%REQUIRE.CHECK_APPLETS%
<P>
%ANSWER.CONTENT%
</P>
%ANSWER.COMMENT%

MATCH=
%REQUIRE.MATCH_CHANGE%
%START_QUESTION_match%
%ANSWER.CONTENT%
%END_QUESTION_match%
%ANSWER.COMMENT%

RANK=
%REQUIRE.MATCH_CHANGE%
%START_QUESTION_rank%
%ANSWER.CONTENT%
%END_QUESTION_rank%
%ANSWER.COMMENT%

SAB=
%REQUIRE.MATCH_CHANGE%
%START_QUESTION_sab%
%ANSWER.CONTENT%
%END_QUESTION_sab%
%ANSWER.COMMENT%

FLASH=
%REQUIRE.DO_FLASH%
%START_QUESTION_flash%
%ANSWER.CONTENT%
%END_QUESTION_flash%
%ANSWER.COMMENT%

# ##########################################
# define format of individual choices within a question
# these will be repeated for all choices in a question
[CHOICE]
# ##########################################
```

```
MC=%START_mc%<INPUT TYPE=RADIO NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%" %DO_MC%
%CHOICE.CHECKED%> %CHOICE.CONTENT%%END_mc%

MR=%START_mr%<INPUT TYPE=CHECKBOX NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%" %DO_MR%
%CHOICE.CHECKED%> %CHOICE.CONTENT%%END_mr%

NUM=%START_num%<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%"><INPUT
TYPE=TEXT NAME="%CHOICE.ID%" SIZE=%CHOICE.WIDTH% MAXLENGTH=%CHOICE.MAXCHARS%
VALUE="%CHOICE.VALUE%" %DO_NUM%> %CHOICE.CONTENT%%END_num%

TM=%START_tm%<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%"><SPAN
CLASS="text_area"><TEXTAREA NAME="%CHOICE.ID%" COLS=%CHOICE.WIDTH%
ROWS=%CHOICE.HEIGHT% WRAP=VIRTUAL %DO_TM%
>%CHOICE.VALUE%</TEXTAREA></SPAN>%END_tm%

FIB=%START_fib%<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%"><INPUT
TYPE=TEXT NAME="%CHOICE.ID%" SIZE=%CHOICE.WIDTH% MAXLENGTH=%CHOICE.MAXCHARS%
VALUE="%CHOICE.VALUE%" %DO_FIB%>%CHOICE.CONTENT%%END_fib%

ESSAY=
%START_essay%
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%"><SPAN CLASS="text_area">
<TEXTAREA NAME="%CHOICE.ID%" COLS=%CHOICE.WIDTH% ROWS=%CHOICE.HEIGHT% WRAP="VIRTUAL"
%DO_ESSAY% >
%CHOICE.VALUE%</TEXTAREA></SPAN>
%END_essay%

SEL=
%START_sel%
<TD>
%CHOICE.CONTENT%
</TD><TD>
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<SELECT NAME="%CHOICE.ID%" %DO_SEL%>
<OPTION VALUE=""></OPTION> # empty option
%CHOICE.OPTION%
</SELECT>
</TD>
%END_sel%

MAT=
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<INPUT TYPE=HIDDEN NAME="%CHOICE.ID%" VALUE="">
%START_mat%
<TD>
%CHOICE.CONTENT%
</TD>
%CHOICE.OPTION%
%END_mat%

HOT=
<SCRIPT>
// main definition just for first choice
if (bFirst) {
        bFirst = false;
```

```
        sContent += '<applet class="qmapplet" id="QMApplet%QUESTION.NUMBER%"
name="QMApplet%QUESTION.NUMBER%" code="QMHotSpot2.class" codebase="%SERVER.JAVA%"
width=%CHOICE.WIDTH% height=%CHOICE.HEIGHT%> \r\n';
        // Applet Question number
        sContent += '<param name="' + iNum++ + '" value="QMApplet%QUESTION.NUMBER%"> \r\n';
        // content of first choice (main image)
        sContent += '<param name="' + iNum++ + '" value="%SERVER.JAVA%%CHOICE.URL%"> \r\n';
        // is this feedback ?
        sContent += '<param name="' + iNum++ + '" value="'
        if ((document.forms[0].name == "FEEDBACK") || (document.forms[0].name == "END")) {
                sContent += 'YES';
        }
        sContent += '"> \r\n';
        // display setting
        sContent += '<param name="' + iNum++ + '" value="%ALIGN_MARKERS%"> \r\n';

}
// then once for each choice
// id of this choice
sContent += '<param name="' + iNum++ + '" value="%CHOICE.ID%">\r\n';
// marker graphic
sContent += '<param name="' + iNum++ + '" value="%SERVER.JAVA%%CHOICE.OPTION%">\r\n';
// location
sContent += '<param name="' + iNum++ + '" value="%CHOICE.VALUE%"> \r\n';
// form fields to indentify each choice with this question
sChoices += '<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%"> \r\n';
sChoices += '<INPUT TYPE=HIDDEN NAME="%CHOICE.ID%" VALUE=""> \r\n';
</SCRIPT>

JAVA=
<SCRIPT>
var iOption = 1;
document.writeln('<applet name="QMApplet%QUESTION.NUMBER%" code="%CHOICE.CONTENT%"
codebase="%SERVER.JAVA%" width="%CHOICE.WIDTH%" height="%CHOICE.HEIGHT%">');
document.writeln('<param name="NAME" value="QMApplet%QUESTION.NUMBER%">');
document.writeln('<param name="VALUE" value="%CHOICE.VALUE%">');
%CHOICE.OPTION%
document.writeln('</applet>');
</SCRIPT>
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<INPUT TYPE=HIDDEN NAME="%CHOICE.ID%" VALUE="">
<INPUT TYPE=HIDDEN NAME="QMApplet:QMApplet%QUESTION.NUMBER%" VALUE="%CHOICE.ID%">

MATCH=
%START_match%
<TD>
%CHOICE.CONTENT%
</TD><TD>
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<SELECT NAME="%CHOICE.ID%" %DO_MATCH%>
<OPTION VALUE=""></OPTION> # empty option
%CHOICE.OPTION%
</SELECT>
</TD>
%END_match%
```

```
RANK=
%START_rank%
<TD>
%CHOICE.CONTENT%
</TD><TD>
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<SELECT NAME="%CHOICE.ID%" %DO_RANK%>
<OPTION VALUE=""></OPTION> # empty option
%CHOICE.OPTION%
</SELECT>
</TD>
%END_rank%

SAB=
%START_sab%
<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">
<SELECT NAME="%CHOICE.ID%" %DO_SAB%>
<OPTION VALUE=""></OPTION> # empty option
%CHOICE.OPTION%
</SELECT>
%END_sab%

FLASH=
<SCRIPT LANGUAGE=JavaScript>
<!--
// Handle all the the FSCommand messages in a specific Flash movie
function %CHOICE.URL%_DoFSCommand(command, args) {
        QM_setInfo("%CHOICE.ID%", command, args);
}
//-->
</SCRIPT>
<SCRIPT LANGUAGE=VBScript>
on error resume next
Sub %CHOICE.URL%_FSCommand(ByVal command, ByVal args)
        call %CHOICE.URL%_DoFSCommand(command, args)
end sub
</SCRIPT>
<SCRIPT LANGUAGE=JavaScript>

if (document.forms[0].name == "QUESTION") {
        document.writeln('<OBJECT classid="%FLASH_ID%" codebase="%FLASH_CODE%" ID=%CHOICE.URL%
WIDTH=%CHOICE.WIDTH% HEIGHT=%CHOICE.HEIGHT%>');
        document.writeln('<PARAM NAME=movie
VALUE="%FLASH_PATH%/%CHOICE.URL%%FLASH_EXT%">');
        document.writeln('<PARAM NAME=quality VALUE=high>');
        document.writeln('<PARAM NAME=bgcolor VALUE=%WHITE%>');
        document.writeln('<EMBED src="%FLASH_PATH%/%CHOICE.URL%%FLASH_EXT%" quality=high
bgcolor=%WHITE% WIDTH=%CHOICE.WIDTH% HEIGHT=%CHOICE.HEIGHT% swLiveConnect=true
NAME=%CHOICE.URL% TYPE="%FLASH_TYPE%" PLUGINSPAGE="%FLASH_PLUG%">');
        document.writeln('</EMBED>');
        document.writeln('</OBJECT>');
        document.writeln('<INPUT TYPE=HIDDEN NAME="%QUESTION.ID%" VALUE="%CHOICE.ID%">');
        document.writeln('<INPUT TYPE=HIDDEN NAME="%CHOICE.ID%" VALUE="">');
}
</SCRIPT>
```

```
# ##########################################
# define format of individual choices within a question
# these will be repeated for all options in a choice
[OPTION]
SEL=<OPTION VALUE="%OPTION.CONTENT%" %OPTION.SELECTED%>%OPTION.CONTENT%</OPTION>

MAT=
<TD><INPUT TYPE=RADIO NAME="%CHOICE.ID%" VALUE="%OPTION.CONTENT%" %DO_MAT%
%OPTION.CHECKED%></TD>
<TD>%OPTION.CONTENT%</TD>

HOT=%OPTION.CONTENT%

JAVA=
document.writeln('<param name="' + iOption + '" value="%OPTION.CONTENT%">');
iOption++;

MATCH=<OPTION VALUE="%OPTION.CONTENT%" %OPTION.SELECTED%>%OPTION.CONTENT%</OPTION>

RANK=<OPTION VALUE="%OPTION.CONTENT%" %OPTION.SELECTED%>%OPTION.CONTENT%</OPTION>

SAB=<OPTION VALUE="%OPTION.CONTENT%" %OPTION.SELECTED%>%OPTION.CONTENT%</OPTION>

# ##########################################


# ##########################################
# map MIME type to styles
[TYPE]
# ##########################################

TEXT=text/plain
HTML=text/html
IMAGE=image/
MULTIMEDIA_EMBED=multimedia/embed
MULTIMEDIA_LINK=multimedia/link
STREAMING_EMBED=streaming/embed
STREAMING_LINK=streaming/link
MATHML=text/mathml
URI=uri

# define formatting for mime types
[STYLE]
# ##########################################
IMAGE=
%START_IMAGE%
<IMG SRC="%SERVER.GRAPHICS%%CONTENT.URL%" %CONTENT.ATTRIBUTES%>
%END_IMAGE%

MULTIMEDIA_EMBED=<EMBED SRC="%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%"
%CONTENT.ATTRIBUTES%><BR>

MULTIMEDIA_LINK=<A
HREF="%SERVER.GRAPHICS%MULTIMEDIA/%CONTENT.URL%">%CONTENT.NAME%</A><BR>
```

STREAMING_EMBED=<EMBED SRC=""mms://ICTOPC10/%CONTENT.URL%" %CONTENT.ATTRIBUTES%>

STREAMING_LINK=<A HREF="mms://ICTOPC10/%CONTENT.URL%">%CONTENT.NAME%</A>

MATHML=<applet class="mathml" codebase="%SERVER.JAVA%" code="webeq.Main" width="%CONTENT.WIDTH%" height="%CONTENT.HEIGHT%" align="middle">
<param name=eq value="%CONTENT.CONTENT%">
<param name=color value="%BOX_BACKCOLOR3%">
<param name=size value="%MATHML_SIZE%">
<param name=parser value="mathml">
</applet>

URI=%START_URI%<A HREF="%CONTENT.URI%" target="QM_URL">%START_URI_CONTENT%%CONTENT.CONTENT%%END_URI_CONTENT%</A>%END_URI%

B. NEW APPLICATIONS

B.1 INDEXMEDIA.ASP

```
<%
' ********************************************************
'    Multimedia Adaptor
'
'
'   Xabier J. Merino Martinez              July 2002
'
' ********************************************************

'ON ERROR RESUME NEXT
%>
<!-- #include file="../functions.asp" -->
<%
QOLCleanUnused
QLLSetSection QLL_CODE_EM_ADMIN

' Set up any local variables to use throughout this page here.
Dim sUserName ' The Current User's validated login name
Dim nuevavar



' Define the local Subs used in this page
Sub WriteLocalStyles()
%>
            p.heading { font-size:16pt; }
<%
End Sub

Sub WriteLocalScript()
%>
            strParticipants = "<%= TX_ADMIN_PARTICIPANTS_STATUS %>";
            strGroups = "<%= TX_ADMIN_GROUPS_STATUS %>";
            strAccounts = "<%= TX_ADMIN_ACCOUNTS_STATUS %>";
            strSchedules = "<%= TX_ADMIN_SCHEDULES_STATUS %>";
            strUsers = "<%= TX_ADMIN_USERS_STATUS %>";
            strSmadmin = "<%= TX_ADMIN_SMADMIN_STATUS %>";
            strServercfg = "<%= TX_ADMIN_SERVERCFG_STATUS %>";
            strBroadcast = "<%= TX_ADMIN_BROADCAST_STATUS %>";
            imgParticipants = new Image();
            imgParticipants.src = "../<%= GX_ADMIN_PARTICIPANTS %>";
            imgGroups = new Image();
            imgGroups.src = "../<%= GX_ADMIN_GROUPS %>";
            imgAccounts = new Image();
            imgAccounts.src = "../<%= GX_ADMIN_ACCOUNTS %>";
```

```
        imgSchedules = new Image();
        imgSchedules.src = "../<%= GX_ADMIN_SCHEDULES %>";
        imgUsers = new Image();
        imgUsers.src = "../<%= GX_ADMIN_USERS %>";
        imgSmadmin = new Image();
        imgSmadmin.src = "../<%= GX_ADMIN_SMADMIN %>";
        imgServercfg = new Image();
        imgServercfg.src = "../<%= GX_ADMIN_SERVERCFG %>";
        imgBroadcast = new Image();
        imgBroadcast.src = "../<%= GX_ADMIN_BROADCAST %>";
        imgParticipants_glo = new Image();
        imgParticipants_glo.src = "../<%= GX_ADMIN_PARTICIPANTS_GLO %>";
        imgGroups_glo = new Image();
        imgGroups_glo.src = "../<%= GX_ADMIN_GROUPS_GLO %>";
        imgAccounts_glo = new Image();
        imgAccounts_glo.src = "../<%= GX_ADMIN_ACCOUNTS_GLO %>";
        imgSchedules_glo = new Image();
        imgSchedules_glo.src = "../<%= GX_ADMIN_SCHEDULES_GLO %>";
        imgUsers_glo = new Image();
        imgUsers_glo.src = "../<%= GX_ADMIN_USERS_GLO %>";
        imgSmadmin_glo = new Image();
        imgSmadmin_glo.src = "../<%= GX_ADMIN_SMADMIN_GLO %>";
        imgServercfg_glo = new Image();
        imgServercfg_glo.src = "../<%= GX_ADMIN_SERVERCFG_GLO %>";
        imgBroadcast_glo = new Image();
        imgBroadcast_glo.src = "../<%= GX_ADMIN_BROADCAST_GLO %>";
        // Changes images and status text
        function hilight(item)
        {
                sTmp = "img" + item + "_glo.src";
                document[item].src = eval(sTmp);
                sTmp = "window.status = str" + item;
                eval(sTmp);
                return true;
        }
        function lolight(item)
        {
                sTmp = "img" + item + ".src";
                document[item].src = eval(sTmp);
                window.status = '';
                return true;
        }

        function help()
        {
                pageHelp("../help/em3_admin_main_help.html");
        }

<%
End Sub

Sub WriteNavString()
%><a href="../index.asp?now=<%= Timer %>" onMouseOver="window.status='<%= TX_HOME_STATUS %>';return
true;" onMouseOut="window.status='';return true;" title="<%= TX_HOME_TITLE %>"><b><%= TX_HOME
%></b><a><font size=2 color="<%= CX_NAVTEXT %>"> ><b>Multimedia Adaptor</b></font><%
End Sub
```

```
Sub WriteSideMenu(nCell)
%>
        <p align="center"> <br><%= TX_CURRENT_USER %><br><i><%= sUserName
%></i><br> </p>
<%
End Sub

Sub WritePageContent()
        Dim iColWidth
        Dim iColImgWidth
        Dim iColSpace

        iColImgWidth="150"
        iColWidth="400"
        iColSpace="60"
        iFooterWidth="90%"
%>


<center><font color="#666666"><H5><b><br><br>The Multimedia Adaptor is a tool for adapting<br>
your assessment containing multimedia to streaming delivery.<br><br>
Introduce the assessment's name or click 'select' to select one from the list.<br><br></b></H5></font>
<FORM METHOD=POST ACTION="http://ictopc8:8080/examples/servlet/PerceptionStreaming" >
<INPUT TYPE=HIDDEN NAME="name1" VALUE="<%= sUserName %>">
<INPUT TYPE=HIDDEN NAME="name2" VALUE="<%= nuevavar %>">
<INPUT TYPE=TEXT NAME="nomassess">
<INPUT TYPE=SUBMIT VALUE="submit">
</form>
<br>
<FORM METHOD=POST ACTION="http://ictopc8:8080/examples/servlet/LookforAssessments" >
<INPUT TYPE=HIDDEN NAME="name1" VALUE="<%= sUserName %>">
<INPUT TYPE=HIDDEN NAME="name2" VALUE="<%= nuevavar %>">
<INPUT TYPE=SUBMIT VALUE="Select the assessment">
</FORM></center>

<%
End Sub

Sub WriteLocalFooter()
End Sub

' Do any initialisation and security checks here.
bValidLicense = False
bValidLicense = ValidLicense(LicenseFilePath)
If (bValidLicense = True) Then
        ' Have we logged on properly.
        sUserName = Request.Cookies("Security")("Name")
        nuevavar = Request.Cookies("Security")("Checksum")
        If (ConfirmLoginDetails(sUserName, Request.Cookies("Security")("Checksum")) = True) Then
                Call BuildPage("../", "QuestionMark Perception - Multimedia Adaptor", GX_V3_LOGO, True, True,
True, 1, "")
        Else
                Call InvalidLogin("../")
        End If
Else
```

```
        Call InvalidLicense(LX_QMARK_LOGO_LINK)
End If
%>
```

B.2 PerceptionStreaming.java

```
import java.io.*;
import java.net.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;


public class PerceptionStreaming extends HttpServlet
{

        public void doPost(HttpServletRequest req, HttpServletResponse res)
                throws ServletException, IOException
        {


        try
        {
        int indice, indice2, indice_comienzo, indice_final, hayenlace, hayembed;
        int parada, long_linea, j, byleidos, bytesle, assessment, correct;
        String mquestion, massessment, linea_mayus, linea_nueva, linea_auxiliar, name, nombre, arch_multimedia;
        String metodostring, topicstring, questionstring, linea_QML, identificadorquestion, arch_multimedia2;
        String assessweb, seguridad1, seguridad2, admin, identidad, nuevotagembed, nuevotaglink;
        char[] href, embed;
        Connection cnnma ,cnnmq, cnna;
        Statement stat;
        ResultSet resultado1, resultado2, resultado3, resultadoa;
        char caracter, metodo;
        PreparedStatement pstmt1, pstmt2, pstmt3, pstmt4, pstma;
        Socket tubo, tubo2;
        PrintStream streamescribir, streamnombre;
        byte arr[] = new byte[1024];
        byte arra[] = new byte[1024];
        File archivonombre, archivo_multim;
        RandomAccessFile rafnombre, raf_enviar;

    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

        href = new char[15];
        href[0] = 'M';
        href[1] = 'U';
        href[2] = 'L';
        href[3] = 'T';
        href[4] = 'I';
        href[5] = 'M';
        href[6] = 'E';
        href[7] = 'D';
        href[8] = 'I';
        href[9] = 'A';
        href[10] = '/';
        href[11] = 'L';
        href[12] = 'I';
```

```
        href[13] = 'N';
        href[14] = 'K';

        embed = new char[16];
        embed[0] = 'M';
        embed[1] = 'U';
        embed[2] = 'L';
        embed[3] = 'T';
        embed[4] = 'I';
        embed[5] = 'M';
        embed[6] = 'E';
        embed[7] = 'D';
        embed[8] = 'I';
        embed[9] = 'A';
        embed[10] = '/';
        embed[11] = 'E';
        embed[12] = 'M';
        embed[13] = 'B';
        embed[14] = 'E';
        embed[15] = 'D';

        nuevotagembed="streaming/embed";
        nuevotaglink="streaming/link";

        arch_multimedia="";
        assessment=0;
        correct=0;

        //rescatar parametros pasados con el metodo post


        res.setContentType("text/html");
        PrintWriter out = res.getWriter();

        if((assessweb=req.getParameter("nomassess"))!=null)
        {
           seguridad1 = req.getParameter("name1");
           seguridad2 = req.getParameter("name2");
        }
        else
        {
    assessweb = req.getParameter("nom");
        seguridad1 = req.getParameter("name3");
        seguridad2 = req.getParameter("name4");
        }

        System.out.println("nombre del assessment:" + assessweb);
        System.out.println("nombre del usuario:" + seguridad1);
        System.out.println("user id:" + seguridad2);
        if(assessweb!=null && seguridad1!=null && seguridad2!=null)
        {

            System.out.println("no faltan parametros");
      //conectar con la base de datos de administracion

           admin ="jdbc:odbc:QM3Security";
```

```
            cnna = DriverManager.getConnection(admin);
            pstma = cnna.prepareStatement("SELECT Password FROM G_User WHERE User_Name = ?");

            pstma.setString(1, seguridad1);
            resultadoa=pstma.executeQuery();

            if(resultadoa.next())
            {
                  identidad = resultadoa.getString(1);
                  System.out.println("a ver si coinciden" + identidad + seguridad2);
               if(identidad.equals(seguridad2))
                  {


                //conectar con la base de datos de los assessments
                     correct=1;
                massessment ="jdbc:odbc:QM3Assessment";
                cnnma = DriverManager.getConnection(massessment);
                     pstmt1 = cnnma.prepareStatement( "SELECT * FROM  S_Item WHERE Block_ID IN ( " +
                                 "SELECT Block_ID FROM S_Index WHERE Session_MID IN ( " +
                                       "SELECT Session_MID FROM S_Header WHERE Session_Name= ? ))");

                     pstmt1.setString(1, assessweb);
                     resultado1=pstmt1.executeQuery();


                //conectar con la otra base de datos para poder leer luego

                mquestion = "jdbc:odbc:QM3Question";
                cnnmq = DriverManager.getConnection(mquestion);


           //ahora se seleccionan las preguntas que hay que procesar

                while (resultado1.next())
                {
                        assessment++;
                   System.out.println("justo despues del while");
                        metodostring = resultado1.getString(2);
                        metodo=metodostring.charAt(0);
                   System.out.println("el metodo es: " + metodo);

                   if(metodo=='1')
                   {
                topicstring = resultado1.getString(4);

                           pstmt2 = cnnmq.prepareStatement("SELECT * FROM Q_QML WHERE Question_MID IN ( "
+
                                                                  "SELECT      Question_MID      FROM      Q_Question
WHERE Topic_ID=?)");


                     pstmt2.setString(1, topicstring);
                           resultado2=pstmt2.executeQuery();
                   }
```

```
else //metodo=2
{
                questionstring=resultado1.getString(6);
                pstmt2 = cnnmq.prepareStatement("SELECT * FROM Q_QML WHERE Question_MID=?");
                pstmt2.setString(1, questionstring);
                resultado2=pstmt2.executeQuery();
      }

      while(resultado2.next())
      {
         linea_QML=resultado2.getString(7);
               identificadorquestion=resultado2.getString(1);
         linea_mayus = linea_QML.toUpperCase ();
         long_linea = linea_QML.length ();

   indice=0;
         indice_comienzo=0;


            while(indice<long_linea)
{
         caracter = linea_mayus.charAt (indice);
         if(caracter==href[0] && indice<long_linea-15)
         {
                  hayenlace=1;
                       hayembed=1;
                  for(indice2=1; indice2 < 15; indice2++)
                  {
                     indice++;
                     caracter = linea_mayus.charAt (indice);

                if(caracter == href[indice2])
                {
                          hayenlace++;
                     }

         if(caracter == embed[indice2])
                  {
                                 hayembed++;
                        }


                 }
                 System.out.println("hay enlace : = " + hayenlace);
                       System.out.println("hay embed : = " + hayembed);
                 if(hayenlace==15 || hayembed==15)
                 {
                    parada=0;
                    j=0;
                    while(parada==0)
                    {
                        indice++;
                        caracter = linea_mayus.charAt (indice);
                        if(caracter=='>')
                        {
                              indice_comienzo=indice;
```

```
                }
                if(caracter=='<')
                {
                        parada=1;
                        System.out.println ("llega aqui");
                }

            }

parada=0;
        indice_final=indice;

        arch_multimedia=linea_QML.substring(indice_comienzo+3,indice_final);



            if (arch_multimedia.charAt(0)!='@')
            {

            arch_multimedia2="@";
            arch_multimedia2=arch_multimedia2.concat(arch_multimedia);


            //enviar el archivo al nuevo directorio para streaming

    tubo = new Socket ("ictopc10", 3500);
    //streamentrada = tubo.getInputStream ();
    streamnombre = new PrintStream (tubo.getOutputStream ());
    archivonombre = new File("nombremedia.txt");
    rafnombre = new RandomAccessFile (archivonombre, "rw");
    rafnombre.writeBytes (arch_multimedia2 + "\n");
    rafnombre.close ();
    rafnombre = new RandomAccessFile (archivonombre, "r");
    while ((bytesle = rafnombre.read (arra, 0, 1024)) > 0)
    {
            streamnombre.write (arra, 0, bytesle);
    }

rafnombre.close ();
    tubo.close ();
    streamnombre.close();

    tubo2 = new Socket ("ictopc10", 3550);
    //streamentrada2 = tubo2.getInputStream ();
    streamescribir = new PrintStream (tubo2.getOutputStream ());
    archivo_multim = new File ("c:/perception3/server/resources/multimedia/" + arch_multimedia);
    raf_enviar = new RandomAccessFile (archivo_multim, "r");

    //arra=arch_multimedia.getBytes();
    //streamnombre.write(arra, 0, 1);
    System.out.println("a ver si hace esto");

    while ((byleidos = raf_enviar.read (arr, 0, 1024)) > 0)
    {
            streamescribir.write (arr, 0, byleidos);
    }
```

```
                    raf_enviar.close ();
                    tubo2.close ();
                    streamescribir.close();
                    archivonombre.delete();


                        //cambiar el enlace en la base de datos

                        linea_nueva=linea_QML.substring(0, indice_comienzo+3);
                        linea_nueva=linea_nueva.concat('@' + arch_multimedia);
                        linea_auxiliar=linea_QML.substring(indice_final, long_linea);
                        linea_nueva=linea_nueva.concat(linea_auxiliar);
                        indice=indice_final+assessweb.length();
                        linea_QML=linea_nueva;
                        System.out.println("la linea cambiada es: " + linea_QML);
                        long_linea=linea_QML.length();
                        linea_mayus = linea_QML.toUpperCase ();

                        System.out.println ("el path es " + arch_multimedia);
                        System.out.println ("indice para seguir" + indice);

                    pstmt3 = cnnmq.prepareStatement("UPDATE Q_QML SET QML_Data= ? WHERE
Question_MID = ? ");
                    pstmt3.setString(1, linea_QML);
                    pstmt3.setString(2, identificadorquestion);
                pstmt3.executeUpdate();
                        pstmt4 = cnnmq.prepareStatement("SELECT QML_Data FROM Q_QML WHERE
Question_MID=?");
                        pstmt4.setString(1, identificadorquestion);
                        resultado3=pstmt4.executeQuery();
                    resultado3.next();


                        }

                        arch_multimedia="";
                    arch_multimedia2="";




//aqui es donde estaba

                    }

            }
                else
                {
                   indice++;
                }
              }
```

```
        }


    }
    if(assessment==0)
    {
            //esto es para el else
            out.println("<HTML>");
            out.println("<head><title>QuestionMark Perception - Multimedia Adaptor</title>");
        out.println("<style><!--");
        out.println("a.column { color:#FFFFFF; text-decoration:none; }");
        out.println("a.column:hover { color:#FF9900; text-decoration:underline; }");
        out.println("a { color:#006699; text-decoration:none; }");
        out.println("a:hover { text-decoration:underline; color:#FF9900; }");
        out.println("td { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");
        out.println("tr.list { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");
        out.println("a.menu { font-family:Verdana,Chicago,Sans-Serif; color:#FFFFFF; text-decoration:none; }");
        out.println("a.menu:hover { color:#FF9900; text-decoration:underline; }");
        out.println("p.heading { font-size:16pt; }");
        out.println("--></style>");
        out.println("</head>");
         out.println("<body bgcolor=\"#FFFFFF\" text=\"#000000\" link=\"#006699\" alink=\"#006699\" vlink=\"#006699\"
marginwidth=\"0\" marginheight=\"0\" leftmargin=\"0\" topmargin=\"0\" >");
        out.println("<table width=100% cellpadding=\"0\" cellspacing=\"0\" border=0>");
        out.println("<tr><td colspan=3><a name=\"top\"><table cellpadding=0 cellspacing=0 border=0 width=100%>");
        out.println("<tr><td align=\"left\"><img src=\"http://localhost/admin/Images/v3_header.gif\" width=\"313\"
height=\"20\" border=\"0\" alt=\"Perception Enterprise Manager header\"></td>");
        out.println("<td align=\"right\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onMouseOver=\"window.status='Log in to Enterprise Manager as a different user'; return true;\"
onMouseOut=\"window.status=''; return true;\" title=\"Questionmark Perception - Enterprise Manager - Login\"><IMG
src=\"http://localhost/admin/Images/link.gif\" alt=\"Log in to Enterprise Manager as a different user\" border=\"0\"
height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");
        out.println("<td align=\"left\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onmouseout=\"window.status=''; return true;\" onMouseOver=\"window.status='Log in to Enterprise Manager as a
different user'; return true;\" title=\"Log in to Enterprise Manager as a different user\">Login</A> </td>");
        out.println("<td align=\"right\"><A href=\"#HELP\" onClick=\"javascript:help();return false;\"
onMouseOver=\"window.status='Get help with this section'; return true;\" onMouseOut=\"window.status=''; return true;\"
title=\"Get help with this section\"><IMG src=\"http://localhost/admin/Images/link.gif\" alt=\"Get help with this section\"
border=\"0\" height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");
        out.println("<td><A href=\"#HELP\" onClick=\"javascript:help();return false;\" onmouseout=\"window.status='';
return true;\" onmouseover=\"window.status='Get help with this section'; return true;\" title=\"Get help with this
section\">Help</A> </td>");
        out.println("</tr></table></td><td width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\"
height=\"20\"></td><td width=\"10\"><img src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\"
height=\"20\"></td></tr>");
        out.println("<tr><td colspan=3 bgcolor=\"#006699\"></td><td width=\"1\"><img
src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"20\"></td><td width=\"10\"><img
src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\" height=\"20\"></td>");
        out.println("</tr><tr><td width=125> </td><td colspan=\"2\" align=\"left\"><table cellspacing=0
cellpadding=0 border=0 width=100%><tr><td align=\"left\" width=100%><nobr><a
href=\"http://localhost/admin/index.asp?now=48418.44\" onMouseOver=\"window.status='Return to the main index
page';return true;\" onMouseOut=\"window.status='';return true;\" title=\"Return to the main index
page\"><b>Home</b><a><font size=2 color=\"#666666\"> <b>Multimedia Adaptor</b></font></nobr></td></tr>");
```

```
        out.println("</table><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"26\"></td></tr></table></td><td width=\"10\"> </td></tr><tr><td width=125><table cellspacing=0 cellpadding=0 border=0 width=125>");
        out.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td>");
        out.println("<td colspan=4 bgcolor=\"#000000\"><img border=\"0\" src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\" alt=\" \"></td></tr><tr><td width=125 valign=\"bottom\" bgcolor=\"#FFFFFF\"><table cellspacing=0 cellpadding=0 border=0 width=125>");
        out.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td>");
        out.println("</tr></table></td><td colspan=2><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td></tr></table></td>");
        out.println("<td><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td valign=\"top\" width=125><table cellpadding=0 cellspacing=0 border=0 width=125><tr>");
        out.println("<td width=124 align=\"center\"><a href=\"http://www.questionmark.com/\" target=\"TOP\" onMouseOver=\"window.status='Questionmark - getting results...';return true;\" onMouseOut=\"window.status='';return true;\"><img border=\"0\" src=\"http://localhost/admin/Images/logo.gif\" alt=\"Visit Questionmark website\" width=\"64\" height=\"96\"><br><br></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");
        out.println("</tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");
        out.println("</tr><tr><td align=\"left\"><p align=\"center\"> <br>Current User<br><i>");
        out.println(seguridad1);
        out.println("</i><br> </p></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td>");
        out.println("<td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td><td width=\"2\"> </td><td valign=top width=100%>");
        out.println("<br><br>");
        out.println("<font color=\"#666666\"><center><h3><b>");
            out.println("********************Error********************<br><br>");
        out.println("</h3><h5>The assessment: ");
        out.println("</h5><h4>");
        out.println(assessweb);
        out.println("</h4><h5>doesn't exit or it hasn't been correctly published<br><br>");
            out.println("Click 'Select' to select an assessment from the list</b></H5></center></font><br>");
            out.println("<center><form method=post action=\"http://localhost:8080/examples/servlet/BuscarAssessments\">");
        out.println("<INPUT TYPE=HIDDEN NAME=\"name1\" VALUE=\"" + seguridad1 + "\">");
            //out.println(seguridad1);
            //out.println(">");
        out.println("<INPUT TYPE=HIDDEN NAME=\"name2\" VALUE=\"" + seguridad2 + "\">");
            //out.println(seguridad2);
            //out.println(">");
            out.println("<INPUT TYPE=SUBMIT VALUE=\"Select\">");
        out.println("</td><td bgcolor=\"#FFFFFF\" width=\"1\"><img src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td width=\"10\"> </td></tr></table>");
        out.println("</body>");
        out.println("</html>");
```

```
      }

}
}
}
if (correct==0)
{
   out.println("<HTML>");
   out.println("<body>");
   out.println("<h3><b>");
   out.println("<img src=\"http://ictopc8/admin/streaming/noauth.gif\")><br>");
   out.println("Sorry!<br>");
   out.println("but you are not authorised to view this page");
   out.println("</b></h3></body>");
   out.println("</html>");

}
else if(correct!=0 && assessment!=0)
{
   out.println("<HTML>");
   out.println("<head><title>QuestionMark Perception - Multimedia Adaptor</title>");
   out.println("<style><!--");
   out.println("a.column { color:#FFFFFF; text-decoration:none; }");
   out.println("a.column:hover { color:#FF9900; text-decoration:underline; }");
   out.println("a { color:#006699; text-decoration:none; }");
   out.println("a:hover { text-decoration:underline; color:#FF9900; }");
   out.println("td { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");
   out.println("tr.list { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");
   out.println("a.menu { font-family:Verdana,Chicago,Sans-Serif; color:#FFFFFF; text-decoration:none; }");
   out.println("a.menu:hover { color:#FF9900; text-decoration:underline; }");
   out.println("p.heading { font-size:16pt; }");
   out.println("--></style>");
   out.println("</head>");
   out.println("<body bgcolor=\"#FFFFFF\" text=\"#000000\" link=\"#006699\" alink=\"#006699\" vlink=\"#006699\"
marginwidth=\"0\" marginheight=\"0\" leftmargin=\"0\" topmargin=\"0\" >");
   out.println("<table width=100% cellpadding=\"0\" cellspacing=\"0\" border=0>");
   out.println("<tr><td colspan=3><a name=\"top\"><table cellpadding=0 cellspacing=0 border=0 width=100%>");
   out.println("<tr><td align=\"left\"><img src=\"http://localhost/admin/Images/v3_header.gif\" width=\"313\" height=\"20\"
border=\"0\" alt=\"Perception Enterprise Manager header\"></td>");
   out.println("<td align=\"right\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onMouseOver=\"window.status='Log in to Enterprise Manager as a different user'; return true;\"
onMouseOut=\"window.status=''; return true;\" title=\"Questionmark Perception - Enterprise Manager - Login\"><IMG
src=\"http://localhost/admin/Images/link.gif\" alt=\"Log in to Enterprise Manager as a different user\" border=\"0\"
height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");
   out.println("<td align=\"left\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onmouseout=\"window.status=''; return true;\" onMouseOver=\"window.status='Log in to Enterprise Manager as a
different user'; return true;\" title=\"Log in to Enterprise Manager as a different user\">Login</A> </td>");
   out.println("<td align=\"right\"><A href=\"#HELP\" onClick=\"javascript:help();return false;\"
onMouseOver=\"window.status='Get help with this section'; return true;\" onMouseOut=\"window.status=''; return true;\"
title=\"Get help with this section\"><IMG src=\"http://localhost/admin/Images/link.gif\" alt=\"Get help with this section\"
border=\"0\" height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");
   out.println("<td><A href=\"#HELP\" onClick=\"javascript:help();return false;\" onmouseout=\"window.status='';  return
true;\" onmouseover=\"window.status='Get help with this section'; return true;\" title=\"Get help with this
section\">Help</A> </td>");
```

```
out.println("</tr></table></td><td width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"20\"></td><td width=\"10\"><img src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\" height=\"20\"></td></tr>");
out.println("<tr><td colspan=3 bgcolor=\"#006699\"></td><td width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"20\"></td><td width=\"10\"><img src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\" height=\"20\"></td>");
out.println("</tr><tr><td width=125> </td><td colspan=\"2\" align=\"left\"><table cellspacing=0 cellpadding=0 border=0 width=100%><tr><td align=\"left\" width=100%><nobr><a href=\"http://localhost/admin/index.asp?now=48418.44\" onMouseOver=\"window.status='Return to the main index page';return true;\" onMouseOut=\"window.status=';return true;\" title=\"Return to the main index page\"><b>Home</b><a><font size=2 color=\"#666666\"> <b>Multimedia Adaptor</b></font></nobr></td></tr>");
out.println("</table><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"26\"></td></tr></table></td><td width=\"10\"> </td></tr><tr><td width=125><table cellspacing=0 cellpadding=0 border=0 width=125>");
out.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td>");
out.println("<td colspan=4 bgcolor=\"#000000\"><img border=\"0\" src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\" alt=\" \"></td></tr><tr><td width=125 valign=\"bottom\" bgcolor=\"#FFFFFF\"><table cellspacing=0 cellpadding=0 border=0 width=125>");
out.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td>");
out.println("</tr></table></td><td colspan=2><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td></tr></table></td>");
out.println("<td><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td valign=\"top\" width=125><table cellpadding=0 cellspacing=0 border=0 width=125><tr>");
out.println("<td width=124 align=\"center\"><a href=\"http://www.questionmark.com/\" target=\"TOP\" onMouseOver=\"window.status='Questionmark - getting results...';return true;\" onMouseOut=\"window.status=';return true;\"><img border=\"0\" src=\"http://localhost/admin/Images/logo.gif\" alt=\"Visit Questionmark website\" width=\"64\" height=\"96\"><br><br></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");
out.println("</tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");
out.println("</tr><tr><td align=\"left\"><p align=\"center\"> <br>Current User<br><i>");
out.println(seguridad1);
out.println("</i><br> </p></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td>");
out.println("<td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td><td width=\"2\"> </td><td valign=top width=100%>");
out.println("<br><br>");
out.println("<font color=\"#666666\"><center><H5><b><br><br>");
out.println("The assessment: ");
out.println("</h5><h4>");
out.println(assessweb);
out.println("</h4><h5>has been correctly adapted for multimedia streaming</b></H5></center></font>");
out.println("</td><td bgcolor=\"#FFFFFF\" width=\"1\"><img src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td width=\"10\"> </td></tr></table>");
out.println("</body>");
out.println("</html>");
```

```
}
}
catch(Exception e)
{
        System.out.println("excepcion" + e);
}
}




}
```

B.3 LookforAssessments.java

```
import java.io.*;
import java.net.*;
import java.sql.*;
import javax.servlet.*;
import javax.servlet.http.*;


public class LookforAssessments extends HttpServlet
{

        public void doPost(HttpServletRequest req2, HttpServletResponse res2)
                throws ServletException, IOException
        {


        try
        {

         ResultSet resultado1, resultado2;
         Connection cnnma, cnnad;
         PreparedStatement pstmt1, pstmt2;
         String nombres, massessment, madmin, name1, name2, sec;
         int filasycolumnas, webpage;

         filasycolumnas=0;
         webpage=1;
         name1=req2.getParameter("name1");
         name2=req2.getParameter("name2");
         Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         res2.setContentType("text/html");

         PrintWriter out2 = res2.getWriter();

         if(name1!=null && name2!=null)
         {

            madmin ="jdbc:odbc:QM3Security";
            cnnad = DriverManager.getConnection(madmin);
            pstmt2 = cnnad.prepareStatement("SELECT User_ID FROM G_User WHERE User_Name = ?");
            pstmt2.setString(1, name1);
            resultado2=pstmt2.executeQuery();

            if(resultado2.next())
            {

                sec=resultado2.getString(1);

        System.out.println("string" + sec + "nada");
                System.out.println("string" + name1 + "nada");

                if(sec.equals(name2))
```

```
                {

                    System.out.println("entra");
                massessment ="jdbc:odbc:QM3Assessment";
                cnnma = DriverManager.getConnection(massessment);
                pstmt1 = cnnma.prepareStatement( "SELECT Session_Name FROM  S_Header");

                resultado1=pstmt1.executeQuery();

                out2.println("<HTML>");
                out2.println("<head><title>QuestionMark Perception - Multimedia Adaptor</title>");


                    out2.println("<style><!--");
                    out2.println("a.column { color:#FFFFFF; text-decoration:none; }");
                    out2.println("a.column:hover { color:#FF9900; text-decoration:underline; }");
                    out2.println("a { color:#006699; text-decoration:none; }");
                    out2.println("a:hover { text-decoration:underline; color:#FF9900; }");
                    out2.println("td { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");
                    out2.println("tr.list { font-family:Arial,Helvetica,Sans-Serif; font-size:10pt; }");


                    out2.println("a.menu { font-family:Verdana,Chicago,Sans-Serif; color:#FFFFFF; text-
decoration:none; }");
                    out2.println("a.menu:hover { color:#FF9900; text-decoration:underline; }");
                    out2.println("p.heading { font-size:16pt; }");
                    out2.println("--></style>");



                out2.println("</head>");




                    out2.println("<body bgcolor=\"#FFFFFF\" text=\"#000000\" link=\"#006699\" alink=\"#006699\"
vlink=\"#006699\" marginwidth=\"0\" marginheight=\"0\" leftmargin=\"0\" topmargin=\"0\" >");

                out2.println("<table width=100% cellpadding=\"0\" cellspacing=\"0\" border=0>");

                out2.println("<tr><td colspan=3><a name=\"top\"><table cellpadding=0 cellspacing=0 border=0
width=100%>");

                out2.println("<tr><td align=\"left\"><img src=\"http://localhost/admin/Images/v3_header.gif\" width=\"313\"
height=\"20\" border=\"0\" alt=\"Perception Enterprise Manager header\"></td>");

                out2.println("<td align=\"right\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onMouseOver=\"window.status='Log in to Enterprise Manager as a different user'; return true;\"
onMouseOut=\"window.status=''; return true;\" title=\"Questionmark Perception - Enterprise Manager - Login\"><IMG
src=\"http://localhost/admin/Images/link.gif\" alt=\"Log in to Enterprise Manager as a different user\" border=\"0\"
height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");

                out2.println("<td align=\"left\"><A href=\"http://localhost/admin/login.asp\" target=\"_top\"
onmouseout=\"window.status=''; return true;\" onMouseOver=\"window.status='Log in to Enterprise Manager as a
different user'; return true;\" title=\"Log in to Enterprise Manager as a different user\">Login</A> </td>");
```

```
out2.println("<td align=\"right\"><A href=\"#HELP\" onClick=\"javascript:help();return false;\" onMouseOver=\"window.status='Get help with this section'; return true;\" onMouseOut=\"window.status=''; return true;\" title=\"Get help with this section\"><IMG src=\"http://localhost/admin/Images/link.gif\" alt=\"Get help with this section\" border=\"0\" height=\"16\" width=\"16\" align=\"absmiddle\"></A></td>");

out2.println("<td><A href=\"#HELP\" onClick=\"javascript:help();return false;\" onmouseout=\"window.status=''; return true;\" onmouseover=\"window.status='Get help with this section'; return true;\" title=\"Get help with this section\">Help</A> </td>");

out2.println("</tr></table></td><td width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"20\"></td><td width=\"10\"><img src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\" height=\"20\"></td></tr>");

out2.println("<tr><td colspan=3 bgcolor=\"#006699\"></td><td width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"20\"></td><td width=\"10\"><img src=\"http://localhost/admin/Images/spacer.gif\" width=\"1\" height=\"20\"></td>");

out2.println("</tr><tr><td width=125> </td><td colspan=\"2\" align=\"left\"><table cellspacing=0 cellpadding=0 border=0 width=100%><tr><td align=\"left\" width=100%><nobr><a href=\"http://localhost/admin/index.asp?now=48418.44\" onMouseOver=\"window.status='Return to the main index page';return true;\" onMouseOut=\"window.status='';return true;\" title=\"Return to the main index page\"><b>Home</b><a><font size=2 color=\"#666666\"> <b>Multimedia Adaptor</b></font></nobr></td></tr>");

out2.println("</table><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"26\"></td></tr></table></td><td width=\"10\"> </td></tr><tr><td width=125><table cellspacing=0 cellpadding=0 border=0 width=125>");

out2.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td>");

out2.println("<td colspan=4 bgcolor=\"#000000\"><img border=\"0\" src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\" alt=\" \"></td></tr><tr><td width=125 valign=\"bottom\" bgcolor=\"#FFFFFF\"><table cellspacing=0 cellpadding=0 border=0 width=125>");

out2.println("<tr><td width=124><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"124\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=\"1\"><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td>");

out2.println("</tr></table></td><td colspan=2><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td><table cellpadding=0 cellspacing=0 border=0><tr valign=top><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"5\"></td></tr></table></td>");

out2.println("<td><img border=\"0\" src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td valign=\"top\" width=125><table cellpadding=0 cellspacing=0 border=0 width=125><tr>");

out2.println("<td width=124 align=\"center\"><a href=\"http://www.questionmark.com/\" target=\"TOP\" onMouseOver=\"window.status='Questionmark - getting results...';return true;\" onMouseOut=\"window.status='';return true;\"><img border=\"0\" src=\"http://localhost/admin/Images/logo.gif\" alt=\"Visit Questionmark website\" width=\"64\" height=\"96\"><br><br></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");
```

```
out2.println("</tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td>");

out2.println("</tr><tr><td align=\"left\"><p align=\"center\"> <br>Current User<br><i>");
    out2.println(name1);
    out2.println("</i><br> </p></td><td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr><tr><td bgcolor=\"#000000\"><img src=\"http://localhost/admin/Images/bk_dot.gif\" width=\"1\" height=\"1\"></td>");

out2.println("<td bgcolor=\"#8C152B\" width=1><img src=\"http://localhost/admin/Images/rd_dot.gif\" width=\"1\" height=\"1\"></td></tr></table></td><td width=\"2\"> </td><td valign=top width=100%>");



    out2.println("<br><br>");
  out2.println("<p align=\"center\">");
  out2.println("<form METHOD=POST ACTION=\"http://ictopc8:8080/examples/servlet/PerceptionStreaming\">");
  out2.println("<table width=\"50%\" cellspacing=\"0\" cellpadding=\"5\" border=\"1\" bordercolor=\"#000000\" background=\"http://ictopc8/admin/reporting/qrvl/images/wks_back.jpg\">");
  out2.println("<tr bgcolor=\"#006699\"><td colspan=\"5\" align=\"center\"><font face=\"Arial\" size=\"2\" color=\"#FFFFFF\">");
  out2.println("<b>Select assessment for adapting</b></font></td></tr>");
  out2.println("<tr><td><table width=\"100%\" border=\"0\" cellspacing=\"0\" cellpadding=\"0\"><tr><td align=\"center\"> </td></tr>");
  out2.println("<tr><td align=\"center\"><table width=\"90%\" bgcolor=\"#000000\" border=\"0\" cellspacing=\"1\" cellpadding=\"5\"><tr>");

  while(resultado1.next())
  {
     nombres = resultado1.getString(1);
     if(filasycolumnas<5)
     {
        out2.println("<td width=\"20%\" valign=\"middle\" height=\"30\" bgcolor=\"#FFFFFF\">");
        out2.println("<input type=\"radio\" checked name=\"nom\" value=\"");
            out2.println(nombres);
  out2.println("\"><br>");
        out2.println(nombres);
        out2.println("</td>");
        filasycolumnas++;
     }
     else
     {
        out2.println("</tr>");
        out2.println("<tr>");
        out2.println("<td width=\"20%\" valign=\"middle\" height=\"30\" bgcolor=\"#FFFFFF\">");
        out2.println("<input type=\"radio\" checked name=\"nom\" value=\"");
        out2.println(nombres);
            out2.println("\"><br>");
        out2.println(nombres);
        out2.println("</td>");
        filasycolumnas=1;
     }
  }
```

```
                    out2.println("</tr></table><br>");
                        out2.println("<input type=\"hidden\" name=\"name3\" value=");
                        out2.println(name1);
                        out2.println(">");
                        out2.println("<input type=\"hidden\" name=\"name4\" value=");
                        out2.println(name2);
                        out2.println(">");
                        out2.println("<input type=submit value=\"Send\" >");
                        out2.println("</td></tr></table></td></tr></table>");
                out2.println("</form>");
                    out2.println("</p>");
                        out2.println("</td><td bgcolor=\"#FFFFFF\" width=\"1\"><img
src=\"http://localhost/admin/Images/wh_dot.gif\" width=\"1\" height=\"1\"></td><td
width=\"10\"> </td></tr></table>");
                    out2.println("</body>");
                out2.println("</html>");

                    cnnma.close();

            }

                    else
                    {
                        webpage=0;
                    }

          }
                else
                {
                        webpage=0;
          }

                cnnad.close();
        }

              else
              {
                  webpage=0;
              }


            System.out.println(webpage);
            if(webpage==0)
            {
                out2.println("<HTML>");
          out2.println("<body>");
          out2.println("<h3><b>");
          out2.println("<img src=\"http://ictopc8/admin/streaming/noauth.gif\"><br>");
          out2.println("Sorry!<br>");
          out2.println("but you are not authorised to view this page");
          out2.println("</b></h3></body>");
          out2.println("</html>");
            }

}
```

```
            catch(Exception e)
            {
                System.out.println("excepcion" + e);
            }

}

}
```

B.3 ServerPerceptionStreaming.java

```
import java.net.*;
import java.io.*;

class ServerPerceptionStreaming
{
	public static void main (String argv[])
	{

	Socket tubo, tubo2;
	ServerSocket tuboservidor, tuboservidor2;
	Thread hiloservidor;
	ServerExtension extension;

	System.out.println ("Servidor:Inicializando al puerto 3500");

	try
	{
	   tuboservidor = new ServerSocket (3500);
	   tuboservidor2 = new ServerSocket (3550);

	   while (true)
	   {
		System.out.println ("Servidor: Esperando una conexion.....");
		tubo = tuboservidor.accept ();
	      tubo2 = tuboservidor2.accept ();
		System.out.println ("Servidor: Conexion recibida.....");

		extension = new ServerExtension (tubo, tubo2);
		hiloservidor = new Thread (extension);
		hiloservidor.start();

	   }

	}

	catch (Exception excepcion)
	{
	   System.err.println ("Servidor: Excepcion en el main: " + excepcion);
	}

	}
}


class ServerExtension implements Runnable
{

	private Socket tubo, tubo2;
	private InputStream streamentrada, streamnombre;
	//private PrintStream streamescribir, streamescribir2;
	RandomAccessFile raf, waf;
	File archivo, archivo2;
```

```
    byte arr[] = new byte[1024];
    byte nom[] = new byte[1024];
    int iNumRead, leidos;
    String nombre;

    public ServerExtension (Socket s, Socket p)
    {

    tubo = s;
    tubo2 = p;

    }

    public void run ()
    {

    System.out.println ("*----------------------------*");
    System.out.println ("extension: Conectando con: " +  tubo.getInetAddress () +
    ":" + tubo.getPort ());

    try
    {

       streamnombre = tubo.getInputStream ();
       archivo = new File ("nombremtmdia.txt");

       raf = new RandomAccessFile (archivo, "rw");

       while (true)
       {

                leidos = streamnombre.read (nom, 0, 1024);

                if (leidos < 0)
                {
                   break;
                }

                raf.write(nom, 0, leidos);

       }
raf.close();

       System.out.println("a ver si llega hasta aqui");


       streamentrada = tubo2.getInputStream ();
       raf = new RandomAccessFile (archivo, "r");
       nombre = raf.readLine ();
       System.out.println("el nombre es: " + nombre);
       archivo2 = new File ("c:/ASFRoot/" + nombre);
       waf = new RandomAccessFile (archivo2, "rw");
       raf.close();
```

```
            while (true)
            {

                    iNumRead = streamentrada.read (arr, 0, 1024);

                    if (iNumRead < 0)
                    {
                       break;
                    }

                    waf.write(arr, 0, iNumRead);

            }

            waf.close();
            tubo2.close();
            streamentrada.close();
            tubo.close();
            streamnombre.close();
            archivo.delete();

        }

        catch (Exception excepcion)
        {
            System.err.println ("extension: Excepcion recibiendo el archivo del cliente: "
            + excepcion);
        }

        System.out.println ("extension: Desconectando: " + tubo.getInetAddress ()
        + ":" + tubo.getPort ());

        System.out.println ("extension: Desconectando2: " + tubo2.getInetAddress ()
        + ":" + tubo2.getPort ());
        System.out.println ("*.....*");

        }

}
```