



# **Question Mark Perception Server**

**Version 2, June 1999**



Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose without the express written permission of Question Mark Computing Ltd.

© 1998-99 Question Mark Computing Ltd. All rights reserved.

## SECOND EDITION

Question Mark and Perception are either registered trademarks or trademarks of Question Mark Computing in the United States of America and the United Kingdom and other countries.

Microsoft, Microsoft Access, Windows, Windows 95, Windows 98, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Other product and company names mentioned herein might be trademarks of their respective owners.





# Contents

## 1 Introduction

1.1 Perception Server	1
1.2 Overview of concepts	1
1.3 What this manual consists of	3

## 2 Installing and configuring Perception Server

### 2.1 How to install

2.1.1 Overview	5
2.1.2 Prepare your system	5
2.1.3 Run the install program	6
2.1.4 Make directories web shareable	7
2.1.5 Sort out permissions	10
2.1.6 Adding your license file	11
2.1.7 Check it works	12

### 2.2 Configuring your installation

2.2.1 What gets installed	13
2.2.2 Changing perception.ini	17
2.2.3 Configuring Enterprise Reporter	24

### 2.3 Advanced installation issues

2.3.1 Installing on other web servers	26
2.3.2 Setting up ODBC manually	27
2.3.3 Installing two copies of Perception on the same PC	30
2.3.4 Using databases in Oracle or SQL Server format	31

### 2.4 Upgrading Perception Server from version 1

2.4.1 Compatibility of data	32
2.4.2 Switching from version 1 to version 2	33

### 2.5 Resolving installation problems

2.5.1 Troubleshooting	34
-----------------------	----

2.5.2 Issues to do with DLL installation _____	35
2.5.3 What to do if the software won't run at all _____	37
2.5.4 Troubleshooting ODBC issues and problems connecting to databases _____	39

### 3 Using and running Perception Server

#### 3.1 Overview and management

3.1.1 How the software works _____	41
3.1.2 Resetting _____	42

#### 3.2 Publishing sessions to the server

3.2.1 Choosing your approach _____	44
3.2.2 Publishing technical issues _____	47

#### 3.3 How participants access sessions

3.3.1 Using perception.dll _____	49
3.3.2 Using open.dll _____	51
3.3.3 Changing the participant login screens _____	53

#### 3.4 Taking sessions

3.4.1 Sequence of events during a session _____	55
3.4.2 Keeping track of progress through a session _____	56
3.4.3 Templates and how server documents are formatted _____	58
3.4.4 Question by question delivery _____	59
3.4.5 Time limits _____	61
3.4.6 Branching at end of session _____	62
3.4.7 Browser issues _____	63
3.4.8 Restarting _____	64

### 4 Security Manager

#### 4.1 Overview

4.1.1 What you can do with Security Manager _____	67
4.1.2 The security database _____	68
4.1.3 How the information in the security database is used _____	71

#### 4.2 Using Security Manager

4.2.1 Running Security Manager _____	72
--------------------------------------	----

4.2.2	Accessing the software (and stopping others from doing so)	73
4.2.3	The Security Manager screens	75
4.2.4	Users explained	76
4.2.5	Viewing and editing users	78
4.2.6	Viewing and editing participants	79
4.2.7	Viewing and editing groups	81
4.2.8	Viewing sessions	82
4.2.9	Making a schedule	83
4.2.10	Viewing schedules	86
<b>4.3</b>	<b>Importing participants from an external file</b>	
4.3.1	Overview	88
4.3.2	Format of imported file	90
<b>4.4</b>	<b>Administration utilities</b>	<b>91</b>

## 5 Enterprise Reporter

<b>5.1</b>	<b>Overview</b>	
5.1.1	Introduction	93
5.1.2	The answer database	93
5.1.3	Logging in	94
5.1.4	The Enterprise Reporter Home page	97
5.1.5	Report Manager	99
<b>5.2</b>	<b>Participants reports</b>	
5.2.1	An example Participant report	100
5.2.2	Choosing a Participant report to view	102
5.2.3	Participant report contents	104
<b>5.3</b>	<b>Session Overview reports</b>	
5.3.1	An example Session Overview report	112
5.3.2	Choosing a Session Overview report to view	113
5.3.3	Session Overview report contents	114
<b>5.4</b>	<b>List reports</b>	
5.4.1	Example List report	116
5.4.2	Choosing to view a List report	118

5.4.3 List report contents	119
<b>5.5 Question Statistics reports</b>	
5.5.1 Example Question Statistics report	124
5.5.2 Choosing to view a Question Statistics report	125
5.5.3 Question Statistics report contents	126
<b>5.6 Survey reports</b>	
5.6.1 Example Survey report	132
5.6.2 Choosing to view a Survey report	133
5.6.3 Survey report contents	134
<b>5.7 Filtering</b>	
5.7.1 How filtering works	139
5.7.2 How to define filtering	141
<b>5.8 Letting participants see reports</b>	
5.8.1 Participant login	143
5.8.2 Calling a Participant report at the end of a session	144
<b>5.9 Advanced topics</b>	
5.9.1 Copying report data into spreadsheets	146
5.9.2 Clearing the answer database	146
5.9.3 What happens if data changes	147
5.9.4 Changing text and graphics used in reports	148
5.9.5 Setting up Perception for local reporting	148
<b>6 Advanced server issues</b>	
6.1 Perception Integration Protocol	151
6.2 Maximizing security	153
6.3 Obtaining participant name from Windows NT	159
6.4 Maximizing the speed of Perception Server	160
6.5 Server variables	164
6.6 The Perception log file format	167



## Appendix A

### Answer database format

A.1 Overview	171
A.2 Results table	172
A.3 Answers table	173
A.4 Session table	175
A.5 Question table	176
A.6 Outcome table	177

## Appendix B

### Security database format

B.1 Overview	179
B.2 Participant table	179
B.3 Group table	180
B.4 Membership table	180
B.5 Schedule table	181
B.6 User table	182

## Appendix C

### Question database format

C.1 Overview	183
C.2 Topic table	183
C.3 Question table	184

## Appendix D

### Session database format

D.1 Overview	187
D.2 Header table	188
D.3 Index table	189

D.4 Question block table	190
D.5 Question reference table	191
D.6 Jump table	192
D.7 Score band table	193

## Appendix E

E.1 Report database format	195
----------------------------	-----

Index	197
-------	-----

# 1 Introduction

## 1.1 Perception Server

Question Mark Perception is a program for creating, delivering and analyzing tests, surveys and assessments over the web or intranets. The software is divided into authoring and server software.

1. Trainers, teachers and other authors create questions and sessions (the name used for tests or surveys created with Perception) using desktop authoring software. These questions and sessions are then uploaded to the web or intranet server.
2. *Perception Server* then runs on the server computer, and controls delivery of these sessions to participants. All access to *Perception Server* is via a browser (for example *Internet Explorer* or *Netscape Navigator*) from a remote machine.

This manual describes *Perception Server*, which has three main components:

- The software that delivers sessions to participants.
- *Security Manager*, software that allows you to set up participants and who is scheduled to take sessions.
- *Enterprise Reporter*, which lets you view and analyze results.

## 1.2 Overview of concepts

If you are new to using Perception, you should read the *Getting Started* manual for a full introduction to Perception. Nevertheless, here is an overview of the concepts aimed at a webmaster who is supporting the installation of *Perception Server*, but who does not need to know all the details.

When an author prepares a Perception session, it consists of the following elements:

- Information on the session itself. Master information on the session is stored inside a session database.
- Questions referenced by the session. These are stored in a question database.

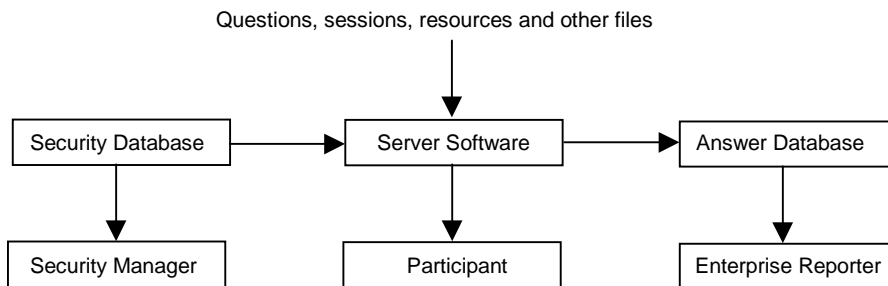
- One or more template files referenced by the session. These are separate files controlling the layout of questions. Authors can choose to use standard template files supplied with the system, or make their own.
- Optionally, resources referenced by questions. These can be GIF or JPEG graphics files, or multimedia files.

During publishing, all these elements are copied to the server. The session and questions need to be copied to the server question and session databases. This can be done by either using software functions to add data into these databases, or else by the author keeping a master copy and replacing the databases each time they are updated. The template and resources files need to be physically copied to the server directories.

Once the session is copied into the server session database, it is available to participants. If *Perception Server* is set up to allow open access to sessions, and the session is set as open, then the new session will appear in the list of sessions offered to participants who access the server. If *Perception Server* is set up to only permit secure access to sessions, then it will only be available if it is scheduled to participants in *Security Manager*.

A session can consist of a single HTML document sent by the server and answered by the participant, but usually will contain a series of blocks of questions sent over a period of time. *Perception Server* keeps track of which participants are doing each session, and sends them more questions and feedback on answers as appropriate.

Answers to questions are stored in a single relational answer database on the server, and you can use *Perception Server*'s reporting software or your own database analysis tools to review the results.



## 1.3 What this manual consists of

This manual describes *Perception Server* in detail. It should be read after reading the *Getting Started* manual, but it can be read independently to the *Authoring manual*.

Chapter 2 describes installing and configuring the server software. You need to follow the instructions in this chapter to set up the software correctly.

Chapter 3 describes the basic process of delivering a session with *Perception Server*, including a description of the primary software that manages the delivery of a session.

Chapter 4 describes the *Security Manager* software that allows you to define participant names and groups, and which sessions they can take to control secure access to sessions.

Chapter 5 describes *Enterprise Reporter*, which lets you to analyze and report on answers to sessions.

Some advanced issues are covered in Chapter 6, and the appendices document the Perception databases.

As well as reading this manual, you are recommended to check the Perception support site for the latest information on *Perception Server*. You can reach this on the web at:

<http://www.questionmark.com/perception/help/> (North American users)

<http://www.qmark.com/perception/help/> (All other users)

The support site includes documentation on some areas not covered in the manuals, including how to write templates, how to use Perception with *SQL Server* and *Oracle* databases and using *Perception Integration Protocol*. It also has a knowledgebase with many questions and answers for Perception users. And if serious bugs are found in the *Perception Server* software, revised versions of the software may be put on the support site for download by Perception users.



## 2 Installing and configuring Perception Server

### 2.1 How to install

#### 2.1.1 Overview

There are six steps to a basic installation of *Perception Server*:

1. Prepare your system. Check you have a suitable webserver and have ODBC installed.
2. Run the server installation program.
3. Make directories web sharable, so your webserver knows where Perception is.
4. Sort out permissions if you need to do this.
5. Add your own license file to replace the temporary one made on installation.
6. Check it works.

See section 2.4 if you are upgrading from version 1 to version 2.

#### 2.1.2 Prepare your system

You need three things to run *Perception Server* – webserver software, ODBC and a browser connected to the webserver.

##### **Webserver software**

For production use of *Perception Server*, you need a Windows NT Server running Microsoft's IIS webserver. For low volume or tryout use, you can use Microsoft's *Personal Web Server* (PWS) on Windows 95, 98 or NT Workstation.

Some information on obtaining the latest version of these Microsoft webservers is on the Perception support website.

## ODBC

*Perception Server* accesses its database files via a technology called ODBC. Your server PC must also have ODBC installed, with a Microsoft *Access* driver of at least version 3.50. This may already be installed on your server. You can check this as follows:

- Go to the *Control Panel* and see if there is a **32-bit ODBC** or **ODBC Data Sources (32bit)** icon.
- If there is one, click on it, and then click the **Drivers** or **ODBC Drivers** tab.
- Check that there is a driver for Microsoft *Access*, and that its version is at least 3.50.

If you do not have this installed, or you want to consider upgrading your version, you should click on the **Install ODBC** option that appears when you put the *Perception* CD in your drive. Or else you can double click on the **installodbc.html** document in the *Perception* CD root directory. This offers you the choice of installing ODBC from the CD or else checking the Microsoft website to see if there is a later version to install.

## Browsers

In order to check that you have installed *Perception Server* successfully, you will need a browser on a PC with a connection to your webserver. Although you can use parts of *Perception Server* with an earlier model browser, for full use you need *Internet Explorer* version 4 or higher, or *Netscape Navigator/Communicator* version 4 or higher.

## 2.1.3 Run the install program

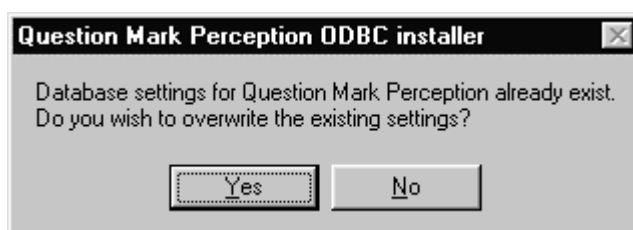
You install *Perception Server* by choosing the option to do so on the menu that appears when you put the *Perception* CD in your drive. Or else, you run the **server.exe** program from the CD root directory. On a Windows NT system, you may need administrative privileges to successfully install the software.

The software will ask you for the directory that you want to install *Perception* in. The default directory is **c:\perception**, but you can set another directory name if you choose. Whatever directory you choose must be accessible to the webserver software. The self-installing program will copy all the files needed for *Perception Server* into the directory you have chosen, and make the necessary subdirectories for you.

If you are installing the authoring and server software on the same PC, you can choose to put them in the same or different directories, as you wish.



The install program will make your Perception databases accessible via ODBC, by allocating them an ODBC **System DSN**. If you have installed *Perception Server* before on the same PC, it will check that you want to overwrite any previous DSNs before doing so. If so, you may see the following message:



If you don't care about any existing installation of Perception, click **Yes**. If you are not sure, click **No**. If you click **No** or get any error messages regarding ODBC installation, you will have to set up your ODBC settings manually as described in section 2.3.2.

## 2.1.4 Make directories web shareable

You now need to tell your webserver that some of the Perception directories are to be accessible over the web. For example, if your website URL is **www.xyzcompany.com**, you might want to access **perception.dll** as **www.xyzcompany.com/qm/perception.dll**. To do this, you have to tell the webserver that the web URL name **qm** maps onto the directory in which you have installed **perception.dll**.

You need to share three directories, as shown in the table below:

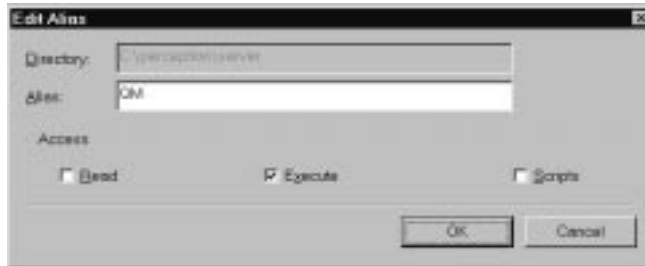
Directory	Suggested share name or web alias	HTTP access needed
c:\perception\server	qm	Execute (only)
c:\perception\qmresources	qmresources	Read (only)
c:\perception\enterprise	enterprise	Read and Scripts (If your webserver does not offer Scripts, choose Execute or Execute Scripts.)

Here is how to set up a web share with IIS 4 under Windows NT Server, or with *Personal Web Server* 4 under Windows 95, 98 or NT Workstation.

1. Make sure that your webserver is running or started.
2. Run *Windows Explorer*, and right-mouse click on the directory you want to share (for example `c:\perception\server`) and choose **Properties**.
3. Select the **Web Sharing** tab. If there is no such tab, then check your webserver is started. Alternatively, you may have another webserver, and need to take alternative steps.



4. Click on **Share this folder**. This brings up a dialog that asks you to enter the name of the share name or alias, and to set the permissions. Set these according to the table above.



5. Click **OK** to set the permissions and **OK** again to confirm the sharing.

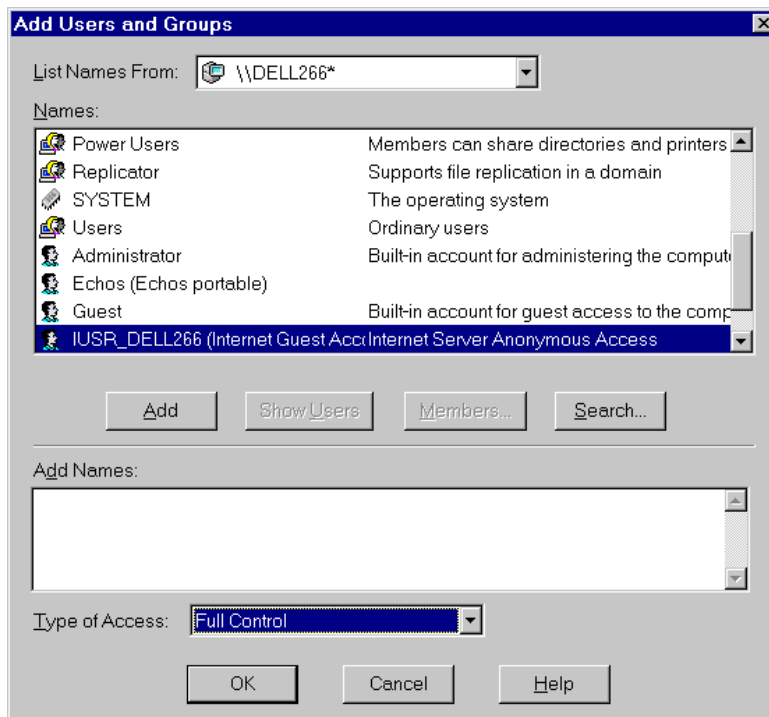
You need to repeat this for each of the three directories.

If you have problems following this approach, you can also share directories under IIS using *Internet Service Manager*. Alternatively, with *Personal Web Server*, you can use *Personal Web Manager*.

## 2.1.5 Sort out permissions

If you are using Windows 95 or 98 you can skip this step.

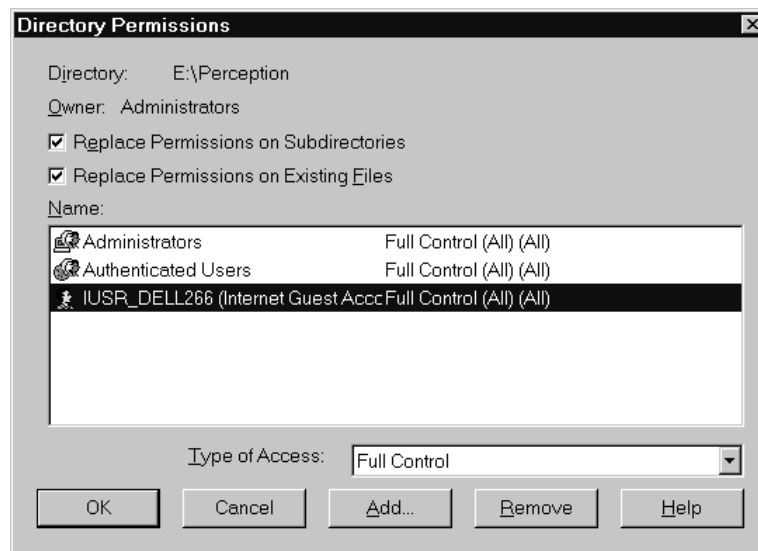
With some Windows NT systems, you may need to set up file and directory permissions to give the server software access. Whether or not you need to do this depends on how your machine is set up. For example, you are likely only to need to do this on NTFS file systems, not on FAT ones.



When IIS or your webserver is installed, a user is created called **IUSR\_YOURMACHINE** where **YOURMACHINE** is the name of your machine. This is also called the Internet Guest Account, but in spite of the name, it doesn't give the end user privileges, only the software they run. When the server software is accessing files, it has the permissions of this user, and so you need to give this user permission to access the Perception directory. To do this, you use *Windows Explorer*, as follows:

1. Right-mouse click on the **c:\perception** directory (or wherever you installed the software), and choose **Properties**, then the **Security** tab, and click on **Permissions**. If there is no **Security** tab, then you probably don't need to set permissions at all.

2. Click the **Add...** button and in the resulting dialog click the **Show Users** button to see a list of all users (not just groups) and select **IUSR\_YOURMACHINE** from the list of users, as in the example above.
3. Set **Type of Access** to **Full Control**, and click **Add** then **OK**.
4. In the **Directory Permissions** dialog, select **Replace Permissions on Subdirectories**, set **Type of Access** to **Full Control** and click the **OK** button.



5. This completes setting the permissions.

If you are particularly concerned about security, you may wish to fine tune the permissions for each subdirectory in the tree. The software needs full control of the **databases**, **working** and **logs** subdirectories, but only read access to the other subdirectories.

## 2.1.6 Adding your license file

*Perception Server*'s usage is controlled by a license file. This controls your system serial number, the number of participants you are permitted to store in the security database and the message displayed at the bottom of each *Perception Server* screen. You cannot change the

license file yourself. They are issued at your Question Mark office, and if you need a different one, you must contact your local office.

License files are usually stored in the same directory as the program DLL files (`\perception\server`), but you can change which directory is used in `perception.ini` if you need to.

*Perception Server* ships with a license file called `license.lic`, which allows default use with a limited number of participants. This license file may expire 30 days after installation. When you purchase Perception, you should be sent either by email or on disk, your own license file. This will have the `.lic` extension, but may have a different name to `license.lic`. When you receive this file:

1. Copy it to the license file directory, usually `\perception\server`.
2. If the server software is running, restart or reset it. See section 3.1.2 for more information about resetting.
3. The new license file will take effect.

## 2.1.7 Check it works

You can then check *Perception Server* works by running a browser and trying to access it. The URL to the programs will be the URL to your server plus `/qm`.

For example, if your server is `www.xyzcompany.com` then the URL to `perception.dll` will be `http://www.xyzcompany.com/qm/perception.dll`. If you are running the software with a browser on the same PC as the server, you can usually access the server with the special URL of `http://localhost/qm/perception.dll`. The use of `localhost` tells the browser to look for a server running on the local machine.

- The program that participants will usually run to take sessions is `perception.dll`. You can try running this. Login with name `test`. Leave the password box blank. You should be able to run the **Welcome to Perception** session.
- The name of the *Security Manager* program is `security.dll`. If you run this, login as **Manager** with password **Secret** (case sensitive). You can change this password from the **Admin** area within the software.

- The third program you can run is an open access program called **open.dll**. If you run this, you will see a list of all sessions in the session database. You are asked for a name, but can type in anything as it's not checked.
- And then you can run *Enterprise Reporter*. You can log in to this from program **login.asp** in the enterprise subdirectory. So for example, you might run it as **http://www.xyzcompany.com/enterprise/login.asp**. You can use the *Security Manager* name and password to get into this. The starting answer database is blank, so you'll need to answer some sessions before there is anything to see.

See section 2.5 for resolving problems installing.

## 2.2 Configuring your installation

### 2.2.1 What gets installed

*Perception Server* installs the following files, all as subdirectories of the master directory chosen on installation (by default **c:\perception**):

#### Main server directory

This directory (usually called **server**) is where the main *Perception Server* programs are run from, and usually contains these files:

- **perception.dll** is the secure entry point for participants into *Perception Server*.
- **open.dll** is the open entry point for participants into *Perception Server*.
- **session.dll** is the part of *Perception Server* that actually delivers the session. Either **perception.dll** or **open.dll** chooses which session to run, and then passes this onto **session.dll** to run it.
- **security.dll** is the *Security Manager* program which controls security .
- **perception.ini** is the configuration file which controls *Perception Server*.

- **license.lic** is a license file that controls use of *Perception Server*. This contains the serial number of your software, and some other important information about your configuration. The software will not run without a license file. A starting version of this file is created during installation. When you purchase the software, you will be sent your own license file (usually with a different name), and you need to copy it to this directory and reset the server DLLs. See section 2.1.6.

### Databases directory

This directory (usually called **databases**) contains the five databases used by the server software. The initial names set on installation are:

- **library.question**, the server question database.
- **master.session**, the server session database.
- **qmpanswer.mdb**, the answer database.
- **qmpsecurity.mdb**, the security database, containing the list of all participants permitted to take sessions and the sessions they are scheduled to.
- **qmpenterprise.mdb**, a database used to hold report definitions within the *Enterprise Reporter* software.

All these databases are supplied in Microsoft *Access* **MDB** format, compatible with *Access* '97. Although the **.question** and **.session** files have a different extension to **.MDB**, they are still in the same file format.

These files are referenced from the software using ODBC, so it is possible to migrate them to different formats (e.g. *SQL Server* or *Oracle*) or rename them if desired, providing the ODBC settings are changed appropriately. See section 2.3.4 for more on this.

### Template and format file directory

This directory (usually called **layout**) contains the files that define the layout used in the HTML produced by *Perception Server*. Changing these allows you to change the way Perception looks within a browser.

- **main.format** is an essential definition file for how questions and sessions look. Files **htmlcore.format**, **text.format**, **style.format** and **scripts.format** are referenced by **main.format**.



- **default.template** is a default template (layout) file. Template files work with the format file to define layout.
- Other **.template** files are also supplied for authors to use in their sessions and for use in the example sessions.
- **login.format** defines the layout of the participant login screens.
- **manager.format** defines the layout used in *Security Manager*.

### Resources directory

This directory (usually called **qmresources**) contains all the graphics and other resource files referenced by questions. It contains these subdirectories by default:

- The **system** subdirectory contains graphics used by the *Perception Server* programs.
- The **multimedia** subdirectory is available to place multimedia resources used by questions.
- The **java** subdirectory is used to place Java programs and the graphics files they need. This includes the graphics and markers used in hotspot questions. The **QMHOTSpot.class** file, which is used to deliver hotspot questions, is copied into this directory, as is the **qxqcore.js** system file referenced by the format file when question by question delivery is on. Don't change or move these two files.
- The **documents** subdirectory is used to hold some documents. This includes **securitymanagerhelp.html**, the *Security Manager* online help file.

Some example graphics and multimedia files are copied on installation for the example sessions databases. Their file names all start with **example**, and they can be deleted if you wish after you have removed the sessions that reference them.

### Perception Integration Protocol directory

This directory (usually called **pip**) is used to hold files used in interfacing Perception to other programs. It contains files which can be referenced when calling Perception from other programs or when using Perception to pass data to other programs.

### Logs directory

A log directory (usually called **logs**) is used to store log files from *Perception Server*.

- **perception.log** contains a log of all accesses to the main server software.
- **perception.err** contains a listing of all errors in accesses to the main server software.
- **pip.log** contains a list of all PIP activity. It is only created if you set PIP logging on in **perception.ini**.

#### **Work in progress directory**

A temporary directory (usually called **working**) is used to store *Perception Server*'s working files. Whenever a session is started, a temporary progress file with extension **.pro** is created in this directory holding the working information *Perception Server* needs to maintain for the session.

#### **Enterprise Reporter directory**

This directory (usually called **enterprise**) contains all the files used by *Enterprise Reporter*, including some in subdirectories. The main files accessed directly are:

- **login.asp** is the main entry point into the reporting software for most users.
- If access to reporting is set up not to require secure login, then **index.asp** can be used as an entry point.
- **plugin.asp** is the entry point for participants accessing reports.
- **settings.asp** contains some settings for *Enterprise Reporter*, that you can be changed.
- **about.asp** is a file which can be called either directly or from *Enterprise Reporter* to give version and other system information.

#### **System DLLs**

These system DLLs needed by *Perception Server* are also copied to your Windows **system** directory:

- **mfc42.dll** and **msvcrt.dll** are Microsoft DLLs needed by all parts of *Perception Server*. The installation software will copy these into your Windows **system** directory, replacing any less up-to-date versions. If your system already contains a more up-to-date version of these DLLs, they will not be updated.

## 2.2.2 Changing perception.ini

The *Perception Server* software relies on the file **perception.ini** to control much of what it does. This file is a configuration file, and by editing it, you can change the configuration of the server. The various settings in **perception.ini**, their default values and what you can set them to are listed below.

The file is usually installed in the **server** subdirectory in the same place as the **perception.dll** and other **.DLL** files. If it's not found here, then it is looked for in the main Windows directory. To edit **perception.ini**, find it in Windows *Explorer* and double click on it, or else open it in *Notepad* or another text editor program. This is an important file, and you should make a back-up copy of it before you make any changes.

The *Perception Server* programs only check the **perception.ini** file on starting or when reset. So after you have edited the file, before the settings take effect, you must either stop and restart your webserver, close down and re-boot your PC or reset the *Perception Server* DLLs. See section 3.1.2 for how to reset the DLLs.

### Directories

These are the full paths on the file system of your webserver where the various Perception files are stored. These are set automatically on installation, and you only need to change them if you are re-structuring the directories. It's important that each directory definition ends with a \ character.

Make sure that the drive letters used are those that are correct for the server PC. For example, if you are accessing the server remotely, you might see part of the server **C:** drive as your **S:** drive. However, the **perception.ini** file should show **C:** not **S:** as it is read by software running on the server PC, not software running on your PC.

**Log directory = c:\perception\logs\**

**Error directory = c:\perception\logs\**

The log directory is where file **perception.log** is stored. The error directory is where file **perception.err** is stored. These are usually the same directory, but can be different.

**License directory = c:\perception\server\**

This is the directory where the license file is stored.

**Temporary directory = c:\perception\working\**

This is the temporary directory where the working progress (.pro) files are stored. One file is made for each session in progress.

**Format directory = c:\perception\layout\**

**Template directory = c:\perception\layout\**

These are the directories where the format and template files are stored. These are usually the same directory but can be different.

**PIP directory=c:\perception\pip\**

This is the directory used for files used in *Perception Integration Protocol* (PIP) that allows Perception to communicate with other programs.

**Format/template file names**

These are the layout file names used by the system, which it expects to find in the above directories. You don't usually need to change these.

**Format file = main.format**

The format file is the master control file for the layout of material created by the server software.

**Template file = default.template**

The template file referenced here is the default template file, and this is over-ridden by whatever is specified in each session.

**Security layout = login.format**

The security layout is used to define the format of the screen when participants log on to the system with **open.dll** and **perception.dll**.

**Manager layout = manager.format**

This file controls the layout of *Security Manager*.

**PIP file=perception.pip**

This is the default PIP file, used if no file is defined.

**Web sharing names**

The following directory settings are web URL paths, not file system paths. They correspond to the web sharing names or aliases that you set up on installing the system. The four entries should all start and finish with */*.

**Graphics directory = /qmresources/**

This is the directory, where graphics files referenced within questions are looked for by the server software. You must copy all graphics referenced in the question database (or from within templates) here. The value to set is the path from the root of your webserver. For example, if the URL of your server is **www.xyzcompany.com**, then the above setting will make a graphics file **picture.gif** be looked for at URL:

**http://www.xyzcompany.com/qmresources/picture.gif.**

**System graphics directory = /qmresources/system/**

This is the directory where the Perception system graphics are stored, for example all the *Security Manager* button graphics.

**Documents directory = /qmresources/documents/**

This is the directory where certain Perception documents are stored, for example the *Security Manager* help file.

**Java applets directory = /qmresources/java/**

This is the directory where all hotspot graphics are stored, and where any Java applets or other files needed by Java applets are stored.

**ODBC DSN names**

**Session database = QMPSession**

**Question database = QMPQuestion**

**Answer database = QMPAnswer**

**Security database = QMPSecurity**

These settings define the ODBC **System DSN** names for the four main server databases. With the above settings, the software looks, for example, for a session database with the ODBC name of **QMPSession**. You can change these if you want to use different ODBC names, see section 2.3.2.

**Database UID = user**

**Database password = secret**

These are only needed if you are using databases under *SQL Server* or *Oracle* rather than *Access*. They are passed along in the ODBC connect string (as UID and PWD). See section 2.3.4 for more information.

### **Logging and debugging settings**

#### **Disable logging = 0**

If you set this to **1**, this will turn off all Perception logging, which will slightly speed up the software. However, the log file can be a very useful resource, especially if things go wrong. Therefore, this should only be set for improved speed once the installation is working perfectly.

#### **Log database activity = 0**

This also should only be adjusted to provide information to Question Mark technical support. Setting it to **1** will add to the log file information on database activity.

#### **Log PIP data = 0**

If you are using PIP, you can set this to **1** to log all PIP transactions. This can be useful in dealing with problems in using PIP. If set, a log file called **pip.log** will be made in the **logs** directory, showing all PIP calls.

#### **Show debug = 0**

#### **Show data = 0**

#### **Show environment = 0**

These three settings should only be adjusted if you need to provide information to Question Mark technical support. Setting each of these to **1** will display debugging information on each server screen.

### **Optimization settings**

#### **Keep database open = 1**

This defines whether or not the software tries to keep open its connections to the databases as much as it can. Setting this to **0** this will significantly slow *Perception Server*. However, it will reduce the number of simultaneous connections to the database, which could have technical or licensing implications.

#### **Template cache size = 30**

This is the number of template files that are held in memory. If you are using more than 30 template files, you might increase this to make reading them faster. If you are changing templates frequently and speed is not vital to you, you could set this to zero; this would mean that templates were always read from hard disk, and never held in memory.

**Preprocess formats = 1**

This sets whether templates and format files are processed (parsed) by the computer when first being read. Having it on will slightly slow the software the first time a block is accessed, but speed up all other accesses. You should usually have it on, i.e. set it to **1**.

**Disable statistics calculation = 1**

In Perception version 1, the answer database kept track of the number of times each session and question was answered and also some statistics about questions and sessions. With version 2, it's recommended to calculate these values in *Enterprise Reporter* when they are needed rather than to store them in the answer database.

To restore compatibility with version 1, for example to continue to use the version 1 *Reporter*, or if you are using these fields for your own purposes, set this to **0**. This will mean that these fields are still calculated, but this will slow down the software.

**Disable database locking = 0**

Only change this if instructed by Question Mark technical support.

**Names of files****Main DLL = session.dll**

This is the name of the main Perception DLL used to run sessions. You would not normally change this.

**Home DLL = perception.dll****Alternative DLL = open.dll**

These define the two DLLs which call **session.dll**. The one in **Home DLL** defines where the **Home** button goes. See section 3.4.6 for more on the **Home** button. If you want this to go to **open.dll**, switch the two around to make:

**Home DLL = open.dll****Alternative DLL = perception.dll****Security help = securitymanagerhelp.html**

This is the name of the *Security Manager* help file, which you would not ordinarily change.

**Server URL= www.questionmark.com**

*Perception Server* sometimes needs to know the address (name or IP address) of the server it is running on, for example when calling one DLL from another. It usually captures this automatically, and it's not necessary to specify it in **perception.ini**.

However, some systems have a number of different names, some of which may only work in some cases. For example, there may be a name that works when accessed locally but which doesn't work from external locations. This might mean that the automatic capture would find the internal name and fail to work when called externally.

If you run into this problem, or need to specify the server URL for some other reason, you can set a fixed server address for everyone, by setting this variable to the server root address, either an IP address or a URL.

This setting does not exist in the shipping **perception.ini** file and will need to be added. It should only have the bare URL - no **http://** prefix and no trailing */*. You can also define the port used here if IIS is running on a non-standard port (i.e. not 80) or if participants come to the server via a non-standard port. For example **Server URL=www.questionmark.com:8080**

### **Security settings**

#### **Enable open access to sessions = 1**

If you are using **perception.dll** to access sessions, you might not want someone to be able to access **open.dll**, which would be a back door through your security. If you want to stop **open.dll** working, set this to **0**.

#### **Disable open list of sessions = 0**

If you want to use **open.dll**, but want to prevent it from giving a list of sessions, you can set this to **1**. This will mean that you can call **open.dll**, but must pass a session ID on the command line; **open.dll** will never generate a list of sessions for people to choose from.

#### **Manager expires = 1**

This is the time in hours before the *Security Manager* password is re-asked.

#### **Sessions expire = 12**

This is the time in hours before buttons for running sessions generated by **perception.dll** time out. By default, any URL created to run a session is valid for twelve hours.

#### **Disable about function = 0**

By default, you can query any of the server programs with a **?about** parameter (for example **www.xyzcompany.com/qm/perception.dll?about**, which will give you some information about the server settings. This information is useful in checking the server is running correctly, but



could be of value to someone trying to break into the system. To disable the **about** command from working, set this to **1**.

**Disable restart function = 0**

If set to **1**, this prevents use of manual restart, which allows participants using **perception.dll** to restart their interrupted sessions. See section 3.4.8 for more on restarting sessions.

**Use no cache header = 0**

This disables the back button in the browser, which is usually recommended. Set it to **1** to allow this.

**Server key = abcdefg**

This key can be any string of characters or numbers. A minimum of 6 and maximum of 32 characters is recommended. It is used as part of the process of generating access IDs and other numbers in the server security system. Setting this to a value of your own will give you extra security against people trying to defeat your security. Even if they know the algorithm used, unless they know this key, they cannot use it.

**Functional settings**

**Clean up = 12**

When the server software is reset, all temporary progress (**.pro**) files are deleted that have not been modified for longer than this clean up time in hours. This means that any session that has been in progress for 12 or more hours by default will be assumed abandoned when the software is reset.

If you do not want sessions to be abandoned in this way, for example you want people to be able to do sessions over a period of days or more, then you should set this to **0**.

**Timing leeway = 120**

This sets the timing leeway in seconds, which extends the time limit for sessions to allow for delays in the client/server communication process. See section 3.4.5 for more information on this issue.

**Enable automatic restart function = 0**

This enables automatic restart. See section 3.4.8 for information on automatic restart, and only set this on when using **perception.dll**, not when using **open.dll**.

**Call PIP file = 0**

Setting this to **1** allows you to call a PIP file every time a session is run. Only set this if you are sure that this is what you want. See section 6.1 or the Perception support site for more details about PIP.

## 2.2.3 Configuring Enterprise Reporter

*Enterprise Reporter* is controlled by a different configuration file, **settings.asp**, which is in the `\perception\enterprise` directory. This is also a text file that you can edit in *Notepad* or another text editing program. Take care that you do not edit it with *FrontPage*. If you have *FrontPage* loaded on your system, it's possible that double clicking on a **.asp** file will run *FrontPage*, but this program can add unwanted codes into the file, and should not be used.

**Setting database DSNs**

This section defines the ODBC DSN names used by *Enterprise Reporter*. *Enterprise Reporter* does not read **perception.ini**, so if you change the ODBC DSNs in **perception.ini**, you also need to change them here as well. These are the default settings:

```
Answer_Database = "QMPAnswer"  
Report_Database = "QMPEnterprise"  
Security_Database = "QMPSecurity"
```

If you are using *SQL Server* or *Oracle*, you may also need to pass a user ID and password into these databases. The syntax for this is:

```
Answer_Database = "DSN=QMPAnswer; UID=user; PWD=password"
```

**Setting security for normal logon**

```
CF_Secure_Access = 1
```

When installed, *Enterprise Reporter* by default requires all users to log on using the names and passwords set in *Security Manager*. This setting controls whether users of *Enterprise Reporter* need to log in. If you set this to **0**, then users can run *Enterprise Reporter* without having to log in first.

Sometimes you might want to restrict access to *Enterprise Reporter* itself, but allow people to view reports with predefined parameters without security. You can permit this for each report type by changing these variables.

**CF\_View\_PRRReport = 1** ' Participant Report  
**CF\_View\_SORReport = 1** ' Session Overview Report  
**CF\_View\_QSReport = 1** ' Question Statistics Report  
**CF\_View\_SRRReport = 1** ' Survey Report  
**CF\_View\_LSReport = 1** ' List Report

If any of these variables are **0**, then even if **CF\_Secure\_Access** is **1**, it's possible to view the report without security. This allows someone to see a report, and then copy the URL to others, so they can see it too.

### Setting security for participant logon

Three settings control whether participants can run *Enterprise Reporter* via **login.asp** to see reports on their own results. See section 5.8 for how this works.

#### **CF\_pview = 1**

This controls whether participant viewing of reports is allowed or not. There are three possible settings:

- If this is **0**, then participants cannot view reports on their results. Set this if you don't want participants to be able to see reports.
- If this is **1**, the default value, then participants can view reports. They log in using the name and password set in *Security Manager*, i.e. the same name and password used to log in to **perception.dll**.
- If this is **2** then participants can log in just by quoting their name, without any password. This means that a participant can see anyone's results if they know their name. Only set this if you don't want results to be confidential.

#### **CF\_preport\_definition = "Example Report (On Screen)"**

If participant reports are allowed, you define here which report definition the participants see. It must be a report that exists in the report database.

#### **CF\_pend\_session = 1**

As described in section 5.8.2, you can call a participant report automatically at the end of a session. If you set this to **0**, you can disable this facility.

### Other settings

#### **CF\_Disable\_icons = 0**

Setting this to **1** allows you to disable the **Home** and **Help** buttons from within *Enterprise Reporter*. This can be useful if you want to pass URLs around to people, without allowing them to navigate.

#### **CF\_Timeout = 300**

This sets the time in seconds that each part of the reporting software is allowed to run for before the webserver will terminate it as having taken too long.

## 2.3 Advanced installation issues

### 2.3.1 Installing on other web servers

The *Perception Server* DLLs use the ISAPI standard for webserver software. ISAPI is a Microsoft standard for web servers. In theory, you can use the DLLs on any ISAPI web server. In practice at the time of writing, the only widely used ISAPI web servers are Microsoft's *Internet Information Server* (IIS) which is shipped as part of Windows NT Server, and its junior cousin *Personal Web Server* (PWS).

The *Enterprise Reporter* software uses *Active Server Pages* (ASP). IIS and PWS support ASP, and in theory you could run this software on another web server which also supports ASP.

*Perception Server* does *not* run on UNIX servers. If your organization is UNIX based, you will need either to get an additional NT server to run Perception, or else consider another product to do your assessment on.

*Perception Server* also does *not* run on non-ISAPI compliant Windows NT servers, for example Lotus Domino or Netscape servers. However it is usually possible to run IIS and another web server on different ports on the same PC, and this is what many organizations do who want to use IIS and another NT web server. You simply load IIS onto one port and, for example Domino on another port, and both web servers will run in conjunction. Then you install and run Perception under IIS. Since IIS is supplied free with Windows NT Server, this doesn't cost anything extra, and is the recommended route.

If you follow this approach, you may need to define the **Server URL** in **perception.ini** as explained in section 2.2.2.

## 2.3.2 Setting up ODBC manually

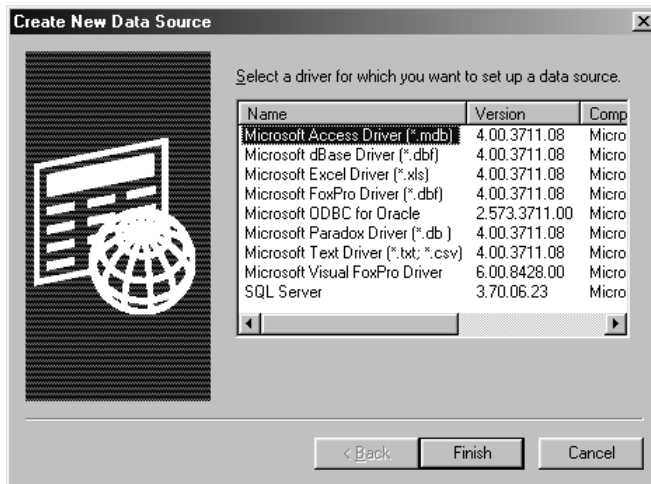
With ODBC, programs reference databases using Data Source Names (DSNs). These are translated into physical databases by the ODBC management system. For example, *Perception Server* finds the answer database by using a DSN called **QMPAnswer**. It doesn't need to know where the physical file is. However, the ODBC system does need to know this, i.e. that the physical file is called **qmpanswer.mdb**, and is stored in directory **c:\perception\databases**.

Usually the install program automatically sets up these DSNs for you. But if there are problems in installing, or if you have more than one copy of *Perception Server* on the PC, you may need to set these up manually. You need to set up five connections, as follows:

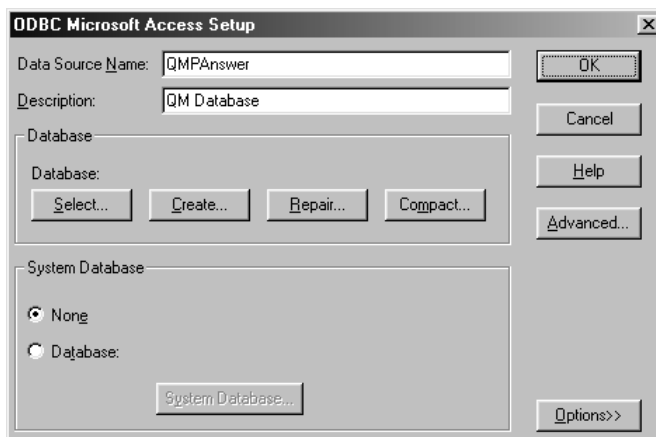
Data Source Name	File Name
<b>QMPAnswer</b>	<b>qmpanswer.mdb</b>
<b>QMPEnterprise</b>	<b>qmpenterprise.mdb</b>
<b>QMPSecurity</b>	<b>qmpsecurity.mdb</b>
<b>QMPQuestion</b>	<b>library.question</b>
<b>QMPSession</b>	<b>master.session</b>

To make each connection, you do the following:

1. Run the **ODBC Data Sources (32bit)** or **32-bit ODBC** icon from the *Control Panel*.
2. Select the **System DSN** tab on the ODBC dialog.
3. Click on the **Add...** button to make a new connection.
4. Choose Microsoft *Access* from the list of drivers, and click **Finish**.

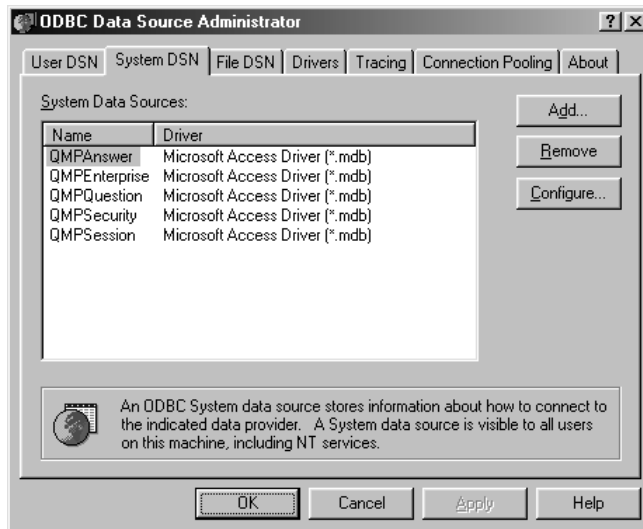


- In the following dialog, set the **Data Source Name** (e.g. **QMPAnswer**), a description (any text, e.g. **QM Database**), and use the **Select** button to browse for the database file name (e.g. **qmpanswer.mdb**).



For the question and session databases (the last two settings), you will need to list files of all type in the browse dialog to find the **.question** and **.session** files to connect to. These are *Access* databases but do not use the **.MDB** extension. Do not make any settings in the **System Database** area; leave this as **None**.

When you are finished the **System DSN** dialog should look similar to the following screen:



If you change the DSN names used by Perception, you will need to change the configuration files to let Perception know of the change. You should change **perception.ini** as explained in section 2.2.2, and change the reporting **settings.asp** as explained in section 2.2.3.

In case you need this information, here is which of the different components of *Perception Server* access which of the databases:

- The security database is accessed by **security.dll**, **perception.dll**, **open.dll**, **session.dll** ,and *Enterprise Reporter*.
- The session database is accessed by **security.dll**, **perception.dll**, **open.dll** and **session.dll**.
- The question database is accessed only by **session.dll**.
- The answer database is accessed by **session.dll** and *Enterprise Reporter*.
- The reports database is accessed only by *Enterprise Reporter*.

### 2.3.3 Installing two copies of Perception on the same PC

It is possible to install more than one copy of *Perception Server* on the same server. There are no technical limits to the number of installations, but there are usually licensing limitations. You are normally only licensed to have a single copy of *Perception Server* in production use, with an additional copy for tryout or backup use. So if you want to install multiple copies, please check your license agreement, and make any necessary additional purchase. Reasons for wanting multiple copies include:

- Each copy can have its own separate databases, so if you have multiple copies, you can have different databases. This might be helpful if you have different departments using Perception, and want to keep their databases separate.
- You might want to set up one secure server (where sessions are only available if they have been scheduled) and one open access server (where sessions are available to all).
- You might want to set up one server as a trial server to tryout material and another as a release server for real use.

When you are installing multiple copies, you follow the steps in section 2.1 for each copy you are installing. You should install each copy into a separate directory. For example, you might install the software into directories **c:\perception\_a**, **c:\perception\_b** and **c:\perception\_c**. In addition, you must choose different web sharing names for each installation. For example, you might choose web sharing names **servera**, **serverb** and **serverc**.

You must then take these additional steps for each installation other than the first.

1. Each installation comes with five databases, which must be mapped to ODBC names. The installation software usually sets up these mappings for you for the first installation. However, you need to set these manually for the second and subsequent installations.

When the installation software asks you if you want to override existing ODBC settings, say no. Then set them up manually as described in section 2.3.1 above.

2. Each installation of *Perception Server* has its own separate copy of **perception.ini**. This is stored in the same directory as the DLLs. You will need to modify **perception.ini** for each installation, and change the settings as follows:
  - The **[Web server]** section needs all references to the web sharing name **qmresources** changed to whatever name you are using.



- The **[Databases]** section needs the ODBC DSN names to be set appropriately.
3. You will also need to edit file **settings.asp** in the **enterprise** directory to replace the DSN names for the answer, security and report databases.

## 2.3.4 Using databases in Oracle or SQL Server format

A default installation of *Perception Server* uses the database in the Microsoft *Access* format they are shipped in. However, all reading and writing to databases by the server software is via ODBC, so it is possible to convert these databases to formats such as *SQL Server* or *Oracle*, and still use *Perception Server*.

The *Access* drivers will give good performance for small or medium size systems. However, if you want to handle large numbers of sessions and participants, particularly with many participants using the system simultaneously, then you can increase the speed and reliability of the system by using *SQL Server* or *Oracle*. This can also make the system more secure.

Whatever you do with your server databases, authors continue to use databases in *Access* format to develop their questions and sessions. This avoids them needing to deal with the complex setup issues that can be associated with professional database engines. They then can publish to the server databases by publishing to them from *Session Manager* using ODBC.

See the *Authoring manual* for more information on publishing to ODBC. This is an extra chargeable facility, and you need a specially configured license file to be able to publish to ODBC, and so to put your databases in *SQL Server* or *Oracle* formats.

In order to use a different database engine with *Perception Server*:

1. You must load this database engine, and install appropriate database drivers.
2. You must set up the Perception databases in the new format. Full information on how to do this with different database engines is on the Perception support website.
3. You could choose to set up the databases with the same ODBC names as the standard installation, but usually ODBC DSN names will be chosen. Often all the databases in *Access* will be merged into one single database in *SQL Server* or *Oracle*. In any case, you must set up the DSN names appropriately for the databases.

4. If the DSN names change, then you must update **perception.ini** and **settings.asp** to show the new names. Moreover, if the database engine requires a name and password to be passed to it, you must set this in these files as well. See section 2.2.2 and 2.2.3 for how to do this.

Full information on setting up *Perception Server* with *SQL Server* and *Oracle* is on the Perception customer support website. Please refer to:

<http://www.questionmark.com/perception/help/databases/index.html> (North American users)

<http://www.qmark.com/perception/help/databases/index.html> (All other users)

## 2.4 Upgrading Perception Server from version 1

### 2.4.1 Compatibility of data

With some minor exceptions, all material created in version 1 of Perception will run in version 2 without modification.

All four of the main databases (question database, session database, security database and answer database) can be used in version 2. And all your existing questions and sessions should work fine in version 2.

Here are a few issues you may need to be aware of:

- You will have to re-create any custom reports made in the version 1 *Reporter* in the version 2 *Enterprise Reporter*. The version 2 reporting software is much more capable than the version 1 format but uses a different report definition format.
- The format and template system has changed considerably in version 2. Many version 1 templates will continue to work, but some may not work. If you have problems in using version 1 templates with version 2, please send them to Question Mark technical support.
- Because of the changes in the template and format system, it's possible that there will be slight differences in the way sessions are presented or appear. You should check that your sessions look satisfactory in the new environment.
- Although material is upward compatible, in that you can use the version 1 databases and files with version 2, the reverse is not always the case. Once you edit or change the databases in version 2, you should not bring them back to version 1.

- There was a problem in versions of *Perception Server*'s **session.dll** prior to version 1.2.3, where the answer database could not cope with outcomes that were longer than 50 characters. The count of who had answered which outcome could go awry. These problems are shown within *Enterprise Reporter*, especially in the survey report. If your answer database has this problem, you can either start again with a clean answer database for version 2, or else contact Question Mark technical support for help in cleaning it up.
- Your version 1 license file will not work with version 2 of *Perception Server*. You should be sent a new license file.
- You must wait for existing sessions to be concluded before switching to version 2. The format of the progress file used to record progress in a session has changed. It's not possible for someone to start a session using the version 1 software, and then finish it with the version 2 software.

## 2.4.2 Switching from version 1 to version 2

You will usually want to install version 2 of *Perception Server* in the same directory as version 1 was installed. You can if you prefer install it in a different directory, in which case it will work as a separate install. And the issues in section 2.3.3 will apply.

It should be easy to switch from version 1 to version 2. In addition to following the steps in section 2.1, here are some notes that are particular for version 1 users to guide you in the process. They assume you are installing Perception version 2 in the same directory as version 1.

1. You can keep your existing **qmresources** directory and any web sharing set up for it. But you will need to set up new web sharing for the **server** and **enterprise** directories. We recommend that the **server** directory is shared with name **qm**, which will replace the sharing of the version 1 directory **qm**.
2. The usual names of the answer and security databases are unchanged from version 1 to version 2. You should find that a default install will use your existing answer and security databases. As described above, you may want to replace your answer database with an empty one.
3. Whereas version 1 used **server.session** and **server.question** as the usual names of its server databases, version 2 uses **master.session** and **library.question**. To access your version 1 material, you must do one of the two things:

- Either from *Control Panel*, re-assign the ODBC DSN **QMPSession** to **server.session** and re-assign **QMPQuestion** to **server.question**.
  - Or using *Windows Explorer*, delete or rename the example shipping databases **master.session** and **library.question**. Then rename your version 1 server session database **server.session** as **master.session**, and rename your version 1 server question database **server.question** as **library.question**.
4. A new **perception.ini** is created for version 2, in directory **server**. If you made changes to your version 1 **perception.ini**, you might need to repeat such changes in the version 2 file. If you are using different databases, don't forget also to adjust the *Enterprise Reporter settings.asp* file.
  5. The version 2 format and template files are stored in directory **layout**.
    - If you made any changes to the version 1 file **html.format**, you may need to consider making similar changes to the version 2 format files, for example to **main.format**. However you are recommended if possible to put such changes in template files, not in the format files.
    - You should copy any version 1 template files that are in use to the **layout** directory.

Once you are satisfied with your version 2 installation, you can delete the version 1 directories **templates**, **qm**, **temps** and **reporting**, as these are no longer needed.

## 2.5 Resolving installation problems

### 2.5.1 Troubleshooting

Here are the three most useful things to look at when you have a problem with *Perception Server*:

1. Look at the **?about** listing. Calling any of the DLLs with the **?about** parameter will give some information about the installation of *Perception Server*. You can also get such information about *Enterprise Reporter* by running the **/enterprise/about.asp** program or clicking the **About** button on the *Enterprise Reporter* home page.

This listing will check whether the Perception databases can be connected to, and give other useful information.

2. Because *Perception Server* is a webserver application, it doesn't always display an error message on the screen. So if you have a problem, the first place to look is the **perception.err** file, usually stored in directory **\perception\logs**. If it doesn't already exist, this file is created when an error is found.

You can look at this file with any ASCII file editor, e.g. *Notepad*. And if there is an error, this will usually describe the error. Even if the error is also described on screen, the message in **perception.err** is sometimes more detailed.

3. Check the answers to common problems in the rest of this section, and search for information about your problem on the Perception support site.

If your problem is not dealt with in this manual or on the support site, then you should contact Question Mark for technical support. If you do contact us regarding a problem with the *Perception Server*, we will need the following information:

- Which version of Windows are you using, and which database format are your databases in?
- Which version of Perception are you using?
- If the problem is in the server DLLs, then please send a copy of the **?about** listing described above. Or if it is in *Enterprise Reporter*, then a copy of its **about** listing.
- A copy of the **perception.log** and **perception.err** files.
- A precise description of the problem. If there is a server error message, we need the exact error message text.
- If the problem relates to something not looking right on a browser, save the HTML file from your browser onto hard disk, and send us this file to look at via email.
- If the problem relates to a particular database, it may be helpful to send the database.

## 2.5.2 Issues to do with DLL installation

Although usually the installation software copies across all DLLs and performs any necessary registration, there are rarely issues that require manual intervention.

### **Incorrect versions of Microsoft DLLs**

*Perception Server* uses two Microsoft DLLs, **mfc42.dll** and **msvcrt.dll**. These need to be version 6 or higher. The installation software should install correct versions of these DLLs,

but sometimes it fails, and then *Perception Server* will not be able to run. You may in this instance see a message in the browser **The operating system cannot run %1**.

You can check the version of these DLLs on your system by finding them in Windows *Explorer*, in the `\windows\system` directory, `\winnt\system32` directory or similar. If you right-mouse click and choose **Properties** then **Version**, you will see the version number.

If your version is not version 6, then the probable reason for this is that the installation software was unable to overwrite them because they were in use.

The version 6 DLLs are on the Perception CD, in the **uncompressed** area of the CD. Here are some ways of installing them manually.

1. You may simply be able to use Windows *Explorer* to copy them from the CD to replace your old files.
2. If this fails, try renaming the existing DLLs to for example `mfc42.old` and `msvcrt.old`, and then copying the newer ones across.
3. If your PC has the option to shut down and restart in MS-DOS mode, then doing this will usually allow you to replace the DLLs. Or on Windows NT, if your installation is on a FAT drive, you can use a Windows 95, Windows 98 or DOS boot disk to take you to a DOS prompt, which will let you do the same.
4. If none of the above help you, then you will need to stop many or all of the Windows NT services so that on startup, the DLLs are not in use. And then you can manually replace or delete them.

#### **Problem registering Enterprise Reporter file qmper.dll**

*Enterprise Reporter* uses a COM object DLL called `qmper.dll`, which needs to be registered with the Windows system for *Enterprise Reporter* to be able to find it. The install program usually registers this DLL for you, but this process can sometimes fail. This can happen if the Microsoft DLLs mentioned above are not present, if you are upgrading from an earlier version of *Enterprise Reporter*, and perhaps in other circumstances.

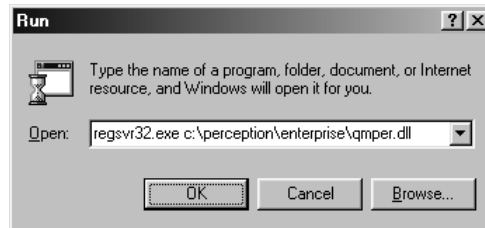
If it does fail to register, you may be advised of this during install, or you else you may see a message when you try to run *Enterprise Reporter*. Such a message might tell you that it couldn't find `qmper.dll` or else that the version found is wrong.

Here is how to manually register the file:

1. The file is copied on install into your main *Enterprise Reporter* directory, usually `\perception\enterprise\`.
2. To register it, choose **Run** from your **Start** menu.

3. Enter the following command line in the subsequent dialog box:  
**regsvr32.exe c:\perception\enterprise\qmpcr.dll** replacing the path if necessary on your system.

Windows should find the **regsvr32.exe** file automatically but if it tells you it can't find it, put the path, e.g. **c:\windows\system\regsvr32.exe** in the command.



4. Click **OK**.
5. The program will then register the file for you and tell you this has been done.



### 2.5.3 What to do if the software won't run at all

One mistake that people sometimes make is to try to run the server software as file programs rather than via their browser. The only way to run the *Perception Server* DLLs or *Enterprise Reporter* is from your web browser.

Also do not try to run **session.dll** directly (unless you are deliberately using PIP where you do call **session.dll**, see section 6.1). If you want to run a session without any participant interaction, do it by passing command line parameters to **open.dll** or **perception.dll**. The command line parameters you may see to **session.dll** are for internal use only, and will not work reliably if you use them yourself.

You need to make sure that your webserver is set up correctly and is running. If you connect to the root directory of your server, e.g. <http://www.xyzcompany.com/>, you should be

able to see the “home page” of your website. You should also check that your webserver is ISAPI compliant, i.e. usually it is either IIS or PWS.

If the server DLLs won't work at all, here are some things you can check.

1. Check that the **server** subdirectory has been set up for web sharing and with “execute” but not “read” rights. Check also that if your webserver differentiates between rights for executing scripts and programs, the server DLLs are programs not scripts.
2. Check you are accessing the correct files with the correct URLs.
3. Check the versions of **mfc42.dll** and **msvcrt.dll** as described in section 2.5.2 above.
4. A failure in ODBC can cause the software to fail to run. See more on things to check in ODBC in section 2.5.4 below.
5. If your Windows NT installation is not fully stable, *Perception Server* may have problems that are not apparent in desktop software. If your **perception.err** file shows the message **An unspecified error occurred**, this can mean that your installation is not properly configured. You may need to re-apply a service pack, re-install ODBC or make other changes to Windows NT to install it properly.

If *Enterprise Reporter* does not work at all, here are some things to check:

1. *Enterprise Reporter* will only work with *Internet Explorer* version 4 or higher, or *Netscape Navigator/Communicator* version 4 or higher. You also must not disable cookies, as these are used to control *Enterprise Reporter* security.
2. Check that your webserver has *Active Server Pages (ASP)* loaded and also the ADO component. These are sometimes not installed when IIS is installed, and you need both of them to run *Enterprise Reporter*.
3. Check that the **enterprise** directory is set up for web sharing, and has both “read” and “execute scripts” permissions.
4. Check that the file **settings.asp** has the correct information about the databases to be accessed.
5. If you can't login to *Enterprise Reporter*, check that you have been set up in *Security Manager* to access reporting. If it's a new installation, you should be able to login with user name **Manager** and password **Secret**. These are case sensitive.



## 2.5.4 Troubleshooting ODBC issues and problems connecting to databases

*Perception Server* needs ODBC. Check carefully that you have installed ODBC. Even if you believe you have installed it, it may be worth re-installing it, or installing a later version from the *Perception CD*, or from Microsoft's website.

If you have installed ODBC, but cannot get a connection to a database, then here are some possible reasons:

- Check that you have set up the ODBC **System DSN** to the correct databases; see section 2.3.2 for a description of this. Check that the name in **perception.ini** matches the **Data Source Name** in Windows *Control Panel*, and that the database pointed to is the correct one in the correct directory.
- Check that the server software has permission to access the directory in which the databases are stored. See section 2.1.5 for more information on this.
- Check that the databases are not set to be read-only; it's necessary for the software to be able to write to the answer, security and reports databases.
- If a problem is with a specific database, check that this database is not being locked by another application outside *Perception Server*.
- If you are using *SQL Server* or *Oracle*, make sure that the user name and password set in **perception.ini** are the correct one for the databases.

The **?about** listing or **perception.err** file may well give you further information about this. If you need more information, you can turn ODBC tracing on. This logs all accesses to ODBC and messages from the ODBC driver, and puts them into an ASCII text file. It's often possible to diagnose problems by looking at the file produced.

Here is how to do an ODBC trace:

1. Click on **32-bit ODBC Manager** or **ODBC Data Sources (32-bit)** from *Control Panel*. This runs the *ODBC Administrator* program.
2. Click on the **Tracing** page on the dialog.
3. Click the **Start Tracing Now** button.

4. Run *Perception Server* and try to recreate the problem that is causing you to want to trace.
5. Go back to *Control Panel* and click the **Stop Tracing Now** button.

Examine the log file, which is usually saved as file **SQL.LOG**. You can adjust the name and path of the log file from the **Tracing** tab if you prefer a different name.

The ODBC log file produced can be quite large. If you need to email it, you may want to zip it up before doing so. And remember to turn tracing off after using it – if you turn it on and forget to turn it off, you can end up with an extremely large log file, and a slowed down system.

## 3 Using and running Perception Server

### 3.1 Overview and management

#### 3.1.1 How the software works

These are the four steps in using *Perception Server*:

1. **Publish sessions onto the server.** Authors write their sessions and questions on Windows desktops, which they then publish to the server session and server question database. The templates and graphics and other resources needed are also put on the server.

The *Authoring manual* describes publishing in detail. Section 3.2 below describes how to organize it from the server perspective.

2. **Define who can take which session.** Either the authors or a System Manager can use *Security Manager* to define who can take which sessions. This allows you to set up participant names and groups, and allocate them to sessions. This information is stored in the security database, and accessed by **perception.dll**.

Chapter 4 covers *Security Manager*.

3. **Participants take the sessions.** There are two entry points for taking sessions, an open access entry point **open.dll** and a secure entry point **perception.dll**. Once logged in, participants then take the session and their results are saved in the answer database.

Section 3.3 describes how participants login to sessions, and section 3.4 describes how a session is delivered and answered.

4. **Report on the results.** You use *Enterprise Reporter* to analyze the results. This reads the answer database and prepares reports in a variety of formats.

Chapter 5 covers *Enterprise Reporter*.

## 3.1.2 Resetting

One issue you need to understand to manage *Perception Server* is resetting. This does not apply to *Enterprise Reporter*, but does apply to the four server DLLs – **open.dll**, **perception.dll**, **security.dll** and **session.dll**.

The first time each of these DLL programs is called, it initializes itself, which includes reading the **perception.ini** file, the format file and other system files. It then stores this information in memory. The second and subsequent times each of these DLLs is called, the DLL does not re-read the system files, but uses the information it has in memory. This re-use of information is an important reason why *Perception Server* can deliver sessions fast.

Thus, these DLL programs are started the first time anyone accesses them, and then stay running indefinitely. In fact once started, they stay running until the webserver software itself is stopped, or until the server PC is re-booted.

So, if you change any of the system files on disk, for example if you edit **perception.ini**, this change will not be registered by the DLLs. They have their copy in memory, and for efficiency reasons, do not check the disk each time they run. Therefore, if you change any system files, you need to tell the DLLs. You do this by resetting them. Resetting a DLL causes it to re-initialize itself, which means it re-reads all system files. Each DLL is reset independently.

You need to reset the DLLs under the following circumstances:

- If you make any changes to **perception.ini**, you need to reset all the DLLs.
- If you add a new license file, or change your license file, you need to reset all the DLLs.
- If you make any changes to **main.format**, or any other **.format** file that it references, e.g. **style.format**, you need to reset all the DLLs.
- If you change **login.format**, you need to reset **open.dll** and **perception.dll**.
- If you change **manager.format**, you need to reset **security.dll**.
- You also sometimes need to reset **session.dll** if you change a template file.
  - The first time you add a template file to the server, you do not need to reset **session.dll**.

- But when you change a template file or upload a new version of a template file, the template file will then be in memory, and you do need to reset **session.dll**.
  - If you set in **perception.ini** that **Template cache size = 0**, then you do not need to reset **session.dll** when you change templates. However, this will also mean that template files are read from disk each time they are needed, which will slow down the software slightly.
- You do not need to reset any DLLs when you change any of the Perception databases or add or change graphics or other resource files apart from templates.

There are three ways of resetting the DLLs.

1. You can do this from the **Admin** page of *Security Manager*. See section 4.4 for more details on this.
2. You can stop and restart the webserver. This will mean that each DLL will re-initialize itself when it is next called. Rebooting the PC has the same effect.
3. You can reset the DLLs by calling them with the **?reset** command. Thus, to reset **session.dll**, you would type the following into your browser:  
**http://www.xyzcompany.com/qm/session.dll?reset**. To reset all four DLLs, you would repeat this for each DLL. If you want to disable command line reset, and only permit reset from *Security Manager*, you can do so by a setting in **perception.ini**.

Resetting **perception.dll**, **open.dll** and **security.dll** only have the effect of re-initializing the programs, but resetting **session.dll** has one extra effect. The system makes a progress file for each ongoing session. These are working files, which hold the current session status. These take up disk space, but are not deleted until **session.dll** is reset. At that time, each ongoing session which has not been accessed for longer than the **Clean Up** time defined in **perception.ini** has its progress file deleted, so that it expires.

The default **Clean Up** time is 12 hours, so if you want to have sessions that last longer than 12 hours, you must change this. Setting a **Clean Up** time of zero means that progress files are never deleted. If you do not want long lasting sessions to be expired, you should set the **Clean Up** value appropriately.

There are some related issues you need to consider when managing an installation of *Perception Server*.

- Progress files will not be deleted unless **session.dll** is reset, either by command or by the webservice stopping and restarting. If the webservice runs for long periods, you should reset **session.dll** at regular intervals, perhaps once a week or once a month, depending on your usage, to stop there being too many dead files.
- As with any computer system, it's important to back up the data being used by *Perception Server*. You should at a minimum back up the Perception databases, and all the resources in directory **qmresources** and templates in directory **layout**.
- All errors are written to file **perception.err**. You should look at this file from time to time, and review any errors.
- A larger amount of information is written to file **perception.log**. You might want to delete or archive this file from time to time, as if it gets very long, it will slightly slow the server software.

## 3.2 Publishing sessions to the server

### 3.2.1 Choosing your approach

The Perception authoring software allows authors to make as many session and question databases as they wish. However, *Perception Server* works with only a single session and question database. When released, sessions usually need to be published from the authoring databases into the server ones.

Publishing is done from *Session Manager*, and full documentation on publishing is in the *Authoring manual*. A user chooses a session, and asks that it be published, i.e. copied, from his/her local database to the server database. The session is copied from one session database to the other. The topics and questions referred to by the session are copied from one question database to another. And the graphics files, multimedia files and template files referenced in the session and questions can also be copied.

Publishing can work in two ways:

- If you publish to **Access...**, then *Session Manager* needs a direct file connection to the databases being published to.
- If you publish via **ODBC...**, then *Session Manager* needs an ODBC connection to the databases being published to. With some database engines (e.g. *Oracle*), it is possible to have a remote ODBC connection over TCP/IP, so that you can publish sessions over the Internet. Publishing to ODBC needs a special setting in your Perception authoring license file – this is not enabled on the standard Perception purchase.

There are a number of different ways of structuring your server databases and access to them. Here are some of the more common approaches:

- **Direct editing of server databases**  
Authors do not make their own databases, but directly add and edit sessions and questions in the server databases (**master.session** and **library.question**).

This is the simplest solution, as the authoring database and the server databases are the same. But it means that the author is working in the same database participants are accessing, which is not best practice, and could lead to file sharing problems. This is appropriate only for small-scale installations, for example with one author, who is responsible for the databases.

- **Direct publishing with Access to server databases**  
Authors make their own databases, but use *Session Manager* to publish to *Access* the sessions and questions from them to the server databases.

This allows authors to work on their own material, and even take databases to other sites or away from the office on portable PCs. However, for publishing, it requires a direct network connection from the authoring machine to the server. Usually the Perception directory on the server is mapped as a network drive on the author's PC.

Because of this, your authors have full power to access the databases, and so need to be trusted. In many organizations, this is not a problem, but if you have authors in different departments who want to protect material from each other, it could be a problem. It also is not appropriate if the server PC is remote from the authors.

Providing your authors are trusted and the server PC is accessible, this is the best approach to take.

- **Direct publishing using ODBC**

If lack of security or the remoteness of the server make the above option impractical, you can let authors publish into the server databases via ODBC.

This requires you to set up ODBC access from the authoring PCs. Moreover, it also requires a specially licensed *Session Manager*. However, it does allow you to let people publish to the databases without giving them file access to them, which is a security improvement.

If your databases are upsized to *SQL Server* or *Oracle* format, you have to publish to them using ODBC.

- **Trial and release databases**

Rather than getting authors to publish directly to the server databases, you can have two versions of the server software installed onto the PC. One version is for trial and testing purposes, and one version is for release. Authors publish directly to the databases for the trial version, and can test out their sessions with them.

When a session is ready to be released to real participants, a System Manager publishes it from the trial server installation onto the real one. This means that authors can do what they want, but only a trusted person can put sessions up on the real server.

- **Single System Manager responsible for server databases**

Alternatively, you can just make a single System Manager entirely responsible for the server databases. Authors send their databases to the System Manager, who publishes them to the server databases.

This makes things very easy for authors and removes any security concerns. But it does give work to a System Manager.

- **FTP upload to a remote server**

If your webserver PC is remote from your authors, it's possible to manage its databases with FTP upload. If all authors are in the same location, you can keep a master copy of the session and question databases at one location. And then periodically replace the server databases by FTP upload. You should usually take the server offline (see section 4.4) while you are doing the FTP.



## 3.2.2 Publishing technical issues

Here are some issues you need to be aware of if many people are publishing to one server and in any case when you are overwriting existing material.

### **Two people using the same resource file name**

The server has one directory for graphics files (`\perception\qmresources`), one directory for Java and hotspot graphics files (`\perception\qmresources\java`) and one directory for multimedia files (`\perception\qmresources\multimedia`).

Because all files are stored on a single directory, if two authors use the same file name for different resource files, then one author is likely to overwrite the other's file.

In a multi-user environment, it's recommended that you prefix all resource files with a code for each author or with the author name, to avoid conflicts. Or else have one person manage publishing and file uploads.

In a large organization, you may wish to consider a separate installation of the server for each department, which will allow each department its own resources directories and databases.

### **Two people using the same template file name**

The server has one directory for template files, and so if two authors choose the same template file name, one person's file could overwrite another's. In a multi-user environment, you need to set up some system for preventing this. Either all template file names can have a prefix for their author, or one person should control the server to check names aren't duplicated.

Although you don't need to reset the server when you update the databases, template files are cached, so if you change or overwrite an existing template file, you must reset `session.dll`, as described in section 3.1.2.

### **Two people using the same session name**

Although duplicate resource and template file names do give a problem, it doesn't matter if two people choose the same session names. Sessions are uniquely identified by their session ID. Session IDs are long randomly generated sequences of digits, which are extremely unlikely to be duplicated.

Suppose one person makes a session called **My session**, and publishes it to the server session database. Then someone else publishes a session also called **My session**. It will not overwrite

the original session. Both sessions will be present in the session database, and though this can be mildly confusing, it is not dangerous.

#### **Two people using the same topic name**

The publishing software prevents two people from using the same topic name. If one person has already published a topic, then someone else will not be permitted to publish the same name. This is kept track of by using a randomly generated number in each question database as the topic internal ID. Unless these IDs match, you aren't allowed to publish into a topic.

In order to avoid conflicts at the publishing stage, you're recommended to get different users to agree on using different top level topic names, but the publishing software will prevent any clashes if users do not agree.

#### **Publishing does not remove questions**

If you publish a session, it publishes all questions in the entire topic tree, but it does not remove existing questions. You will need to manually remove any existing questions you do not want.

#### **Uploading while tests are happening**

You need to be careful about publishing while people are taking sessions. There should be no difficulty if you are adding new material, but if you change a session whilst it is in progress, then this could clearly impact on the session in progress.

#### **Security when publishing**

If a session is confidential, then you should set **Make session available only if scheduled via Security Manager** on the **Security** tab in the *Session Manager Control Block* dialog. This ensures that a session is not to be used in **open.dll**. Or else, you could disable **open.dll** by making a setting in **perception.ini**. If you don't do this, then anyone can see the session by running **open.dll**.

Also if some users of *Security Manager* are not trusted, you should make sure to set a password for the session on the **Security** tab in the *Session Manager* control information page. This will mean that no one can schedule the session without quoting this password.

## 3.3 How participants access sessions

### 3.3.1 Using perception.dll

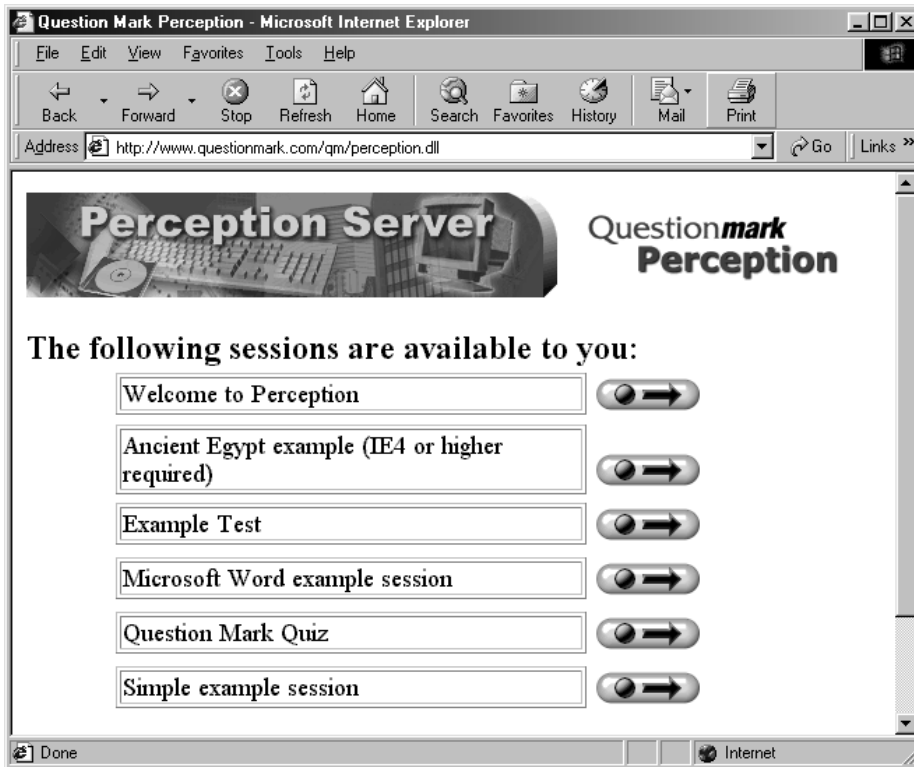
There are two standard entry points to *Perception Server*. The secure entry point is **perception.dll** and the open access point is **open.dll**. See also *Perception Integration Protocol*, described in section 6.1, for another way of accessing sessions.

Each of **perception.dll** and **open.dll** asks the participant for his/her name and then presents a list of sessions to take. If you are using **perception.dll**, the participant is asked for a name and password, as in the screenshot below. A valid name and password must be entered, or the program will go no further. Participant names and passwords are not case sensitive.



Once a name has been entered, and the participant clicks the **Start** button, the software then looks for available sessions. A list of all sessions is made that have been scheduled to the participant in *Security Manager*.

Sessions are presented in alphabetical order of schedule name. The schedule name is a name set in *Security Manager*, that can be the same or different to the session name.



The screenshot above shows an example screen, with six sessions available. To start a session, the user clicks on the graphic with ball and arrow to the right of the session name.

You can pass information on the command line to bypass the login screens. This allows you to set up your own HTML pages, which call *Perception Server*, or else to call *Perception Server* from other programs. The possible syntax is as follows:

**perception.dll?NAME=name&PASSWORD=password**

This passes the participant name and password to the software, so the participant does not have to type it in. The first screen the participant will see will be the list of sessions scheduled.

**perception.dll?NAME=name**

If the participant has a blank password, you can just pass the name without any password. For example, if you define a name **demo** and schedule some freely available sessions to it, then calling **perception.dll?NAME=demo** will allow anyone access to these sessions.

**perception.dll?SESSION=sessionid**

This runs the particular session. The participant is asked for a name and password, and provided the session is scheduled, it is gone to, without showing the participant a list of sessions. The parameter passed is the session numeric ID, not its text name. You can see this from *Session Manager* by right-mouse clicking on a session and choosing **Properties**. (Or else if you choose to view the source of the login screen HTML, you will see it by the session.)

For example, you can define a link in an HTML document to a particular session. Participants can click on this link. Their name would be validated if they are scheduled to the session. They would go straight to it without a choice of other sessions.

**perception.dll?NAME=name&PASSWORD=password&SESSION=sessionid**

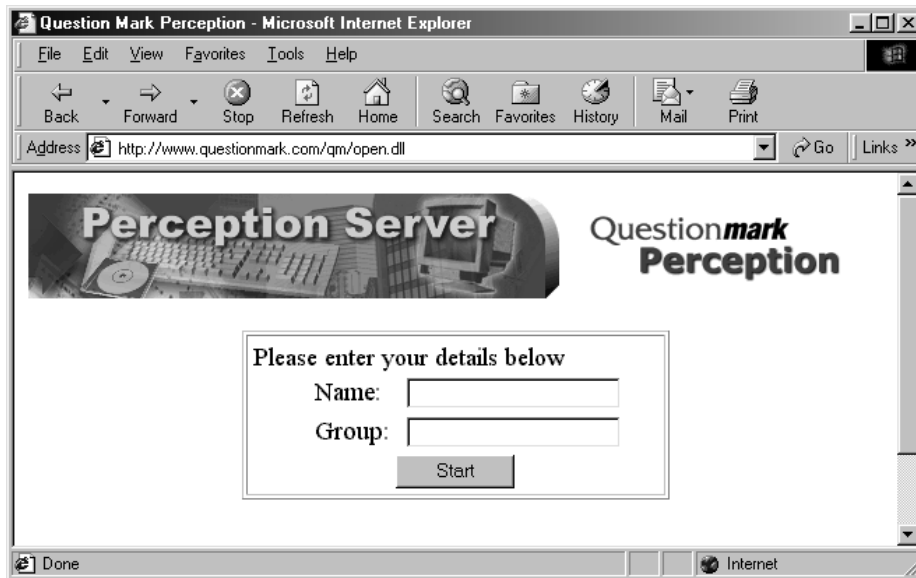
**perception.dll?NAME=name&SESSION=sessionid**

This passes all the information on the command line. The information is checked, and providing it is authorized, the session is run immediately. Neither of the login screens are shown. This might be useful if you are calling the software from another program, and simply wish to go to the session with a defined name.

### 3.3.2 Using open.dll

**Open.dll** is an open access program that does not check the participant's name against the security database, and which offers all or most sessions, not just the ones scheduled. It can be useful if you are offering tutorials or quizzes, or are not fussed about security. You can disable it if you do not want open access to sessions.

If you are using **open.dll**, then the participant is asked for a name and group. Any values can be entered in, and these will be accepted by the system and passed onto the answer database. You can change **login.format** to make the participant choose their group from a drop-down list, rather than be able to enter any text – see section 3.3.3 for how to do this.



The program then gives a list of sessions in alphabetical order of session name. All sessions in the server session database are shown, except those that have **Make session available only if scheduled via Security Manager** set in *Session Manager*.

If you have secure sessions, which you do not want seen in **open.dll**, then you *must* make this setting for each such session in *Session Manager* (on the **Security** tab of the **Control Block** dialog). Or else you can turn off **open.dll** completely by setting **Enable open access to sessions=0** in **perception.ini**.

You can use command line parameters with **open.dll** in the same way as with **perception.dll**, except that no password is needed as all names are accepted. You can also use the parameter **GROUP=groupname**, which sets the group the participant is in. In **perception.dll**, the group information comes from the security database. Which group a participant is in is stored in the answer database, and is useful in reporting. Thus:

**open.dll?NAME=name&GROUP=group**

This sets a name and group, and shows a list of sessions.

**open.dll?SESSION=sessionid**

This asks for name and group, but then runs a session without asking the participant which one is wanted.

**open.dll?NAME=name&GROUP=group&SESSION=sessionid**

This runs the session without showing either of the login screens.

You have to specify at least one of **NAME** and **GROUP** when using **open.dll**, but if you only specify one, the other is set blank.

### 3.3.3 Changing the participant login screens

The layout of the login screens is defined in a file called **login.format**, which is stored in the **layout** directory.

This is an ASCII text file, and you can edit it if you want to change the layouts. Whatever changes you make will apply to all use of this installation of *Perception Server*. If you want different login screens for different sessions, you need to make two separate installations.

To edit the file:

1. Open it in *Notepad* or another text editor. Do not use *FrontPage* as this can cause problems.
2. Make the changes you want and save it.
3. Reset **open.dll** and **perception.dll** to make the changes take effect.

The entire layout of the login screens is defined in **login.format** (in conjunction with **main.format**). Here are some things you can do by changing this file:

- All the text used in the login screens is defined in this file, and can be safely changed. For example, if you want the box for group to be called department, then change:

**ASK\_GROUP\_TEXT=Group:**

to

**ASK\_GROUP\_TEXT=Department:**

- You can choose to open sessions in a new window. If you do this, then the browser will usually open the session in a new window with no toolbars, and with as much screen space as possible available for your questions. This can be a good security measure, to

make for example printing or copying your questions harder. It also gives more space on the screen for your session content. To make this happen, set:

**OPEN\_NEW\_WINDOW=1**

- Participants usually start a session by clicking on a graphical icon to the right of the session name. If you want the list of sessions to be displayed as a series of buttons, where the participant clicks on the button to start a session, then set:

**USE\_GRAPHICAL\_LIST=0**

This can look good in *Internet Explorer*, but less good in *Netscape Navigator* browsers.

- **open.dll** usually lets the participant type in any group name. If you want to change this so that they can choose from a list of groups, set:

**OFFER\_GROUP\_LIST=1**

You also then need to set the groups that participants can choose from in the **GROUP\_LIST** variable in **login.format**. Instructions for doing this are given in the file.

- If you want users of **perception.dll** to have an option to see sessions they can restart, then set:

**OFFER\_RESTART=1**

See section 3.4.8 for more about what this means.

- You can change the graphic shown at the top of each screen by changing the line:

**HEADER=<IMG SRC="%SERVER.GRAPHICS%system/server.gif">**

This displays the graphic **server.gif** in directory **qmresources\system** at the top of each screen. You could put alternative HTML here as the header for the page, or reference a different graphic. Or else, you could keep **login.format** as it is, but simply replace **server.gif** with your own graphic file.



- You can also change the graphic used to click on to start sessions by changing the line:

```
START_SESSION_GRAPHIC=%SERVER.GRAPHICS%system/qmp_go.gif
```

See the Perception support site for more on the syntax of format files. If you change **login.format**, and you want to return to the original version, you can find a copy of this on the Perception CD in the **uncompressed** directory.

## 3.4 Taking sessions

### 3.4.1 Sequence of events during a session

After it has validated a participant is permitted to take the session, **perception.dll** or **open.dll** make a call to **session.dll** to actually run the session. What happens during a session is the same whichever of **perception.dll** or **open.dll** was used to do the validating.

When a participant starts a session, a progress (**.pro**) file is made in the temporary working area on the server, and a unique access ID is allocated to the participant (hidden within the browser HTML). The software keeps track of the participant using this access ID, and records the ongoing situation in the progress file.

The sequence of events during a session is as follows:

1. **session.dll** is called with the details to start a session. This is checked, and a progress file is made, and the answer database is updated to show that the participant has started the session.
2. The first block of questions is constructed and sent to the participant.
3. When the participant responds to the questions, the answer database is updated with any answers.
4. If feedback after each block is called for, this is given, and the server then waits for the participant to click **Continue** after seeing this.

5. The software then evaluates any jumps after the block, and determines the next block of questions, or whether the session is over.
6. If there is another block, this is constructed and sent to the participant. The process then repeats itself at step 3.
7. When the session is over, the answer database is updated to show that the session has finished, and a final page is sent to the participant, containing any end of session feedback and the information for the appropriate score band.
8. The end page can result in a jump either to another session, to another URL or “home” to **perception.dll** or **open.dll** to show a list of sessions.

Note that the answer database is only updated to show that a session is completed at step 7. If a participant starts a session but does not finish it, the answer database will show an uncompleted attempt. You may want to encourage your participants (and define your layouts appropriately) so that participants tend to go to the end of sessions.

Because of the nature of the browser/server connection, if someone “double clicks” on a button to start a session, (i.e. they click it once and then again), then this can lead to the server thinking that two separate sessions are being started. Step 1 is essentially repeated. The participant only sees one, and only answers questions in one. But you may see a started but not finished session in the answer database.

### 3.4.2 Keeping track of progress through a session

The server keeps track of the current progress of a participant through a session. Whenever a page is sent to the participant, it is always the current page, even if the participant tries to evade this. For example, if a participant:

- Answers a block of questions.
- Sees some feedback to the questions.
- Uses the **Back** button on the browser to go back and try to answer the questions a different way.

Then, rather than accepting the new answers, the software will repeat the feedback to the original ones.

And the **Refresh** button on the browser always gives the current page as far as the server reckons it. So if a participant:

- Answers the first block of questions.
- Is sent the second block of questions.
- Clicks **Refresh** – he/she will see a copy of the second block of questions.

And if a participant:

- Answers a block of questions.
- Is sent the second block of questions.
- Uses the browser button to go back (showing the first block).
- Clicks **Refresh** – he/she also will see a copy of the second block of questions.

The only exception to this is at the beginning of a session, where if someone clicks the **Refresh** button, they will start a new copy of the session. A second entry in the answer database is made, and the session is started again. If you want to minimize the possible impacts of this, then it may be wise to have an introductory block before the real questions, so people cannot get further than this and still be able to have **Refresh** start a new session.

Note that some browsers may not implement the re-loading process properly, making it dangerous to use the **Refresh** button with them. This is a known problem with some versions of *Internet Explorer 3*. With all browsers, clicking the **Refresh** button will refresh the page, losing all answers made so far, so you should caution people to avoid this button if possible. See also section 3.4.7.

### 3.4.3 Templates and how server documents are formatted

When *Perception Server* sends a document to the participant, it converts the session and question data defined by the author into HTML, so that the browser can read it. Because questions can be randomized, this conversion is done at runtime, separately for each document.

The rules for converting into HTML and the styles and layouts used are defined within ASCII files called format and template files. Format files are system files, which you should not usually change, unless you have good reason. The main one is called **main.format**, and this references other **.format** files. The format files define the basic structure of the HTML conversion process, and provide the default settings, which template files can modify. If you do change any format files, make a careful note of your changes, so that you can re-make them if newer format files are distributed as part of a later Perception release.

Template files modify the format files to provide particular layouts for particular sessions. Perception ships with some templates that you may want to use. Alternatively, authors can make their own templates.

All template and format files are usually stored in directory `\perception\layout`.

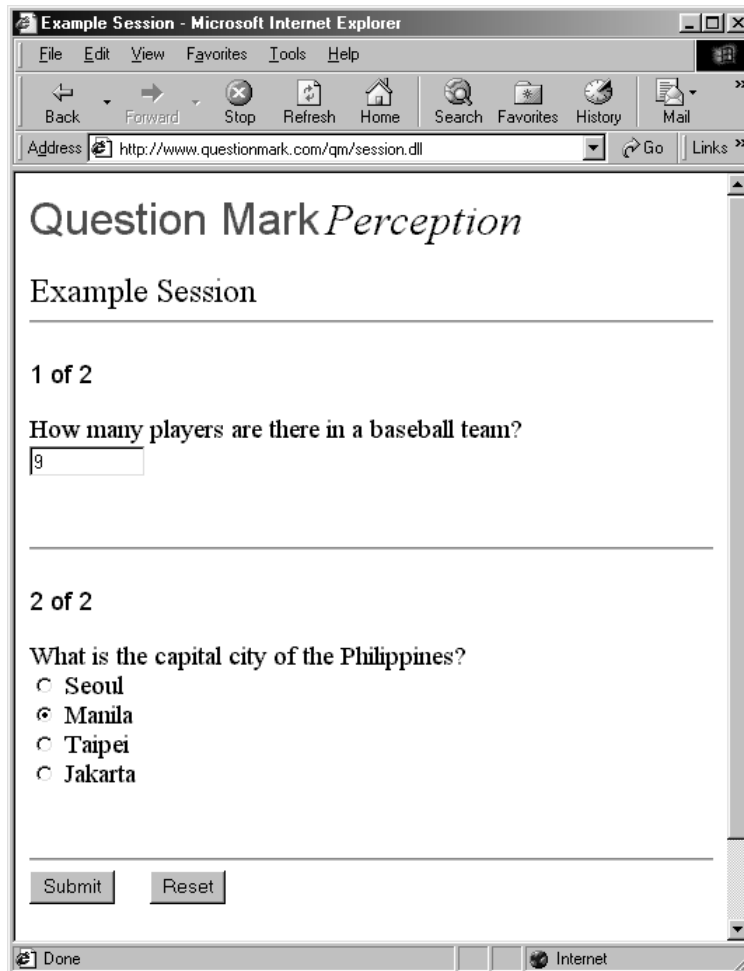
At the time of writing this manual, you edit templates using an ASCII editor, e.g. *Notepad*, but it's possible that other tools will be made available. Editing templates is not documented in this manual, but is documented online in the support site:

<http://www.questionmark.com/perception/help/templates/index.html> (North American users)

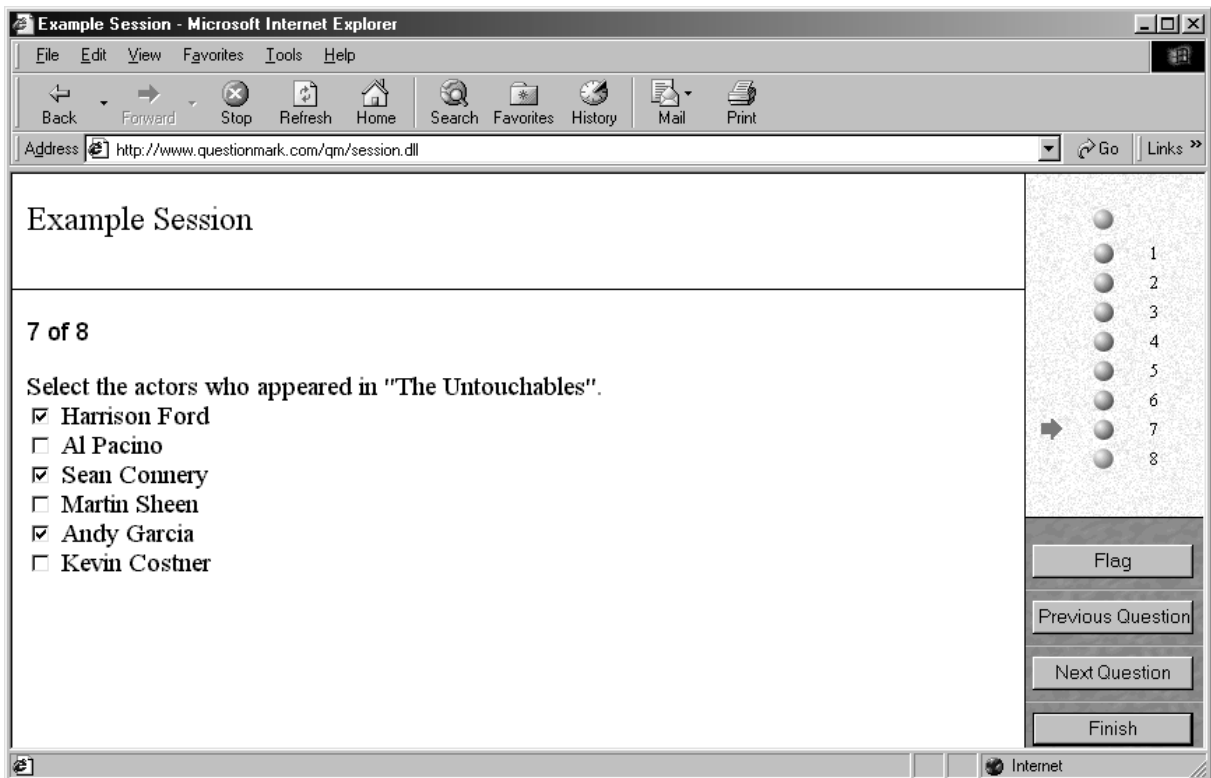
<http://www.qmark.com/perception/help/templates/index.html> (All other users)

### 3.4.4 Question by question delivery

The standard way of presenting questions has each in a single document in the browser, one question underneath the other. The participant can scroll through the document to view questions, and submits his/her answers by clicking the **Submit** button at the foot of the document. The screenshot below shows an example of this:



An alternative method, question by question delivery, is available. Here, each question is displayed on a single screen, and the participant clicks the **Next Question** button to see the next question. When the participant is ready, he/she clicks **Finish** and all the questions are submitted. The screenshot below shows an example of this:



Question by question delivery is turned on for a session or block by a setting in the template file used for the session. Any template that uses question by question delivery will have this line in it:

**USE\_QXQ=1**

See the templates documentation for other settings that you can make to control question by question delivery.

Question by question delivery only works with *Internet Explorer* version 4 or higher. It does not work with lower model browsers, and at the time of writing, it does not work with

Netscape browsers at all. On most browsers, if question by question delivery is not possible, the questions should be displayed normally, but on some browsers it's possible that there will be display or other errors.

You should only use question by question delivery if you are reasonably sure that all your participants will have *Internet Explorer* version 4 or higher. In practice, it may be wiser to use version 4.01 or version 5 or higher. Although question by question delivery does work with *Internet Explorer* version 4.0, version 4.01 fixes a number of stability issues in version 4.0.

### 3.4.5 Time limits

If a session has a time limit, then this is counted and enforced by the server software. All answers received up to the end of the time limit are stored in the answer database, but all answers received after the time limit has expired are discarded (lost).

How the time limit is displayed to the participant is controlled in the session template. Usually if a time limit is set, then the time remaining is shown in the browser window. And the session automatically is submitted when no time is left. If the session has multiple blocks, each block will briefly appear and then be submitted once the time has run out. Additionally a warning message can be given five minutes (or another amount of time) before the time has expired. You can change the way this works by adjusting the template file.

There will always be a delay in sending questions from the server PC to the participant PC, and in returning the answers. Therefore, the time remaining shown on the participant PC will differ slightly from the reckoning the server has. Moreover, if there are connection delays or the server is very busy, this time difference could be significant.

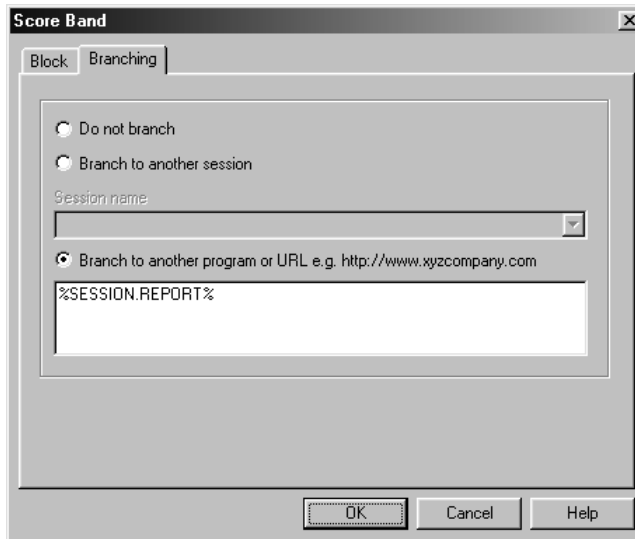
To deal with this issue, a timing leeway is defined in **perception.ini**. This is a single setting for all sessions on the server, giving an extra amount of time before the time limit is enforced. So for example, if the value is set:

**Timing leeway=120**

then 120 seconds are allowed to elapse before a time limit is final. So for example, a 30 minute session will actually get a time limit of 32 minutes. You can adjust this timing leeway to a different value if you want. However, if you set it too low, this could lead to some people running out of time without them realizing it.

### 3.4.6 Branching at end of session

At the end of each session, the participant is shown a final screen, which can vary by score band. Different actions at the end of a session are possible. Authors choose which is used by settings in the score bands of sessions in *Session Manager*. There are three options authors can select for each score band:



- If **Branch to another session** is selected, then a **Next session...** button is displayed on the final screen. You can change the name of this button by changing the **LINK\_SESSION\_TEXT** variable in the session template file. If the participant clicks the button, they will run the next session. This is done by a direct call to **session.dll**, without going back to **open.dll** or **perception.dll**.

There is no security control on branching to another session. It's assumed that if you have permission to take one session, then you have permission to take all sessions branched to from the session.

- If **Branch to another program or URL** is selected, then a link is displayed on the final screen, which the participant can follow to go to the link. The wording for this link is set in the template file in the **LINK\_URL\_TEXT** variable.

Branching will usually cause the participant to exit the Perception software. However, one special link that authors can define is to put the wording **%SESSION.REPORT%** as their branch URL, as in the screenshot above. This will cause the final screen to offer a link to *Enterprise Reporter* to show a report on the participant's results. See section 5.8.2 for information on which report is shown.



- If **Do not branch** is selected, then a **Home** button is usually displayed on the final screen. Clicking the **Home** button takes the participant back to the list of sessions to choose another session. You can control the way the **Home** button works in the following ways:
  - Whether this goes back to **perception.dll** or **open.dll** is defined by the **Home DLL** setting in **perception.ini**, which is usually set to **perception.dll**.
  - You can change the name of the button by setting the **LINK\_HOME\_TEXT** variable in a template file, for example:
 

```
LINK_HOME_TEXT=Return to base
```
  - You can remove the **Home** button completely by not having any text, i.e. setting the following in a template file:
 

```
LINK_HOME=
```
  - You can set that the **Home** button goes to a fixed URL by making two settings as follows in a template file, (obviously using your own URL instead of the example one):
 

```
LINK_METHOD=GET
HOME=http://www.questionmark.com/
```

See the templates section on the Perception support site for fuller information on how to control the **Home** button.

### 3.4.7 Browser issues

A Perception session is run by two programs, *Perception Server* running on the server PC and the participant's browser running on the participant's computer.

- *Perception Server* is only active when it sends a block of questions to the participant and when it receives back answers. It has no information about what is happening at the participant's computer except at these times.
- The browser receives the data from *Perception Server*, and handles the participant user interface until the answers are submitted.

If you lose your connection to the server, you can easily recover it. See the section on restarting below. And if connection time is at a premium, you can deliberately connect to the server, get a block of questions, go offline, and then re-connect to answer the questions. The server won't even know this is happening.

Participants can also save a document from a session in progress on their hard disk, turn off their PC, and then turn it back on, load the document, answer any questions and submit it. This book-marking facility allows people to take a break during a session, or even answer one block one day and the other one the next day. Obviously, this assumes that a session doesn't have a time limit to prevent this. See also section 3.1.2 regarding resetting the server and how to avoid long lasting sessions being cleaned up; you need to set an appropriately long **Clean Up** time in your **perception.ini** file.

One point of vulnerability in the robustness of the testing process is while participants are entering their answers, before they submit. For example if a block has 20 questions, in it and the participant answers 10 of them, but is not yet ready to submit, then there are things that can go wrong, for example:

- Clicking the **Refresh** or **Reload** button will refresh the page, and lose the answers made so far.
- Exiting from the browser will also lose the answers so far, though this is harder to do by accident.
- On some early model browsers, pressing the **Enter** key can submit the answers, even if this is not intended.

Clicking **Refresh** or quitting the browser will also affect question by question delivery. The answers to questions are only submitted to the server when you click **Finish**. So a careless participant could lose their answers by clicking the wrong buttons.

A reasonable solution to this issue is to set up the browser in “kiosk” mode where refreshing is harder. You can use the facility in **login.format** that allows you to present the session in a window without a **Refresh** button, see section 3.3.3. It may also be worth thinking about dividing a large session into a number of smaller blocks to avoid these issues.

### 3.4.8 Restarting

Because sessions are controlled by the server software, it is possible to restart a session if it is lost for any reason. For example, if the participant PC or browser fails, you can simply restart the session from the same or another PC. This will restart the session at the point it was last submitted to the server software.

Restarting works by looking for progress files. Once a progress file disappears, for example because it expires or is cleaned up, restarting will no longer work. If someone has taken two separate instances of a session, then restarting will find one of them, but it's not defined which one is found. Once the first session has been completed, the second session will then be available for restart.

There are two ways that restarting can happen. Manual restart means you only restart if you ask to. Automatic restart means you always restart a session if possible.

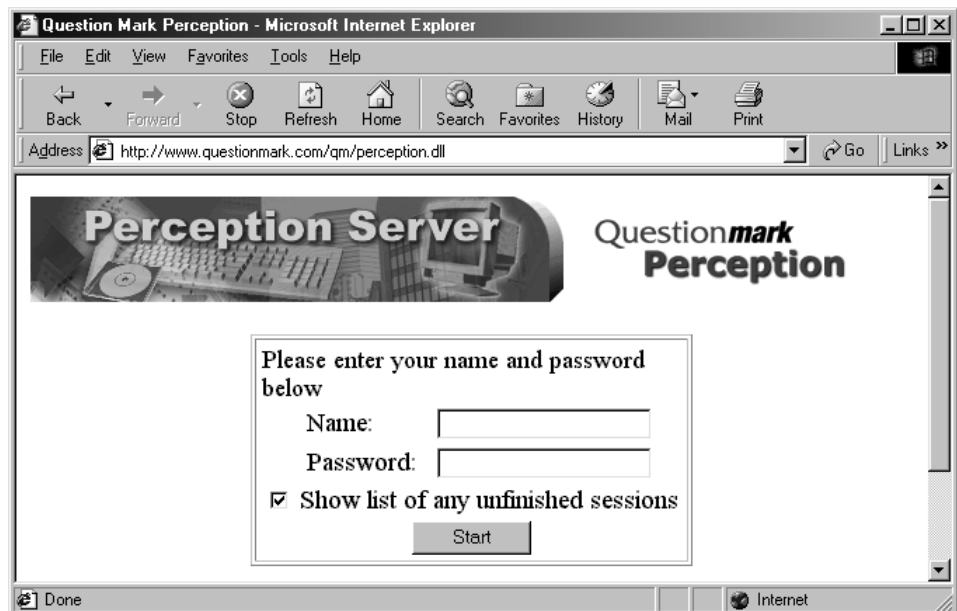
### Manual restart

Manual restarting allows someone to restart a session by specifically asking to. It only works with **perception.dll** not with **open.dll**. When someone tries to manually restart, the server software will scan all sessions in progress for the person logged in, and present a list of all sessions in progress so the participant can choose one to restart.

Manual restart is enabled by default. If you don't want to permit manual restarting, you should put the command **Disable restart function=1** in your **perception.ini**. This will stop people from restarting in this way.

There are two ways in which a participant can try to manually restart:

- By using the **?RESTART** parameter, e.g. **perception.dll?RESTART**. If you call **perception.dll** like this, it asks for your name and password, and gives you a list only of sessions that you have already started (or that someone with your name has started). Clicking on any of the entries will then resume the session at the position at which it was last accessed. So if you start a session, abandon it, and want to come back, then simply call with the **RESTART** parameter. You can optionally pass a name and session on the command line as well, for example **perception.dll?RESTART&NAME=John Doe&PASSWORD=Secret**.
- Alternatively if you set **OFFER\_RESTART=1** in **login.format**, then each time someone runs **perception.dll** they have the option to look at restarting sessions or new sessions. Their login screen looks like the screenshot below:



If they select **Show list of any unfinished sessions**, they will see only any unfinished sessions to restart. If they do not select this, they will see new sessions as normal. It's not possible to see both on the same page.

#### **Automatic restart**

Automatic restart means that whenever someone starts a session, the system checks to see if a person of the same name has a session in progress, and if so, it takes you to the last place in the session accessed. This means that people always go back to the same place they were at before, and once they've started a session, they cannot start a second copy until it is finished.

Automatic restarting is not enabled by default. It only works if you set it on in your **perception.ini** file. You should put the command **Enable automatic restart function=1** in your **perception.ini** file to make it work.

These methods of restart are designed for situations where each person logs in with a unique name (i.e. usually through **perception.dll**). If different people log in with the same name, because either you permit this, or because you use **open.dll** which does not control which name people use, then restarting will mean that one person could go to another person's session. You should only enable automatic restart if this is not possible on your system, and you should only use manual restart where this is not likely.

The only time automatic restart does not give access to a session that is available is if there are no attempts left in *Security Manager* for it. In such cases, the only way to restart the session is using manual restart.

# 4 Security Manager

## 4.1 Overview

### 4.1.1 What you can do with Security Manager

*Security Manager* is the part of Perception that lets you control who can take which session. It also controls who is permitted to use *Enterprise Reporter*. The software is used from a browser, over the web or intranet. You define which people are allowed onto the system and which sessions they can take. You can permit participants only to be able to take a session at a certain time, or that they can only do it a certain number of times before it expires.

In a typical use of *Security Manager*, you would:

1. Enter every participant's name into the system, with a unique password. For example, if you have 1000 students or employees, you make 1000 participants. You can either type in the names, or else import them from an existing file.
2. Allocate the participants into groups, which could correspond to student classes or employee categories.
3. Set up new users to access *Security Manager*. You can define different users, each of which can have different capabilities. These same user names can also be used to login to *Enterprise Reporter*.
4. Schedule the different groups, the participants within a group or simply individual participants to take different sessions. Scheduling can limit access to a session to a particular time-period, and you can also limit the number of attempts a participant is allowed.

You can also use *Security Manager*, without having to set a different participant name for each person accessing the system. A couple of examples are:

- You might want to define a demo or tryout participant name with a blank password, and schedule some sessions for open access to anyone who enters this name. Or by setting up a link from an HTML page, for example,

**<http://www.xyzcompany.com/qm/perception.dll?NAME=demo>**

you can ensure that people see the list of open sessions without having to enter a name.

- If you don't know the names of all your participants, or don't want to enter all their names into the system, you can use a single generic name. For example, you could define a name of **manager**, with a single password for all users, and schedule some sessions to this name. Whenever someone logs in as this name, they are offered these sessions, and either you do not record the individual name of each participant, or else you ask this name (and do not check it) in the introduction block of the session.

It's important to remember that people can sidestep the security process by using **open.dll** rather than **perception.dll** to access any sessions. **open.dll** allows someone entry without having to quote a password. However, a session can be setup so that it can only be accessed if it is scheduled to a participant. You should select **Make session available only if scheduled via Security Manager** from the **Security** tab of the **Control Block** dialog in *Session Manager*.

There is a section in the **perception.ini** file which further controls security of sessions. **open.dll** can be disabled by editing the following setting:

**Enable open access to sessions=1**

By setting this variable to **0**, **open.dll** will not work. Alternatively, you can edit the following:

**Disable open list of sessions=0**

If this is set to **1** then **open.dll** will only run if a session ID is passed as a parameter. If it is, then that session will run, but if it is not then access will be denied. This prevents browsing of sessions, but allows you to run a session if you know its numeric ID.

## 4.1.2 The security database

The settings you make in *Security Manager* are stored in a security database (usually **qmpsecurity.mdb**).

*Security Manager* (**security.dll**) reads and writes from this database. Then the *Perception Server* software (**perception.dll**) interrogates this database when it runs, and only permits people registered in the security database to take sessions, and only to take the sessions that you have assigned them.

The database contains information on users, participants, groups and schedules:

### Users

The database distributed with Perception contains only one user called **Manager**. This user has full access to *Security Manager*. Anyone who logs in as this user can define extra users who can each have different levels of access.

A user can be allocated to just one group, or some groups or even all groups. Furthermore, they can have various permissions over these groups. For example, they might be able to create and edit participants but then only view schedules.

A user name can be up to 50 characters long.

### **Participants**

A participant is someone who takes a session, usually the name or other unique identifier of a real person, e.g. “John Doe.” You can also use a participant to be a generic identifier. For example, a company might set up “Branch manager”, or “Van driver” as a generic identifier for people who can take certain sessions. (If the participant is not a real person, and the real person's name is wanted, this would be asked from within the session itself, e.g. in an introduction question.)

When defining a participant, you set up three text fields:

- The participant name, up to 50 characters, which must be unique in the database. You must use different names for each participant. Some organizations use employee or student numbers for participant names. Others use email addresses. You can use a person's real name, but then you need to be able to cope with duplicates – two people called John Doe for example. In such a case, you will have to alter the way one of these two is entered.
- The participant password, up to 20 characters. This can be set by you, or generated at random by the system. If security is not critical, you can choose to leave passwords blank. Or else base the password on the participant name in some way – for example in a teaching environment, some users have a student number as the participant name and the person's name as the password; this is not fully secure, but makes passwords easy to communicate. A password is not case-sensitive so that a password such as **Secret** will be accepted for both **SECRET** and **secret**.
- You can also define a participant details field of up to 50 characters, containing extra information on the participant. This can be used for any purpose you want, or else left blank. For example, some organizations put a location or department name, or else if the real name is not used as the participant name, then it can be put here.

The participant name and details values are stored in the answer database with the results from a session and can be used in reporting. The password is used only to secure access, and is not stored in the answer database.

## Groups

You can put participants in to groups to make them easy to manage and schedule. It's possible to view participants by groups in *Security Manager*, and also to schedule people by group. You don't have to use groups, but they are very useful, especially for large databases.

As well as being a way of organizing participants, groups also allow you to organize the security access to *Security Manager* and *Enterprise Reporter*.

A group name is up to 50 characters long, and can have any number of participants in it. A participant can be a member of no groups, one group, or any number of groups.

## Schedules

The security database does not hold a list of sessions, these are taken from the session database. However, it does hold a list of schedules, which are assignments of either groups or participants to sessions. When defining a schedule, you set up:

- The schedule name, up to 50 characters. This is usually the session name, but can be different. It appears as the name of the session that the participant can choose to run.
- Which session is scheduled.
- Whether the schedule is for a participant or a group. When a participant is scheduled, it can be scheduled as a member of a group or not. If it is scheduled as a member of a group, then a user assigned to that group can edit the schedule and view reports on the results from the schedule. If it is scheduled as a member of no group, then you need to have permission to view all groups to be able to change the schedule later or to view results from the schedule in *Enterprise Reporter*.
- Whether or not the schedule has a limited number of attempts set, and if so what that limit is (e.g. 5 attempts allowed).
- Whether or not the schedule is only valid in a date and time range, and if so what that range is, e.g. from 9:00 31<sup>st</sup> January 2000 to 12:00 10<sup>th</sup> February 2000.

There can be any number of users, participants, groups and schedules defined, limited only by the capacity of the database engine, the size of your hard disk and the speed of your server. The number of users and participants can also be limited by the contents of your license file.

You are also able to access the security database directly, to make changes programmatically. See Appendix B for details on this.



### 4.1.3 How the information in the security database is used

The **perception.dll** program uses the information in the security database in the following way. It is possible to bypass some of these steps by using parameters to pass the name or session, as described in section 3.3.1.

1. The software asks the participant for his/her name and password. No further interaction is allowed until a participant name and password are entered that match an entry in the database.
2. The software then checks the database for any schedules that the participant is scheduled to, or that any groups that the participant is currently a member of are scheduled to, and lists all the sessions for the participant to choose.

What counts is which groups a participant is a member of at the time **perception.dll** is running, not which groups the participant was a member of when the schedule was made. So if you schedule a session to group X, you can later add people to group X to allow them to take that session.

3. The software then sends a list of sessions that the participant can run. The participant starts the session by choosing from the list.

Once a participant has seen this list, he/she is approved for the session. The security database is not checked again when the session is run. The list is valid for a length of time as specified in **perception.ini** (the value of **Sessions expire**, by default 12 hours). So if a participant gains access to a session, any subsequent change to the security database does not countermand this, until a short time has passed.

4. However, when a session with a limited number of attempts is run, the security database is checked just before the session is run, and the field containing the number of attempts is reduced by one. If no more attempts are permitted, access is denied. This is done in such a way that a participant cannot bypass the system and do the session more times than is permitted.

Note, that limiting a participant to a specified number of attempts at a session is not entirely a reliable process under the HTTP protocol used on the web, because the server can only count the number of times the session is sent to the participant, which can be different to the number of times it is received.

For example, if a participant clicks twice on the button to run a session, or if he/she reloads the first block of a session by mistake, this can count as two attempts. You can view the record of attempts from *Enterprise Reporter*, and if you want to allocate additional attempts, you can do so by editing the schedule.

## 4.2 Using Security Manager

### 4.2.1 Running Security Manager

You run *Security Manager* by pointing your browser at **security.dll**. You need a browser that is able to use cookies, for example under Windows *Internet Explorer* version 4 or higher, or Netscape *Navigator* version 3 or higher. It is possible to use *Internet Explorer* version 3, but some functions may not work fully.

If you are using *Security Manager* frequently under Windows, you may wish to make a Windows desktop shortcut to *Security Manager*, as follows:

1. Go to your Windows desktop.
2. Right-mouse click and select **New** and **Shortcut**.
3. Enter the *Security Manager* URL on your system, for example:

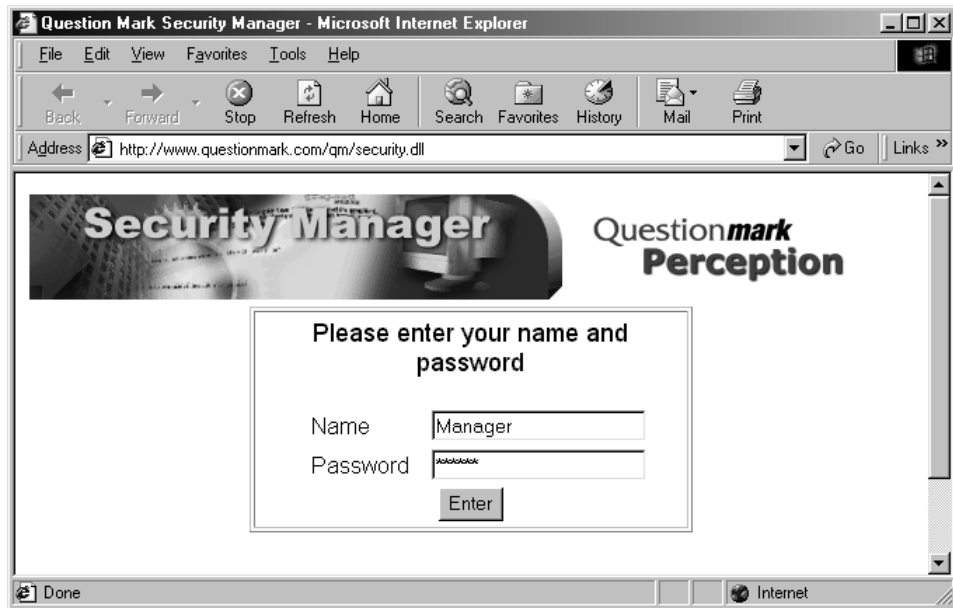
**`http://www.xyzcompany.com/qm/security.dll`**

on the command line.

4. Enter a description, e.g. *Security Manager*, and click **Finish**.

## 4.2.2 Accessing the software (and stopping others from doing so)

*Security Manager* asks you for a name and a password, which you must enter correctly, or else you cannot proceed.



When the software is first installed, the **Name** is set to **Manager** and the **Password** to **Secret**. The name and password are stored in the security database, and you can change them from within *Security Manager*. You should do this soon after installing the software, so that other people cannot access the system.

To change the name and password:

1. Click on the **Admin** button.
2. Select the **Update name and password** radio button.
3. If you are concerned about security, you may want to define a new name instead of **Manager** to log in under. Type this name in the **Manager name** box.
4. Choose a new password and enter it in the **Manager password** box and then retype it in the **Repeat password** box. It's important to choose a password that will not be easily guessed.

5. Click on the **Update** button.



You are recommended to change the password frequently, following the same guidelines used on your system for other important passwords. If someone can guess the password you are using then they could access *Security Manager* and delete participants or change schedules, which could cause serious damage to your assessment system.

Some other security recommendations are:

- Ensure that the security database (**qmpsecurity.mdb**) is itself protected from direct access, e.g. by someone using *Access*. Ensure that there is physical security to prevent people from accessing the computer it is stored on and/or that permissions are set appropriately within the operating system to prevent someone from being able to read from or write to it other than via *Security Manager*.
- To avoid you having to type in a password on every screen, *Security Manager* uses a “cookie” within your browser to record that you are permitted in. The time that this cookie is valid for is defined in **perception.ini**, but is usually one hour. If you keep your browser running after using *Security Manager*, then someone else could come to your computer and go into *Security Manager* using the same cookie for a limited period. If this is a possibility, then you should make sure to exit all copies of your browser after using *Security Manager*, if you are leaving your desk for any reason. Once the browser has been closed the cookie will be removed and the name and password will need to be supplied to use *Security Manager*.

There is a login facility on the **Admin** screen which allows another user to login. You could also login as a user which has less access.

- There is a theoretical possibility that someone could remotely access *Security Manager* with a program that tried all different permutations of name and password until it came up with the right one. All accesses to *Security Manager* are logged in the **perception.log** file, so if you think this is a possibility you should check the log file from time to time.

If you forget your name and password to your security database, you can do one of the following:

1. Replace the security database with the starting one. You will find this on the Perception CD, in the **uncompressed\databases** directory, with name **qmpsecurity.mdb**.
2. Return it to us, and we will reset its password, subject to verification of your identity.

### 4.2.3 The Security Manager screens

There are six views in *Security Manager*:

- The **Admin** button takes you to an administration screen, which allows you to change your login name and password, reset the *Perception Server* DLLs or take it on or offline, and login as another user.
- The **Users** button shows you all the users in the database excluding the root manager name. It also allows you to add new, delete or edit existing users.
- The **Participants** button shows you the participants in the database, and allows you to add more, change them or assign people to sessions.
- The **Groups** button lets you view and add groups, and allows you to assign them to sessions. You add participants to groups in the **Participants** view.
- The **Sessions** button shows you the sessions in the session database.
- The **Schedules** button shows the currently defined schedules.

When you first run *Security Manager*, you view participants, and you can change to any of the other views by clicking the appropriate button. Clicking **Help** at any stage will display online help on using *Security Manager*. If you are logged on as a user without full capabilities, then only some of the views and functions will be available.

The screen layout in *Security Manager* will vary slightly with the browser you are using and the settings on your computer. If you need to change the layout of *Security Manager* for any reason, aspects of the layout are defined in file **manager.format**, which you can change.

## 4.2.4 Users explained

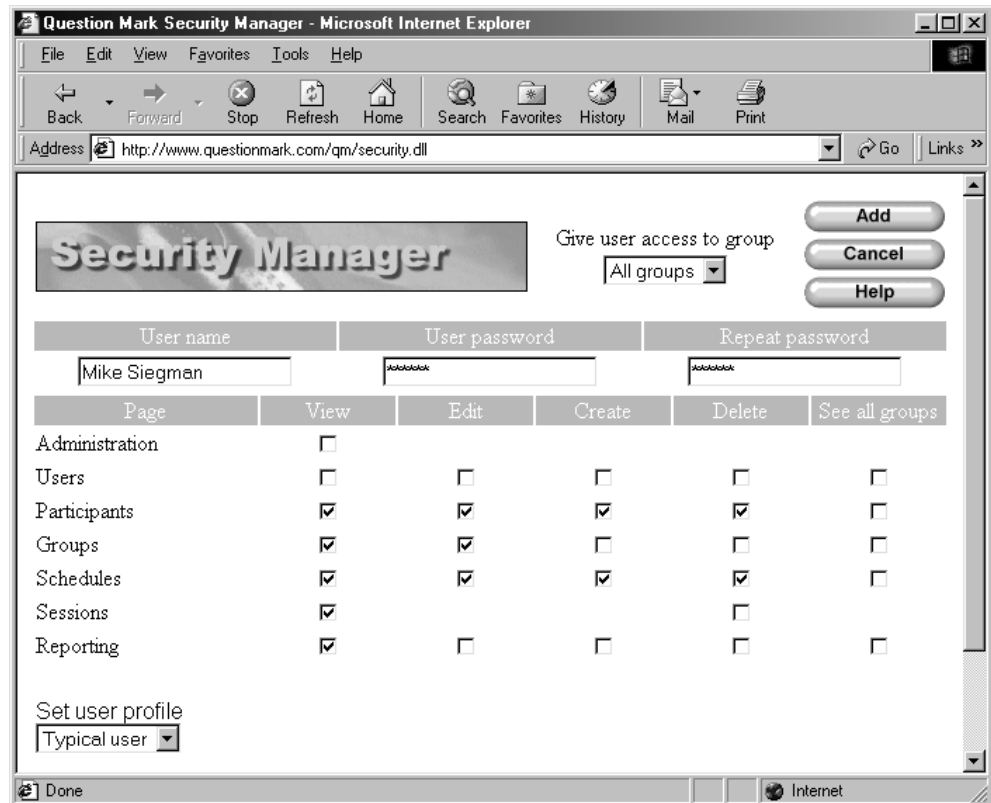
The security database which is shipped on the Perception CD has one user who has full access to the system. In *Security Manager*, you can define extra users, who have different levels of access. These users, who may be authors, instructors, supervisors or whatever, are given names and passwords and capabilities. The number of users permitted on your system may be limited in the license file.

A user may be allocated to just one or some groups, or may have authorization over all groups. A user may be able to do everything in *Security Manager* or *Enterprise Reporter*, or just some things. For example, a user could be permitted to view reports on all groups, but only schedule participants in one group. Or a user could be permitted full access to *Security Manager*, but not allowed entry into *Enterprise Reporter*.

Profiles are available within *Security Manager* so that you can easily define new users. Profiles only affect creating users, and you can freely edit the settings after creating them. The default profiles available are as follows. You can amend the list of profiles or add new ones by editing **manager.format**.

- **Reporter**  
This user will be allowed to login to *Enterprise Reporter*. You can either give access to some groups or all groups.
- **Guest**  
A guest is only given permissions to view the participants, groups, schedules, sessions and login to *Enterprise Reporter*. They are not permitted to view other users or to change the security database.

- **Typical user**  
A **Typical user** has all the permissions of a guest. In addition, they can edit participants, groups and schedules, and also create and delete participants and schedules.



- **Group owner**  
This user has full access to participants, groups, schedules and reporting.
- **Super user**  
A **Super user** has all the permissions of a **Group owner** but in addition to this they can also view other users and delete sessions.
- **Administrator**  
An administrator has no restrictions at all within *Security Manager* except the ability to change the root manager's name and password.

Which groups a user is allocated to only matters if they do not have **See all groups** capability for each area. If a user is restricted to only one or some groups, then they can only view, edit

and delete records for that group. If someone is not in any group or not scheduled as a member of a group, then only someone with access to all groups can view, edit or delete them.

## 4.2.5 Viewing and editing users

New users are created in the **Users** page in *Security Manager*. The root manager can create new users, and can also give other users the power to create users. When users have permission to create or edit other users, they can only give the new users the permissions that they have themselves.

You define for each user a name and password, which is used to log on with. These are case sensitive. And you also can define one or more groups that the user is allocated to. Then you define for each page of *Security Manager*, and also for running *Enterprise Reporter*, what access level the user has:

- **View** means the user can see the page. If you don't have permission to view then the other settings are ignored.
- **Edit** means the user can edit the records on the page. For example if a user has permission to edit participants then they can change the participant name, password, details and which groups they are a member of.
- **Create** means the user can create new records. For example if a user is allowed to create participants then they can make or import new participants.
- **Delete** means the user can delete existing records.
- **See all groups** means the user is not restricted to the groups allocated. For this function, he/she can do things for all groups.



## 4.2.6 Viewing and editing participants

The screenshot below shows ten participants:



If you have a large database with many participants, you can limit the number of participants shown on each page by selecting a number in the **Records per page** area. You can then move forwards and backwards between pages. You can choose to **Sort by** the **Participant name** or **Participant details**.

You can also choose to view participants from just one group by selecting in the **View participants from group** area. In the screenshot all participants are shown. You can also choose to view those just in one group, or else just those participants that are not in any group.

You can search for a particular participant by typing the name, or the first few letters of the name you are searching for in the **Find participant** text box.

You can also select to show or hide passwords.

The following can be done on the **Participant** screen:

- To delete one or more participants, you tick the boxes by the participants you want to delete and click the **Delete** button. Deleting a participant also deletes all the schedules for that participant.
- To edit (or view more details on) a participant, you select the participant and then click **Edit**. This allows you to change the name, details or password, to see which sessions the participant is scheduled to, and to see and change which groups the participant is a member of.
- To schedule participants to a session, tick the boxes by the participants you want to schedule, and click the **Assign** button. See section 4.2.9 for more details on this.
- To add new participants, click the **New** button, and type in the names, details and passwords of up to 20 new participants at a time. The software will generate random passwords, which you can choose to use or override. You can add new participants to one selected group. If you want to add them to more than one group, you must add them and then edit them to do this.

Each participant name in the database must be unique, so you cannot enter a name twice. Depending on what license you have purchased for Perception, there may be a limit to the number of participants you are permitted in your security database. You can extend your license by contacting your local Question Mark office.

- If you want to print a list of participants, you can easily do this from your browser.
- See also section 4.3 on how to import participants from an external file.

## 4.2.7 Viewing and editing groups

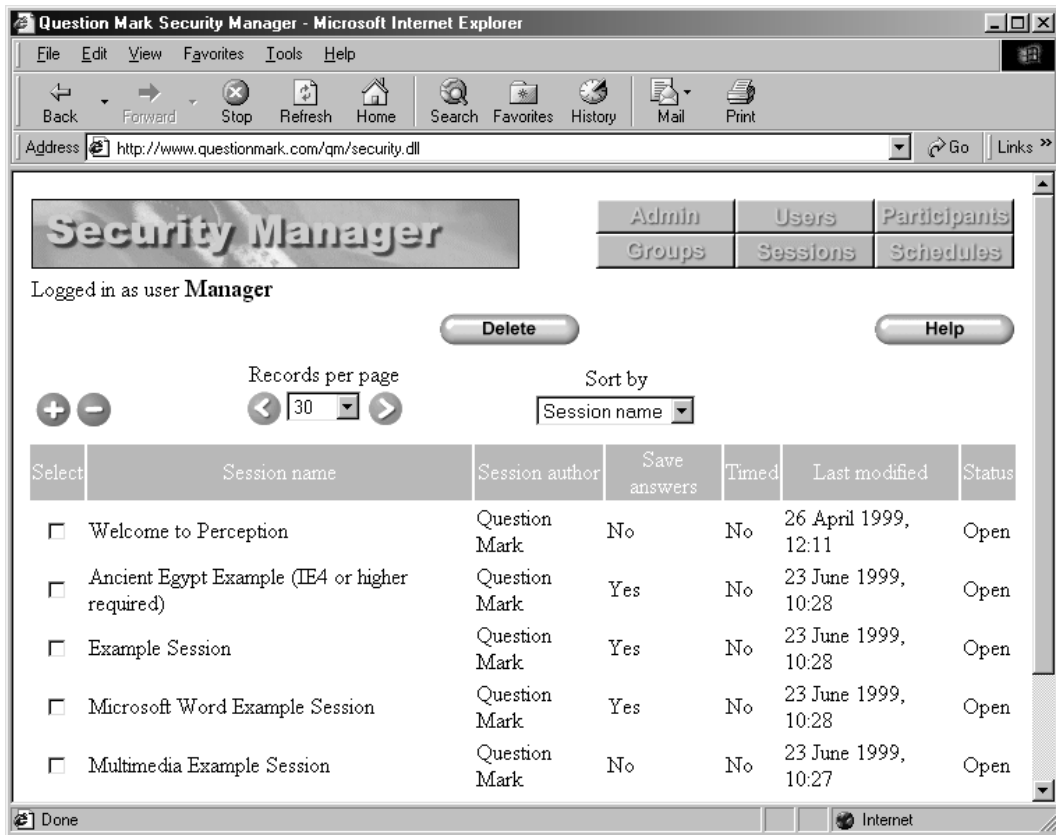
Viewing and editing groups works much like viewing and editing participants, except that there is less to edit. The screenshot below shows five groups:



- To delete one or more groups, you tick the boxes by the ones you want to delete and click the **Delete** button. Deleting a group also deletes all the schedules for that group.
- To change the name of a group, you select it and click **Edit**.
- To schedule a group or groups to a session, tick the boxes by the groups you want to schedule, and click the **Assign** button. See section 4.2.9 for more details on this.
- To add a new group, click the **New** button, and type in the name of the new group. You add participants to groups while viewing or adding participants.

## 4.2.8 Viewing sessions

When you view sessions, you see the list in the current session database. Sessions are added to the session database using *Session Manager*. In the screenshot below, there are five sessions.



You can delete sessions by selecting them, and clicking the **Delete** button. If a session has a password set in *Session Manager*, you must quote this before deleting it. Deleting permanently removes the session from the session database, and once you do it, it cannot be reversed. Deleting a session only effects the session database. It does not delete the questions, graphics or templates that the session references, or any references to the session in the answer database.

To add new sessions, you use *Session Manager*, which is a desktop application.

## 4.2.9 Making a schedule

When you click the **Assign** button, having selected one or more groups or participants, you then view a screen which allows you to assign the groups or participants to a session. Some sessions are password protected in *Session Manager*. If so, you need to quote the password to be able to schedule the schedule.

In the screenshot below, group **Class A** is being assigned to take a session.

Question Mark Security Manager - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print

Address <http://www.questionmark.com/qm/security.dll> Go Links >>

**Security Manager** 1 group selected for schedule.

Add  
Cancel  
Help

Class A

Create schedule for each participant in this group

Schedule name	Session name
<input type="text"/>	Selection Test
Limit attempts	Attempts allowed
<input type="checkbox"/>	1
Limit time	
<input checked="" type="checkbox"/>	
Schedule starts	Schedule stops
Time : 09 :00 Date : 01 December 1999	Time : 17 :00 Date : 20 December 1999

Done Internet

- The **Session name** field shows a list of all the sessions in the session database. You choose here which session is to be scheduled. If you want to assign multiple sessions, you must repeat the scheduling process for each session.
- The **Schedule name** field is the name of the session as shown to the participant when they log in to **perception.dll**. If you leave this blank, as in the screenshot, this will be set to the session name.
- If you want to limit the number of attempts, you tick the **Limit attempts** box, and set the number of attempts in the **Attempts allowed** box. Don't do this if you are scheduling a

group unless you set to create a schedule for each participant in the group, as otherwise the limiting will not make sense. See section 4.1.3 for some more information on how limiting attempts works in practice.

- If you want to make the session only available in a time period, you tick the **Limit time** box, and set the time and date the session is to become available and the time and date it is to be available until. In the screenshot, the session becomes available at 9am on 1<sup>st</sup> December 1999 and stops being available at 5pm 20<sup>th</sup> December 1999. The 24-hour clock system is used (9:00 means 9am, 21:00 means 9pm).

As described in section 4.1.3, there is a short latency time after the expiry of a session (usually one hour), after which it can still be available. Also once a participant starts a session, its expiry here does not stop him/her continuing it.

Any session scheduled also permits the participant to do any session that is branched to at the end of the session.

#### **Something you need to note when you are scheduling a participant**

If you are scheduling participants, it can matter which group the participant is considered to be in for scheduling purposes. This group controls who is allowed to edit the schedule, and also, very importantly, who can view the results in the answer database. If a user is only allowed to view reports on group **Class A**, then if someone is not scheduled as a member of this group, the user will not see reports on them.

To schedule a participant as a member of a particular group:

1. Select the group under **View participants from group** in the **Participants** screen.
2. Tick the boxes for the participants.
3. Click the **Assign** button.
4. You should then see at the top left hand of the scheduling screen **Schedule this participant as a member of group**. Make sure this is checked on.

Alternatively if you want to schedule all members of the group, you can instead schedule the group itself rather than the participants.

To deliberately not schedule a participant as a member of any group:

- Either check **Schedule this participant as a member of group** off.
- Or select to **Show all** on the participants screen, and then the screen to set up the schedule will show **This participant not scheduled as a member of any group**.

You can see what is set by looking at the **Schedules** view. If a participant is scheduled as a member of a group, both the participant name and the group name will be shown here.

**Two options in the way you schedule a group**

When you are scheduling a group, you can choose whether or not to **Create schedule for each participant in this group**.

1. If you select this, then each participant currently in the group is scheduled. So if there are ten members of the group, ten separate schedules are made. Each participant is scheduled separately.
2. If you do not select this, then the group itself is scheduled. Only one schedule record is made. And whoever is a member of the group when **perception.dll** is accessed will be able to take the session.

## 4.2.10 Viewing schedules

Clicking the **Schedules** button shows you a list of all schedules in the database. You can see who is scheduled to what, and how many attempts are left. The list also shows which schedules are active, i.e. currently available, and which are expired. The screen shows one line for each schedule. You can choose to limit the number shown per page and sort the schedules in the usual way.

In the screenshot below, group **Class A** is scheduled to take the session **Selection Test**. And two participants are scheduled to take the session **Microsoft Word Example Session**, with one attempt permitted.

The screenshot shows the 'Security Manager' interface in a Microsoft Internet Explorer browser window. The address bar shows 'http://www.questionmark.com/qm/security.dll'. The interface includes a navigation menu with buttons for Admin, Users, Participants, Groups, Sessions, and Schedules. The user is logged in as 'Manager'. There are buttons for Delete, Edit, and Help. Below these are controls for 'Records per page' (set to 30), 'Sort by' (Schedule name), 'Find participant' (with a search box and Find button), and 'View schedules for group' (with a Show all dropdown).

Select	Schedule name	Session name	Participant name	Group name	Limit attempts	Attempts allowed	Limit time	Schedule starts	Schedule stops	Status
<input type="checkbox"/>	Microsoft Word Example Session	Microsoft Word Example Session	Dan Schumer		Yes	1	No			Active
<input type="checkbox"/>	Microsoft Word Example Session	Microsoft Word Example Session	Lisa Hamilton		Yes	1	No			Active
<input type="checkbox"/>	Selection Test	Selection Test		Class A	No		Yes	01 December 1999, 09:00	20 December 1999, 17:00	Active



If you schedule several participants to do a session, an independent schedule is created for each participant, and this is shown separately on the list. If the group name is also shown, then this means that the participant is scheduled for reporting and editing purposes as a member of this group. If the group name is not shown, then only someone with permission to view all groups will be able to see the results from this schedule in *Enterprise Reporter*.

- To delete a schedule, select it, and click the **Delete** button. This will remove the schedule from the database.
- To edit a schedule, select it and click the **Edit** button. This allows you to change the number of attempts or time period it is valid for, and also to change the schedule name.

If a session is deleted from the session database, its schedule will remain in the security database, unless you specifically delete it. This means that you can set up a schedule for a session, remove the session, and then replace it without losing the schedules. But obviously if someone tries to run a session that isn't there, they will receive a suitable error message.

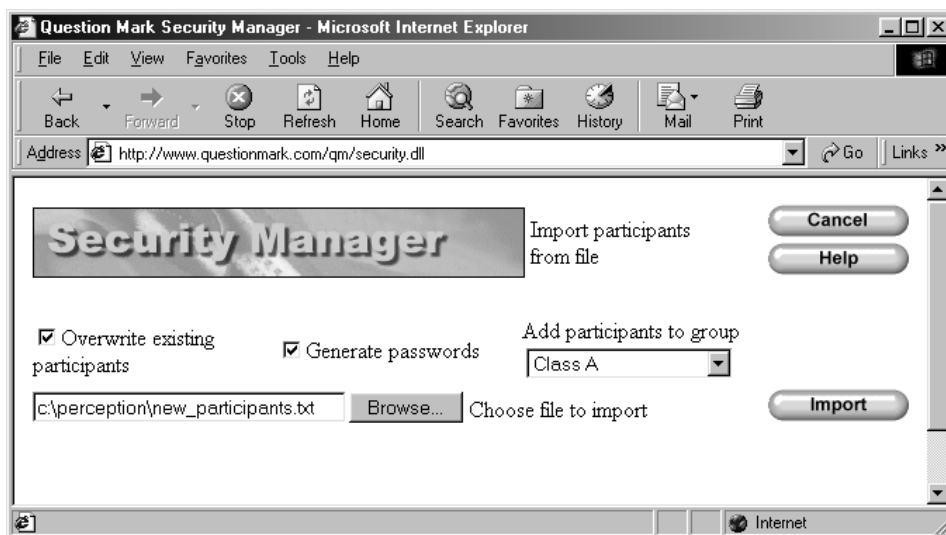
## 4.3 Importing participants from an external file

### 4.3.1 Overview

As well as manually entering participants, you can import them into *Security Manager* from another file. This is useful if you already have a list of participants in another program or database, and want to bring them into Perception.

The import format is comma separated ASCII, which is a very common file format, that is easily created from most databases or program files. See section 4.3.2 for a full description of the file format.

You import participants from the **Participants** screen of *Security Manager*. This happens over the web or intranet connection – you upload a file containing the participants from your local machine into the database on the webserver.



To import a list of participants click the **Import** button from the **Participants** screen then:

1. Choose a file name to import by typing in a file in the text box, or using the **Browse** button to locate it on your computer. If you do not see a **Browse** button, then your browser does not support file upload, and you cannot import participants. You will need to upgrade your browser to a more modern version (e.g. *Internet Explorer 4* or *Netscape Navigator 3* under Windows).

2. Tick the **Overwrite existing participants** box if you want details on any participants that are already in the database to be overwritten with details in the file. Therefore, if this is selected, and participant **John Doe** is already in the security database and is in the import file, the importing will replace any current details or password for the existing entry.
3. Tick the **Generate passwords** box if you want any participants without passwords specified in the file to be given randomly generated passwords.
4. If you want to set that all the imported participants are members of a group, set this under **Add participants to group**. This will add the participants to a single group. If you want to add them to multiple or different groups, you have to do this manually.
5. Click the **Import** button, and the file is uploaded to the server, processed, and the participants are added into the security database.

If you have problems importing, check that the file you are importing is in ASCII format, with commas delimiting fields as set out in section 4.3.2. If in doubt, try viewing it in Windows *Notepad*. If you are creating the file in a word-processor, you have to save it in text format, not in the word-processor's native format.

Depending on what license you have purchased for Perception, there may be a limit to the number of participants you are permitted in your security database. You will not be allowed to import participants if this limit would be exceeded. You can extend your license by contacting your local Question Mark office.

Importing participants is a very powerful tool to assist you in managing a security database. Most simply if you already have your participants set up in another application, you can import them automatically and save a large amount of retyping.

However, you can also use importing to regularly update the Perception security database to keep it synchronized with another database or system. Because it is possible to overwrite existing participants, you can choose to import the whole database or a section of it. For example, supposing you have an existing system with four categories of people, with large numbers of people in each category, with frequent changes of personnel, you could do the following:

- Find a way from your existing system of exporting four separate files, each with a list of people in one category.
- Set up a security database with four groups, one for each category. Make schedules of the groups to do certain sessions.

- At regular intervals (e.g. once a week), export the current contents of the database into four files and import them one by one into the four groups.
- Your Perception security database will then be synchronized with your other system.

## 4.3.2 Format of imported file

The format of files to be imported is comma separated ASCII (sometimes known as comma delimited or CSV file). It should be possible to import these files from non-Windows or PC sources, as well obviously as from Windows machines.

The file must be formatted according to these rules:

- Each line contains information on one participant.
- Each line should be terminated by a standard line end (either hex **0D0A** or **0A**).
- Lines can contain up to three fields: participant name (up to 50 characters), participant details (up to 50 characters) and participant password (up to 20 characters). The first field is mandatory, the other two are optional.
- If the details field is blank, it will be set blank in the database on import. If the password field is blank, it will be set blank too, unless you set **Generate passwords** on import, in which case the software inserts a randomly generated password.
- Each field should be separated from the next one with a comma character.
- You can optionally enclose each field with double quotation mark characters (").
- You cannot use the comma character nor the double quotation mark character " in names, as these have special meaning.

For example, possible formats for lines would be:

Name  
 Name, Details  
 Name,, Password  
 Name, Details, Password  
 "Name", "Details", "Password"

Here is an example import file, importing three participants with no details but a password defined.

**Joanne Badgio, , JX4RTZ**

Denise Thompson, , Y35TLQ  
Tony Zellar, , UEWT4P

It's very important to prepare your files in ASCII format, and not in binary format, as binary files will not import. If using a word-processor, you should save the file as text or ASCII format. If using a spreadsheet or database, you should be able to save it directly in a comma-delimited format.

If you are having trouble importing files, and you should check the format of the file before importing, by opening it in Windows *Notepad*. If you can't view it in *Notepad*, then it's probably not in the right format.

## 4.4 Administration utilities

The administration area allows users with access to perform various tasks such as taking *Perception Server* offline or resetting it, and logging in as another user.

Question Mark Security Manager - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print

Address <http://www.questionmark.com/qm/security.dll> Go Links >>

Change the Security Manager login name and password

Update name only  Update name and password

Manager name	Manager password	Repeat password
Manager		

---

Reset Perception Server DLLs

session.dll  perception.dll  open.dll  security.dll

---

Set Perception Server status

Offline  Online

Offline message

---

Login as different user

Name  Password

Done Internet

### **Changing the master Security Manager name and password**

If you are logged in as the master user, you can change the user name and password – see section 4.2.2 for more details. Other users who are allowed into the **Admin** page cannot do this.

### **Resetting**

You can reset each of the four Perception Server DLLs from this page. See section 3.1.2 for more information on resetting.

To reset a DLL, select one of **session.dll**, **perception.dll**, **open.dll** or **security.dll** and then click the **Reset** button. To reset all four DLLs, do this four times.

### **Taking the server offline or back online**

You can take the part of *Perception Server* that delivers sessions (**session.dll**) offline. This means that if anyone tries to start a session or submit answers to a session in progress, they will get a message telling them that the server is offline temporarily, and asking them to try again. All data submitted is saved and can be resubmitted.

The main reason to take **session.dll** offline is if you are altering the databases or other system files. For example if you want to take a copy of the answer database, you can take **session.dll** offline while you are doing it, so that it won't be updated in the middle of any copying. Or if you FTP a new session database to the server, you can turn off access while you are doing it.

To take the server offline, click on **Offline** and then click the **Set Status** button.

To put it back online, click on **Online**, and click the **Set Status** button. Resetting **session.dll** or restarting the webserver always turns **session.dll** back online.

A standard message defined in the format file is given to people who access **session.dll** while it is offline. You can supplement this message by typing a short additional message in the **Offline message** box.

### **Logging on as a different user**

You can login as a different user. Type in the name and password, remembering that these are case sensitive. And click the **Login** button.

# 5 Enterprise Reporter

## 5.1 Overview

### 5.1.1 Introduction

All Perception results are stored in a single answer database, usually file **qmpanswer.mdb** in Microsoft *Access* '97 format. The same database is used for all sessions taken on the server. Information is written to the answer database as soon as the session is started, and updated as the participant progresses through the session.

*Enterprise Reporter* reads the answer database and presents the information in your web browser. *Enterprise Reporter* needs *Internet Explorer* 4 or higher or *Netscape Navigator* 4 or higher. You cannot use it with earlier model or other browsers. Your browser must have cookies enabled.

*Enterprise Reporter* ships with some standard report definitions and lets authors define their own reports as well. All report definitions are stored in a report database, usually called **qmpenterprise.mdb**.

### 5.1.2 The answer database

When a session is created in *Session Manager*, the author chooses whether results are to be saved. He/she can choose whether to save **Full data**, which includes the answers the participant gives to every question, or just **Summary data**, which only consists of the score and other overall data.

Perception ships with a blank answer database, and you can only make reports when participants have answered one or more sessions. Obviously, you can only report if the session saves its answers. Also, if only **Summary data** is saved then some reports are not available.

The answer database is entirely separate from the security database and session database. Whereas these databases show what sessions are available to take on the server, the answer database records information on which sessions have been taken.

If a session is scheduled in *Security Manager*, or changed in *Session Manager*, nothing happens in the answer database until the session is actually taken. And if a session is deleted from *Security Manager* or removed from the session database, its results still remain in the answer database.

The answer database format is fully documented in Appendix A. Although most users will find that *Enterprise Reporter* gives you all the information you need, you are free to write your own programs to report on or analyze the answer database. You can also use report-generating programs or *Access* to make your own reports.

### 5.1.3 Logging in

The default setup of *Enterprise Reporter* requires you to log in before you can view any reports. You log in from your browser by accessing the following URL:

**<http://www.xyzcompany.com/enterprise/login.asp>**

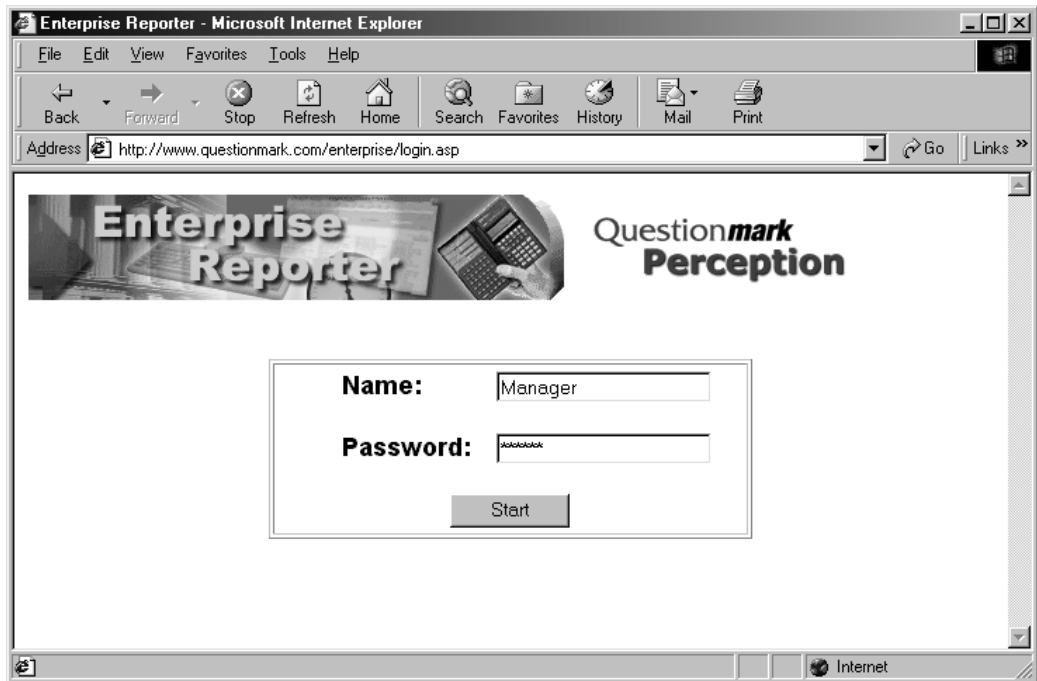
where you should replace **www.xyzcompany.com** with your organization's URL.

See section 2.2.3 for how you can set up *Enterprise Reporter* so as not to need login. If set up in this way (with **CF\_Secure\_Access** set to **0** in **settings.asp**), then you run *Enterprise Reporter* by going directly to: **<http://www.xyzcompany.com/enterprise/index.asp>**.

If you use *Enterprise Reporter* frequently, you may well wish to bookmark or add the URL to your favorites to run it.

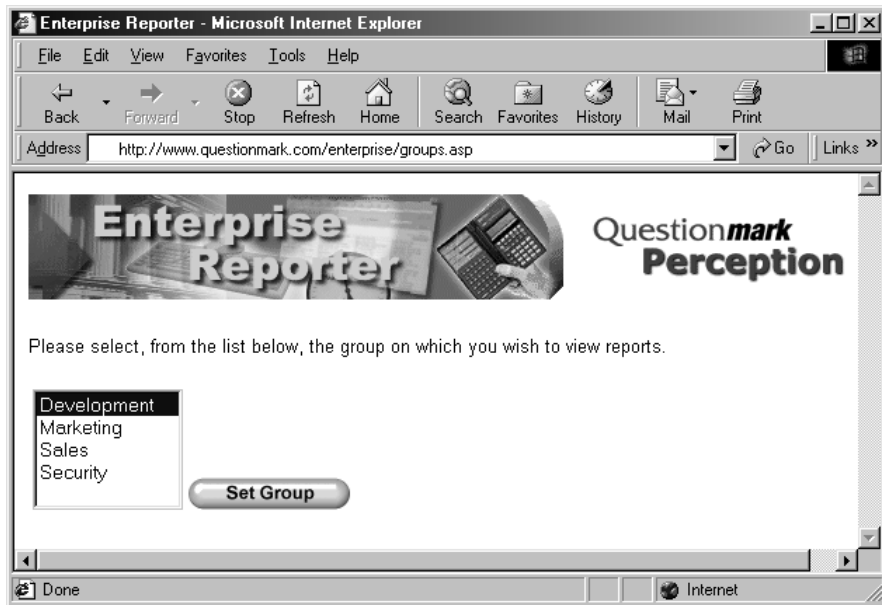
If you log in securely, you must enter your name and password in the login dialog as in the screenshot below. You use the name and password for a user defined in *Security Manager* – see section 4.2 for how to set up these names. You can also use the master *Security Manager* name and password, which starts as **Manager** and **Secret** until it is changed.





If a user has permission in *Security Manager* to view all groups, then successful login takes you to the *Enterprise Reporter* home page. And the user will be able to report on all data in the answer database.

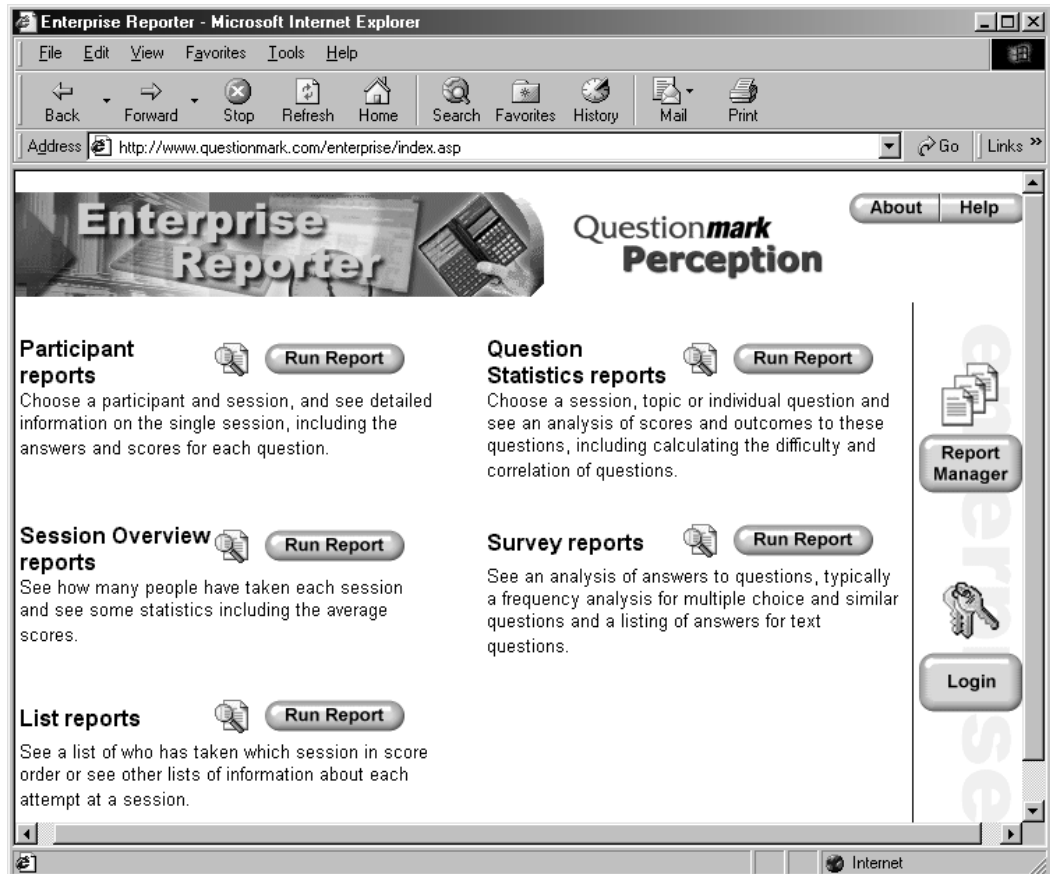
If a user has permission in *Security Manager* to view only one group, then successful login will also take you to the home page. But you will only be able to see reports where the participant was scheduled as a member of this group. All data from sessions where no group is set or where a different group is set will be filtered out, so you won't see the data in reports. If a user has permission to view multiple groups, but not all groups, then the user must choose one of these groups to view in *Enterprise Reporter*. You see a screenshot similar to the one below, and choose one group to log in with. You are then restricted to viewing this group only until you log in again.



Participants can also log in to *Enterprise Reporter* to see reports on their own results. See section 5.8 for more information on this.

## 5.1.4 The Enterprise Reporter Home page

When you first run *Enterprise Reporter*, you see its home page as in the screenshot below. You also come back to this page whenever you click the **Home** button on any *Enterprise Reporter* screen.



There are five report types in *Enterprise Reporter* documented in this manual. Additional report types may be added in the future.

- **Participant reports** are reports with information on a single participant taking a single session. They typically contain full details of the results and answers in that session, and are useful as a record of the session or to help coach the participant on his/her errors. See section 5.2 for a full description of these reports.

- **Session Overview reports** gives you summary information on one or more sessions, including how many people have started and finished the session. See section 5.3 for a full description of these reports.
- **List reports** allow you to make lists or other reports with one line per participant taking a session. They are useful to make class lists or summary listings of results. See section 5.4 for a full description of these reports.
- **Question Statistics reports** analyze the answers to questions, giving you data on scores and outcomes, and allowing you to measure the difficulty and quality of questions. See section 5.5 for a full description of these reports.
- **Survey reports** also analyze the answers, but in a format especially useful for surveys. The report typically gives a frequency analysis or listing of answers. See section 5.6 for a full description of these reports.

Clicking the **Run Report** button by each report description will allow you to view the report. To create, edit and delete report definitions, click the **Report Manager** button. See section 5.1.5 below for more on this.

Clicking **Help** on this or other screens gives you context sensitive online help on using the software. Clicking the **About** button will give you version and system information. Clicking the **Login** button will allow you to re-login to *Enterprise Reporter* as a different user, or if you are restricted to one of many groups, selecting **Change Group** will let you change your group.

## 5.1.5 Report Manager

The *Report Manager* part of *Enterprise Reporter* lets you create, edit and delete report definitions. You are only allowed access to these if permitted in *Security Manager*.



Report definitions are stored in the reports database (usually **qmpenterprise.mdb**). The starting database includes some standard reports, but you can freely edit these or remove them if you choose. There is no password control on report definitions. Anyone who has permission to edit reports can edit any report definition, and similarly anyone who has permission to delete reports can delete any one.

Each report has a name, which defines the report identity. This name can be up to 50 characters long, with any characters except the apostrophe ( ' ) and pipe ( | ) characters in it. Names are not case sensitive – **MY REPORT** and **my report** are the same reports.

To create a report, click the **Create** button by the appropriate report type. You then set a report name, and can choose to **Save and Exit**, which will bring you back to the *Report Manager* screen, or else **Save and View**, which will take you straight to viewing the report.

To edit a report, select it from the drop-down menu, and click the **Edit** button. You then make any changes, and can if you wish also change the name, which will cause a new report to be created. Again you click **Save and Exit** or **Save and View**.

To delete a report, select it from the drop-down menu, and click the **Delete** button. You will be asked to confirm that you want to delete the report definition, and then it will be permanently removed from the database.

## 5.2 Participant reports

### 5.2.1 An example Participant report

A **Participant report** is a report on the results of one instance of a participant taking a session.

It consists of three sections:

- A header section with information about the session, usually including the participant name, session name and score achieved.
- A section with information about each question, which can contain the answer given, the score and the feedback, or other information.
- A topic analysis section, with the scores in each topic of the session.

What is included in the report is entirely configurable by settings in the report definition. The report can be laid out for the screen or for printing. Here is a fragment of an example typical report, laid out for printing.

**Participant report** [Home](#)

---

Total score:	6	Date/time started:	05 July 1999, 11:54
Maximum score:	8	Date/time finished:	05 July 1999, 11:54
Percentage score:	75%		
Status:	Finished Normally		

**Questions shown:** Only wrong questions

Question description:	What is the capital city of the Philippines?
Topic:	Example Question Types
Question type:	Multiple Choice
Answer given (full):	Jakarta

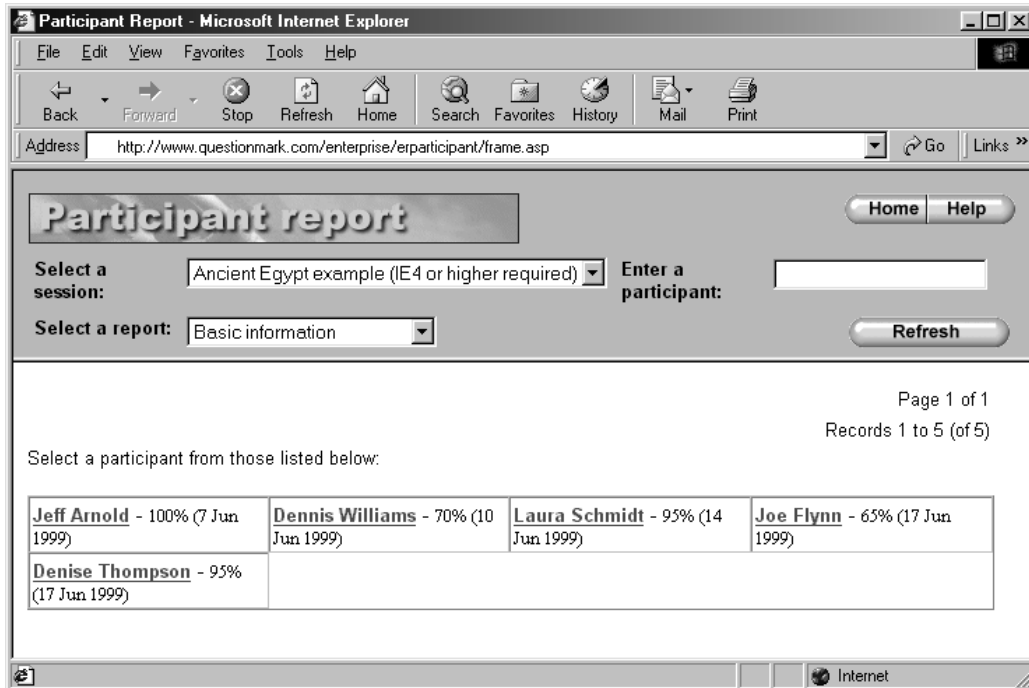
  

Topic name	Point scoring	Percentage scoring	Number of questions
Example Question Types	6/8	75%	8

## 5.2.2 Choosing a Participant report to view

Each **Participant report** contains information on one participant taking one session. Here is how you choose your report:

1. Click the **Run Report** button for this report type on the home page.
2. Choose the session you want to see reports from, by choosing from those available in the **Select a session** drop-down box.
3. Choose the report definition you want to use from those available in the **Select a report** drop-down box.
4. You will then see a list at the bottom of the screen of all the participants who have taken the session, along with the score achieved and the date the session was run. Results are usually shown in date order, i.e. earliest first. The screenshot below shows an example:



Participant report

Select a session: Ancient Egypt example (IE4 or higher required) Enter a participant:

Select a report: Basic information Refresh

Page 1 of 1  
Records 1 to 5 (of 5)

Select a participant from those listed below:

<u>Jeff Arnold</u> - 100% (7 Jun 1999)	<u>Dennis Williams</u> - 70% (10 Jun 1999)	<u>Laura Schmidt</u> - 95% (14 Jun 1999)	<u>Joe Flynn</u> - 65% (17 Jun 1999)
<u>Denise Thompson</u> - 95% (17 Jun 1999)			



5. You can choose to limit the results to show only those for one participant by putting the participant name in the **Enter a participant** box and clicking **Refresh**. The **Refresh** button can also be used at any time to refresh the page.

If there are more than 100 results, these are shown in pages, and you can use the forward and backward arrows to move between pages.

6. To view a report, you click on the participant's name on the screen, and the report is displayed.

If a user is only permitted to view results from a single group, then only results from this group are shown. Otherwise, no filtering applies to the selection of results. You can use the list report as an alternative way of choosing participant reports if you want to view reports with some sort of filtering criteria.

## 5.2.3 Participant report contents

There are four parts to defining a participant report:

### Layout information:

- Participant reports can be laid out in two ways. Choose **Screen view** for a layout suited to screen viewing or choose **Printable** for a report you plan to print.
- The **Font base size** means the size of the font used in reports. Size **1** is smallest, size **2** is medium and size **3** is large.
- The **page header text** is text displayed at the top of the report, and the **page footer text** is displayed at the bottom of the report. Each piece of text can be up to 1,000 characters long. You can put a title here or introduction, reference or identification information here. Expert users can put HTML codes, for example references to graphics. You should avoid apostrophe ( ' ) or pipe ( | ) characters.

Participant report

Help

Please fill in the details below to customize your report: Save and Exit Save and View Cancel

New report name:

Layout information: [more information](#)

Choose report format:  Screen view  Printable

Font base size:

Enter page header text:

Enter page footer text:

### Header information:

In this section, you can choose whether to include information in the header of the report. To include a particular value, tick its box. To include them all, click the **Select all** button. You can also choose alternative labels. For example, the participant name is usually described as **Participant name**. If you want it to be called, for example, **Student name**, then put this in the **Alternative label** column.

**Header information:** [more information](#)

Choose fields to display in header:

Field name	Selected	Alternative label
Participant name:	<input type="checkbox"/>	
Participant group:	<input type="checkbox"/>	
Participant details:	<input type="checkbox"/>	
Session author:	<input type="checkbox"/>	
Session name:	<input type="checkbox"/>	
Session ID:	<input type="checkbox"/>	
Session last modified:	<input type="checkbox"/>	
Date/time of making report:	<input type="checkbox"/>	
Date/time started:	<input type="checkbox"/>	
Date/time finished:	<input type="checkbox"/>	
Time taken:	<input type="checkbox"/>	
Status:	<input type="checkbox"/>	

Total score:	<input type="checkbox"/>	
Maximum score:	<input type="checkbox"/>	
Percentage score:	<input type="checkbox"/>	
Questions presented:	<input type="checkbox"/>	
Questions answered:	<input type="checkbox"/>	
Score band title:	<input type="checkbox"/>	
Special 1:	<input type="checkbox"/>	
Special 2:	<input type="checkbox"/>	
Special 3:	<input type="checkbox"/>	
Special 4:	<input type="checkbox"/>	
Special 5:	<input type="checkbox"/>	
Special 6:	<input type="checkbox"/>	
Special 7:	<input type="checkbox"/>	
Special 8:	<input type="checkbox"/>	
Special 9:	<input type="checkbox"/>	
Special 10:	<input type="checkbox"/>	

Select all Clear all

The following data is available.

<b>Participant name</b>	This is the name of the participant, as passed from <b>perception.dll</b> or <b>open.dll</b> .
<b>Participant group</b>	The participant is in this group. If the session was started by <b>perception.dll</b> , then this is set according to what is scheduled in <i>Security Manager</i> .  If using <b>open.dll</b> , then it is whatever is entered into or passed to the program.
<b>Participant details</b>	If using <b>perception.dll</b> , this is the details field as set in the security database, or if none is set, it is blank.  If using <b>open.dll</b> , this is always blank.
<b>Session author</b>	This is the author of the session, the name set in the session database when the session was created in <i>Session Manager</i> .
<b>Session name</b>	This is the name of the session, as set in the session database by <i>Session Manager</i> .
<b>Session ID</b>	This is the unique numeric identifier for the session.
<b>Session last modified</b>	This is the date the session was last modified or changed in <i>Session Manager</i> .  It is separately recorded for each time the session is taken. So if two participants show different values for this field for the same session, then you know that the session has been edited between the time that the two people took it.
<b>Date/time of making report</b>	This is the date and time of producing the report.
<b>Date/time started</b>	This is the date and time when the session was begun.
<b>Date/time finished</b>	If the session has finished, this is the date and time when the session was finished.
<b>Time taken</b>	This is only set if the session is finished, and is the time taken as measured by the server between the session start and when the score band page is generated and sent to the participant.
<b>Status</b>	This tells you what state the session is in: <ul style="list-style-type: none"> <li>• <b>Finished Normally</b> means that the session has concluded in the normal way.</li> <li>• <b>Recently Started</b> means that the session has started, but no answers have yet been received back.</li> </ul>

	<ul style="list-style-type: none"> <li>• <b>In Progress</b> means that the session has started, and answers from at least one block have been received. But the session is not yet over. (A session counts as finished when the score band is calculated.)</li> <li>• <b>Timed Out</b> means that the session time limit was exceeded. Usually sessions are automatically submitted before this can happen, so such statuses should be unusual.</li> </ul>
<b>Total score</b>	This is the total score in points achieved by the participant. If a session is in progress, this is the current value, up to and including the last block submitted.
<b>Maximum score</b>	This is the maximum score in points that the participant could achieve. If a session is in progress, this is the current value, up to and including the last block submitted.
<b>Percentage score</b>	This is only set if the session is finished, and is the <b>Total Score</b> divided by the <b>Maximum Score</b> , shown as a percentage.
<b>Questions presented</b>	This is the number of questions sent to the participant (ignoring explanation questions).
<b>Questions answered</b>	This is the number of questions answered.
<b>Score band title</b>	This is only set if the session is finished, and is the title of the score band or end block achieved.
<b>Special 1 to Special 10</b>	<p>These fields are set only if a question is set in <i>Session Manager</i> to correspond to one of the fields.</p> <p>For example, if you have a question in an introduction block asking someone their department name, it is possible in <i>Session Manager</i> to assign its answer to one of ten special fields, for example <b>Special 1</b>. When the question is answered, the value goes into this field and is displayed in the report.</p> <p>If using these fields, you will want to set a suitable alternative label for the fields.</p>

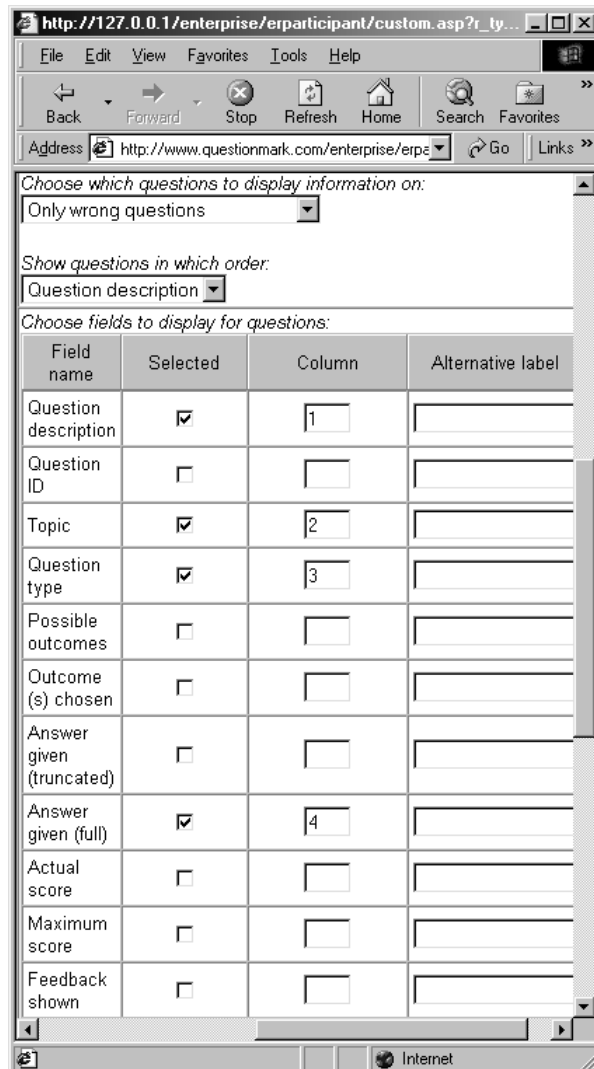
**Question detail information:**

This area allows you to define what information is displayed for each question presented to the participant. Firstly, you choose which questions to display:

- **All questions** means display information about all questions presented in the session.
- **Only answered questions** means exclude questions that were left unanswered.
- **Only unanswered questions** means only display unanswered questions.
- **Only wrong questions** means only display answered questions where the score was less than the maximum score.
- **Wrong & unanswered questions** means display unanswered questions and answered ones where the score was less than maximum.
- **Only correct questions** means only display answered questions where the score is the maximum score.
- **No questions** means do not display information about questions. This means that the report consists only of the header and any topic analysis.

Then you choose which order to display them in. You can choose from ordering by **Question ID**, **Topic** or **Question description**.

Finally, you choose which values to display for each question, which order they are in and what they are called. To have the information shown, select its box in the **Selected** column. To define the order of the information put numbers from 1 to 11 in **Column**. To change the description or label of the information in the report itself, fill in the **Alternative label**.



The possible values that can be displayed for each question are:

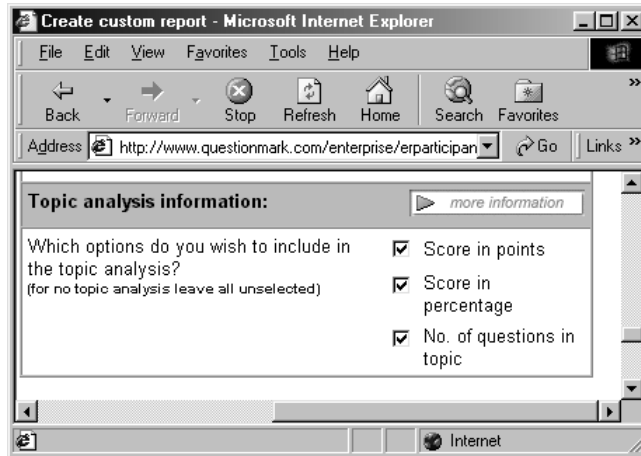
<b>Question description</b>	<p>This is the text description of the question as defined in <i>Question Manager</i>.</p> <p>Note that the question wording is not stored in the answer database and cannot be given in reports. So it's important to set the question description to something that is useful in reports.</p>
<b>Question ID</b>	This is the internal unique numeric identifier for the question.
<b>Topic</b>	The question is in this topic.
<b>Question type</b>	This is the question type.
<b>Possible outcomes</b>	This is a list of all the possible outcome names for the question.
<b>Outcome(s) chosen</b>	This is a list of the outcome or outcomes chosen by the participant. See the <i>Authoring manual</i> for a full description of outcomes.
<b>Answer given (truncated)</b>	This is the answer to the question, truncated to 50 characters.
<b>Answer given (full)</b>	<p>This is the full answer to the question.</p> <p>It contains the same information as <b>Answer given (truncated)</b>, except if the answer is longer than 50 characters, it is given in full here.</p>
<b>Actual score</b>	This is the score in points for the question.
<b>Maximum score</b>	This is the maximum score possible for the question.
<b>Feedback shown</b>	<p>This feedback arises from the outcomes chosen.</p> <p>Even if the feedback is not shown to the participant, for example because it is set not to be shown in the session, it is still stored here.</p> <p>Because there can be many different correct answers to some questions, there is no column for a "correct answer" in <i>Enterprise Reporter</i>. If you want to include a "correct answer" in your reports, you should put the correct answer into your feedback wording in <i>Question Manager</i>, and then display it in reports.</p>



**Topic analysis information:**

This allows you to include a breakdown of score by topic at the end of the participant report.

You can choose to include in the analysis the **Score in points**, **Score in percentage** and **No. of questions in topic**. Select each one to include it, or leave them all empty to not display a topic analysis.

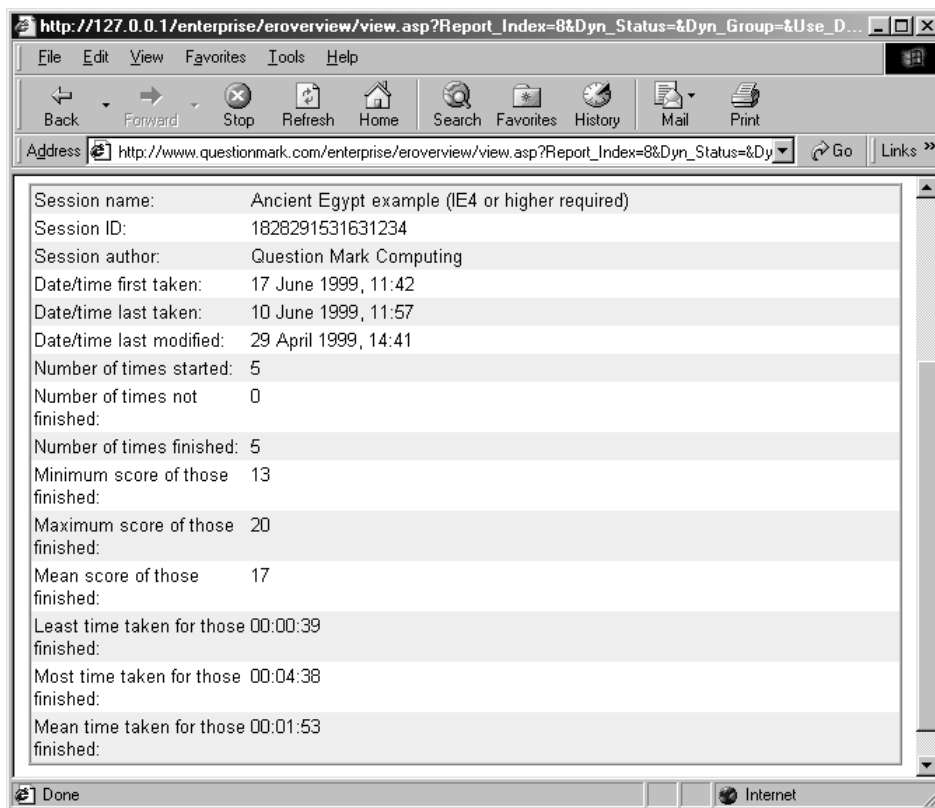


## 5.3 Session Overview reports

### 5.3.1 An example Session Overview report

A **Session Overview report** gives some statistics about the number of times each session has been started and finished, and the average scores and times taken. You can either make the report for all sessions, or for selected sessions.

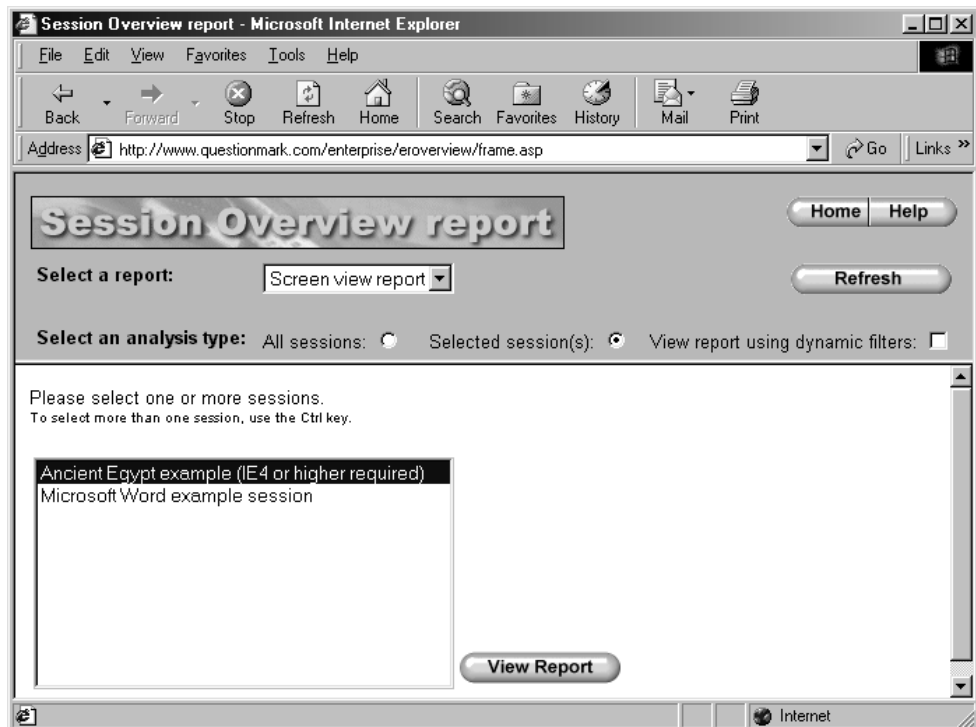
The report can be laid out for the screen or for printing. Here is a fragment of an example typical report, laid out for printing.



## 5.3.2 Choosing a Session Overview report to view

Here is how you view a **Session Overview** report.

1. Click the **Run Report** button for this report type on the home page.
2. Choose the report definition you want to use from those available in the **Select a report** drop-down box.
3. Select either **All sessions** or **Selected session(s)**. If you choose to select sessions, then to choose one session, click on it. To choose multiple sessions, hold the **Ctrl** key down whilst clicking.
4. Click the **View Report** button.



See section 5.7 for a description of how you can also select which data is shown by using filters.

### 5.3.3 Session Overview report contents

When defining a **Session Overview report**, you specify the font size, whether the report is formatted for **Screen view** or if it is **Printable**, and optional header and footer text. These all work in a similar way to the **Participant report** described in section 5.2.3. You can also set filter data for the report as described in section 5.7.

The contents of the report are defined in the **Session information** section of the report. You first choose the order in which sessions are displayed if information on more than one session is displayed in the report. You can choose between alphabetical ordering of **Session name**, **Session ID** or **Session author**.

Session information: [more information](#)

Select the session ordering method:

Select which session fields you require:

Field name	Selected	Order	Alternative label
Session name	<input checked="" type="checkbox"/>	<input type="text" value="1"/>	Survey name
Session ID	<input checked="" type="checkbox"/>	<input type="text" value="2"/>	
Session author	<input checked="" type="checkbox"/>	<input type="text" value="3"/>	
Date/time first taken	<input checked="" type="checkbox"/>	<input type="text" value="4"/>	
Date/time last taken	<input checked="" type="checkbox"/>	<input type="text" value="5"/>	
Date/time last modified	<input checked="" type="checkbox"/>	<input type="text" value="6"/>	
Number of times started	<input checked="" type="checkbox"/>	<input type="text" value="7"/>	
Number of times not finished	<input checked="" type="checkbox"/>	<input type="text" value="8"/>	
Number of times finished	<input checked="" type="checkbox"/>	<input type="text" value="9"/>	

You can then select each or all of the following. If you want to change the name of each piece of information in the report, you can do so by filling in the **Alternative label**. For example if you want to include the **Session name** but describe it as **Survey name**, then tick the box under **Selected**, and enter **Survey name** under **Alternative label**. You can also set the **Order** of fields by entering numbers from 1 to 15 by data.

<b>Session name</b>	This is the name of the session, as set in the session database by <i>Session Manager</i> .
<b>Session ID</b>	This is the unique numeric identifier for the session.
<b>Session author</b>	This is the author of the session, the name set in the session database when the session was created in <i>Session Manager</i> .
<b>Date/time first taken</b>	This is when the session was first taken.
<b>Date/time last taken</b>	And this is when it was most recently taken.
<b>Date/time last modified</b>	This is the date and time when the session was last changed in <i>Session Manager</i> , prior to the first set of results being stored in the answer database.
<b>Number of times started</b>	Number of times the session has been started.
<b>Number of times not finished</b>	Number of times the session has been started but not finished.
<b>Number of times finished</b>	Number of times the session has been started and finished.  A session is considered finished if the score band has been calculated.
<b>Minimum score of those finished</b>	Lowest score in points achieved by a participant who finished the session.
<b>Maximum score of those finished</b>	Highest score in points achieved by a participant who finished the session.
<b>Mean score of those finished</b>	Average score in points achieved by participants who finished the session.
<b>Least time taken for those finished</b>	Shortest time taken for someone to complete the session.
<b>Most time taken for those finished</b>	Longest time taken for someone to complete the session.
<b>Mean time taken for those finished</b>	Average time taken for someone to complete the session.

## 5.4 List reports

### 5.4.1 Example List report

A **List report** is a list of results. In screen view, it contains one line in the report corresponding to one person taking a session. Therefore, if the answer database shows that 100 participants have each taken 5 sessions, then if there is no filtering, a **List report** will contain 500 lines or rows of information. If the lines are long, you can also make a version of the report to be printed with shorter lines.

You have a lot of flexibility as to what is shown on each line of the report. You can include a link that users can click on to get a participant report for fuller information and you can also include some information on questions answered within the report.

Here is an example list report fragment showing some summary information. The user can click on the icon in the **Detailed information** column to get more information.



Report generated: 17 June 1999, 13:02

Filters for this report:  
No filters specified.

Detailed information	Session name	Participant	Percentage score
	Ancient Egypt example (IE4 or higher required)	Denise Thompson	95
	Ancient Egypt example (IE4 or higher required)	Dennis Williams	70

Here is an example fragment of a list report where the scores to the first few questions of a survey are shown for each person.

**List report** [Home](#)

Report generated: 17 June 1999, 13:19

Filters for this report:

Session name	Participant	Percentage score	Question 1	Question 2
Ancient Egypt example (IE4 or higher required)	Denise Thompson	95	2	0
Ancient Egypt example (IE4 or higher required)	Dennis Williams	70	2	0

Done Internet

## 5.4.2 Choosing to view a List report

Here is how you view a **List report**.

1. Click the **Run Report** button for this report type on the home page.
2. Choose the report definition you want to use from those available.
3. Click the **View Report** button.



Filters are particularly useful in the **List report**, as if you do not use them you will get a list of all the results in the database. See section 5.7 for a description of how you can also select which data is shown by using filters.



## 5.4.3 List report contents

When defining a **List report**, there are four sections of information to define.

### Layout information

This allows you to define the font size, header and footer text and whether the report is for screen view or printing, similarly to the **Participant report** as described in section 5.2.3.

You can also define the **Number of records per page**, which is the number of records shown in each document sent to the browser. If you have a very long report, it's usually easiest to show it over multiple pages so that you don't have to wait for the server to compile the whole report before you see the first line.

**List report - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail Print

Address [http://www.questionmark.com/enterprise/erlist/custom.asp?r\\_type=list&Report\\_Name=Basic+list+report&iSubmit=2&](http://www.questionmark.com/enterprise/erlist/custom.asp?r_type=list&Report_Name=Basic+list+report&iSubmit=2&) Go Links >>

## List report

Help

Please fill in the details below to customize your report: Save and Exit Save and View Cancel

Editing report: Basic list report

### Layout information:

more information

Font base size: 2

Number of records per page (range 10 to 1000) (leave blank for all)

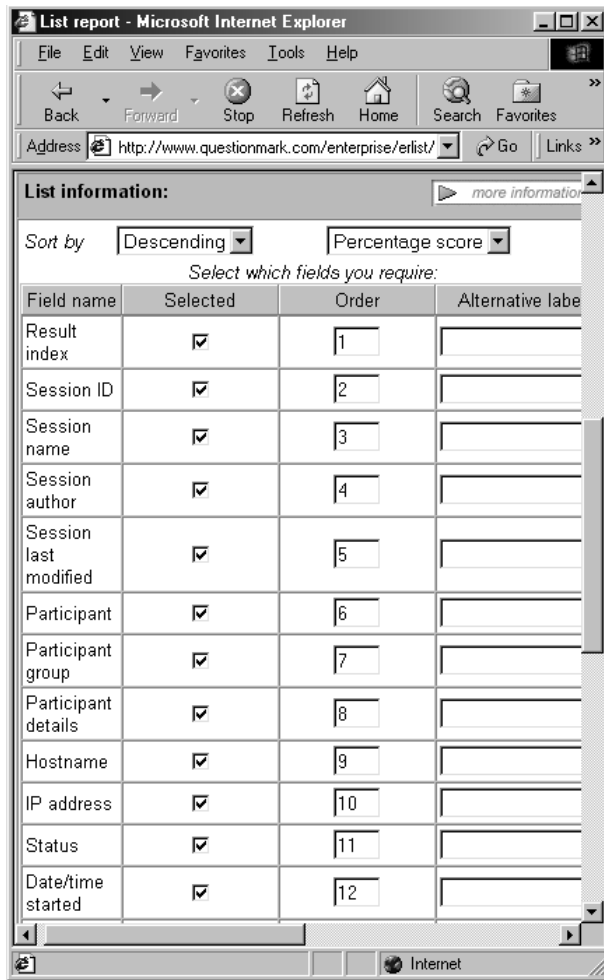
Choose report format:  Screen view  Printable

Enter page header text: Enter page footer text:

Internet

### List information

You can choose to sort the results in the **List report** in **Ascending** or **Descending** order of **Participant**, **Group**, **Session name**, **Date started**, **Status** and **Percentage score**. For example, if you choose **Descending** order of **Percentage score**, then the highest scoring results will be shown first.



You can then choose which of the following data you want to include in your report. Tick the **Selected** column to include each item of data, fill in the **Order** column with numbers from 1 to 28 to define the order of data, and use the **Alternative label** column to name the data the way you want it.

<b>Result index</b>	Index number of the results data in the answer database.
<b>Session ID</b>	This is the unique numeric identifier for the session.
<b>Session name</b>	This is the name of the session, as set in the session database by <i>Session Manager</i> .
<b>Session author</b>	This is the author of the session, the name set in the session database when the session was created in <i>Session Manager</i> .
<b>Session last modified</b>	This is the date the session was last modified or changed in <i>Session Manager</i> .  It is separately recorded for each time the session is taken. So if two participants show different values for this field for the same session, then you know that the session has been edited between the time that the two people took it.
<b>Participant</b>	This is the name of the participant, as passed from <b>perception.dll</b> or <b>open.dll</b> .
<b>Participant group</b>	This is the group that the participant is in. If the session was started by <b>perception.dll</b> , then this is set according to what is scheduled in <i>Security Manager</i> . If using <b>open.dll</b> , then it is whatever is entered into or passed to the program.
<b>Participant details</b>	If using <b>perception.dll</b> , this is the details field as set in the security database, or if none is set, it is blank.  If using <b>open.dll</b> , this is always blank.
<b>Hostname</b>	This is the computer name from which the participant logged in to start the session, if this is known.  In some circumstances, this information will not be available. If not known, the IP address is displayed.
<b>IP address</b>	This is the identity of the computer from which the participant logged on to start the session, as reported to the server.
<b>Status</b>	This tells you what state the session is in: <ul style="list-style-type: none"> <li>• <b>Finished Normally</b> means that the session has concluded in the normal way.</li> <li>• <b>Recently Started</b> means that the session has started, but no answers have yet been received back.</li> <li>• <b>In Progress</b> means that the session has started, and answers from at least one block have been received.</li> </ul>

	<p>But the session is not yet over. (A session counts as finished, when the score band is calculated.)</p> <ul style="list-style-type: none"> <li>• <b>Timed Out</b> means that the session time limit was exceeded. Usually sessions are automatically submitted before this can happen, so such statuses should be unusual.</li> </ul>
<b>Date/time started</b>	This is the date and time when the session was begun.
<b>Date/time finished</b>	If the session has finished, this is the date and time when the session was finished.
<b>Total score</b>	This is the total score in points achieved by the participant. If a session is in progress, this is the current value, up to and including the last block submitted.
<b>Maximum score</b>	This is the maximum score in points that the participant could achieve. If a session is in progress, this is the current value, up to and including the last block submitted.
<b>Percentage score</b>	This is only set if the session is finished, and is the <b>Total Score</b> divided by the <b>Maximum Score</b> , shown as a percentage.
<b>Time taken</b>	This is only set if the session is finished, and is the time taken as measured by the server between the session start and when the score band page is generated and sent to the participant.
<b>Score band title</b>	This is only set if the session is finished, and is the title of the score band or end block achieved.
<b>Special 1 to Special 10</b>	<p>These fields are set only if a question is set in <i>Session Manager</i> to correspond to one of the fields.</p> <p>For example, if you have a question in an introduction block asking someone their department name, it is possible in <i>Session Manager</i> to assign its answer to one of ten special fields, for example <b>Special 1</b>. When the question is answered, the value goes into this field and is displayed in the report.</p> <p>If using these fields, you will want to set a suitable alternative label for the fields.</p>

## Other information

The screenshot shows a web browser window titled "List report - Microsoft Internet Explorer". The address bar contains the URL "http://www.questionmark.com/enterprise/elist/custom.asp?t\_type=1". The main content area is titled "Other information:" and contains the following settings:

- Link to participant report (Screen view only)
- Participant report name: Full report on screen
- Display question information
- Number of questions to display: 3
- What information to display: Score
- Label text (blank for default):

This section allows you to add some valuable information into the **List report**.

If you select **Link to participant report** an icon will be displayed by each set of results. Then if the user clicks on this icon, he/she will see more information on the results. This is in the form of a **Participant report**. You can choose which report by selecting from the drop-down menu by **Participant report name**.

If you select **Display question information**, then you can have some information on questions included in the report. You define this as follows:

- Set **Number of questions** to display to choose how many columns of questions to include in the report, from 1 to 100. For example, if you put 10 here, then the answers to the first ten questions will be shown. The numbering is the numbering shown to the participant, so the third column will show the third question asked to that participant. If question order is randomized, then answers to the same question may appear in different columns.
- You then choose **What information to display**. You can choose to display the **Outcomes**, **Full answer**, **Truncated answer** or **Score** to each question. If truncated, the answer is limited to 50 characters.
- You can if you wish define some **Label text**. Leave this blank to use the default label of **Question**. Whatever label you choose is followed by the question number.

Question information is always displayed after the other information in the list report.

#### Filter information

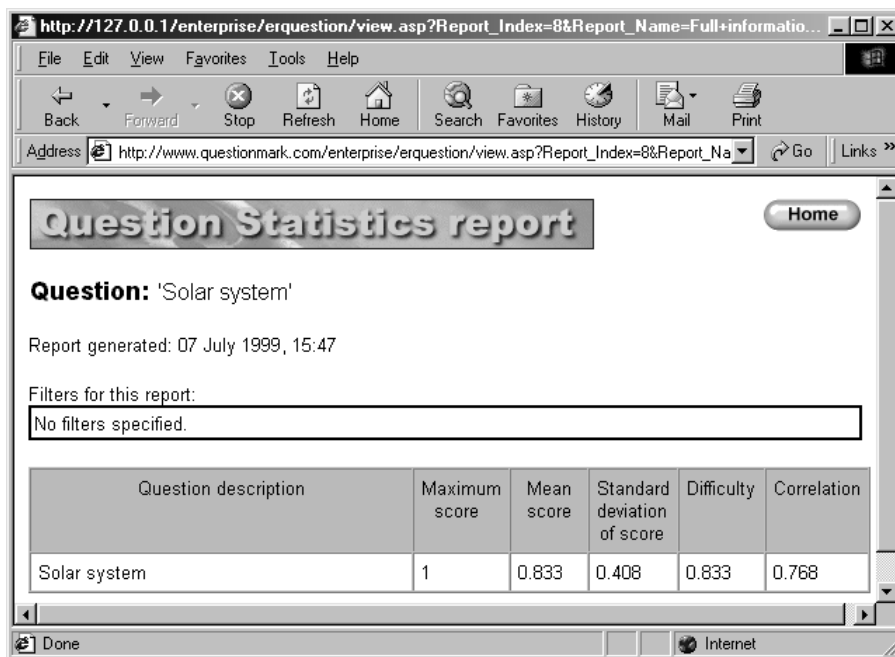
This defines the default filters for the report. See section 5.7 for more on filters.

## 5.5 Question Statistics reports

### 5.5.1 Example Question Statistics report

A **Question Statistics report** analyzes questions and gives you information on the scores and outcomes of the question, including calculating the correlation of the question's score with the session score, and other useful statistics.

Here is an example fragment of the report showing results on a single question:



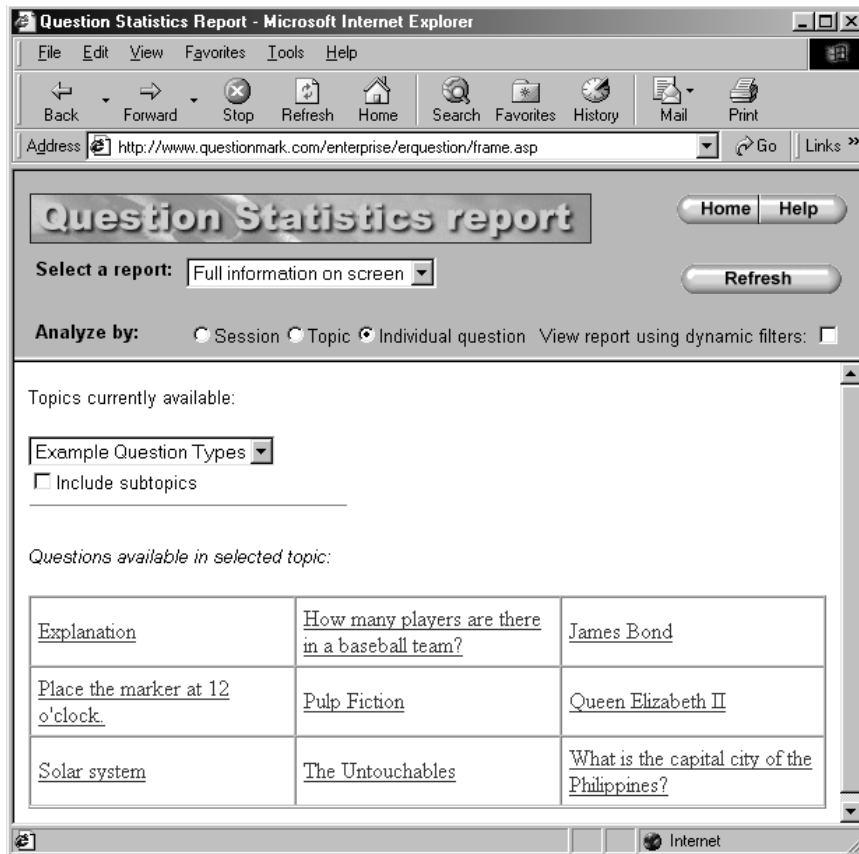
The screenshot shows a web browser window with the following content:

- Address bar: [http://127.0.0.1/enterprise/erquestion/view.asp?Report\\_Index=8&Report\\_Name=Full+informatio...](http://127.0.0.1/enterprise/erquestion/view.asp?Report_Index=8&Report_Name=Full+informatio...)
- Browser menu: File, Edit, View, Favorites, Tools, Help
- Navigation buttons: Back, Forward, Stop, Refresh, Home, Search, Favorites, History, Mail, Print
- Address bar: [http://www.questionmark.com/enterprise/erquestion/view.asp?Report\\_Index=8&Report\\_Na](http://www.questionmark.com/enterprise/erquestion/view.asp?Report_Index=8&Report_Na)
- Page title: **Question Statistics report**
- Question: 'Solar system'
- Report generated: 07 July 1999, 15:47
- Filters for this report: No filters specified.
- Table with 6 columns: Question description, Maximum score, Mean score, Standard deviation of score, Difficulty, Correlation
- Table data row: Solar system, 1, 0.833, 0.408, 0.833, 0.768
- Status bar: Done, Internet

## 5.5.2 Choosing to view a Question Statistics report

Here is how you view a **Question Statistics** report.

1. Click the **Run Report** button for this report type on the home page.
2. Choose the report definition you want to use from those available.
3. Choose whether to analyze by **Session**, **Topic** or **Individual question**.
4. Then choose which to report on.
  - If you analyze by **Session**, then you see a list of sessions and can choose one of them. You will see an analysis of all questions presented in that session. Any answers to the same question in a different session are filtered out.
  - If you analyze by **Topic**, then you see a list of topics and can choose one of them. You can choose whether to include questions in all the subtopics of the topic in your report. Your report will analyze all answers to the questions in the topic from any session.
  - If you analyze by **Individual question**, then you see a list of topics and individual questions in the topic. You choose one question, and the report will analyze all answers to that question.
5. Click the **View Report** button. Or if you are selecting by question, click on the question description.



You can also set some dynamic filters to control which answers are included in the report. See section 5.7 for more on filters.

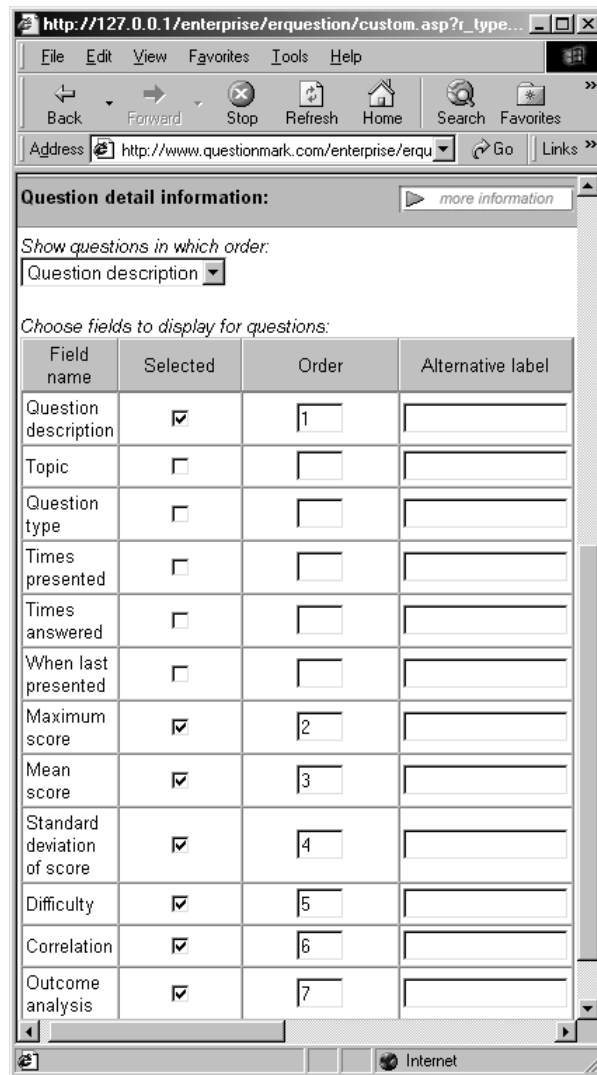
### 5.5.3 Question Statistics report contents

When creating a **Question Statistics report**, you define in the **Layout section** any header or footer text, whether the report is designed for screen view or printing and the font size. This works as for participant reports described in section 5.2.3. You also can define the number of records (questions) on each page of the report. In a large database, creating the report can take some time, and it can be worth putting a small number of questions, e.g. 10, on each page. You can also define **Filter information** – see section 5.7 for more on filtering.



The report contents itself are defined by a **Question detail information** section and an **Outcome analysis information**. The former describes information displayed for each question and the latter describes information displayed for each outcome within a question.

### Question detail information section



You can choose to order questions within the report by one of **Question ID**, **Question description** or **Topic**. And then you can choose to include some or all of these in the report. In the usual

way, you tick on the **Selected** column to include the information, and you can set the order and description by setting values in the **Order** and **Alternative label** columns.

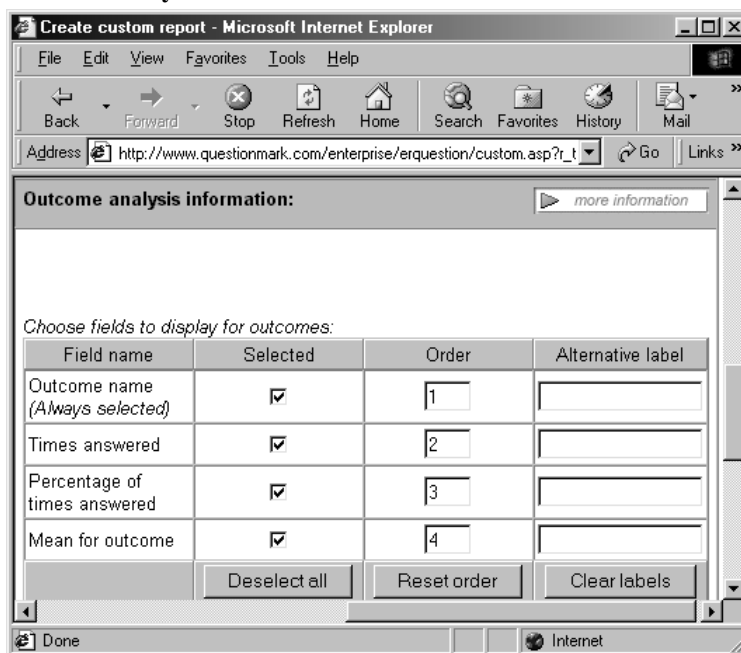
The meaning of the columns is as follows:

<b>Question description</b>	This is the description of the question, set in <i>Question Manager</i> .
<b>Topic</b>	This is the name of the topic the question is in, taken from the question database. If it's in a subtopic, a / character is used. For example <b>history/usa</b> means the <b>usa</b> subtopic of the <b>history</b> topic.
<b>Question type</b>	This is the type of the question, for example <b>Multiple Choice</b> .
<b>Times presented</b>	This is the number of times the question was presented, including times it was not answered.
<b>Times answered</b>	And this is the number of times it was answered. A question can be presented but not answered for a number of reasons including the following: <ul style="list-style-type: none"> <li>• Explanation questions can never be answered.</li> <li>• The participant submitted their answers to the block containing this question, but did not answer this question.</li> <li>• The participant abandoned the session without submitting their answers to the block containing this question.</li> <li>• The server sent the question to the participant, but the participant never received it – though this should be unlikely.</li> <li>• The time limit for the session expired, so the participant answers were not accepted.</li> </ul>
<b>Maximum score</b>	This is the highest score achieved by any participant who answered the question. Note that if no one has achieved the maximum possible score for the question, then this might differ from the maximum possible score.
<b>Mean score</b>	This is the average score in points for all the times the question was answered.
<b>Standard deviation of score</b>	This is the standard deviation of the scores of answered questions.  The standard deviation is a statistical measure of how the scores vary. With the same mean score, the higher the standard deviation, the more the scores vary.
<b>Difficulty</b>	This is the difficulty (or facility) of the question, which is the average score divided by the maximum achievable score for the question.  This is presented as a real number between 0 and 1. A difficulty of 0 means that the question is very hard (no one got it right). A difficulty of 1 means that the question is very easy (everyone got it

	<p>right). Difficulty can be a useful measure of how effective your questions are, but may not be meaningful if you only have a small number of results for the question.</p> <p>If you are assessing to differentiate between people, questions are most useful if they are neither too easy nor too hard. A very easy question is answered correctly by everyone, and so doesn't help you measure skills. Similarly a very hard question is answered correctly by very few people, and so adds little to your measurement. An ideal difficulty is 0.5, which roughly means that half your participants answered the question correctly. You may want to look carefully at questions with difficulty of more than 0.75 or less than 0.25, as these questions are less good at discriminating between people.</p> <p>On the other hand, if you are setting a test that you expect most people to get a high score in, then you will want to look at all questions that are difficult. For example, if you set a competency test that you expect almost everyone to get 90% or more in, then any question with a difficulty of 0.5 or less may be worth checking to see that it is not incorrectly set or ambiguous.</p>
<p><b>Correlation</b></p>	<p>The correlation (or discrimination) of a question measures how well the question correlates to the results of a session. It is only available if you choose the report to be on all questions in a session. Otherwise, it is not calculated, as it needs a session to be meaningful.</p> <p>It is presented as a number from -1.0 to +1.0. A high correlation (close to +1.0) means that the question is measuring the same thing as the session. A low correlation means that there is little relationship between participants getting the question right and getting a good score in the session. A negative correlation means that getting the question right correlates with getting a bad score for the session.</p> <p>As an example, if you have a test on mathematics, and include one question on football, you would expect that the correlation on the football question would be low. Participants who get it right are no more likely to do well at the other questions than participants who get it wrong.</p> <p>The correlation is calculated using a standard statistical method (the Pearson product-moment correlation). This method is not reliable if you use negative scores for questions, and you should be careful in</p>

	<p>interpreting correlations for such sessions.</p> <p>Questions with a correlation of less than 0.25, or with a negative correlation, indicate that participants who answer them well do not necessarily do well in the session as a whole. You may wish to consider removing them from the session, unless there are other reasons for having them in.</p> <p>If the difficulty is extreme (close to 0.0 or 1.0), then the correlation will not be that valid, and you should not consider its value as too important. You should in any case temper any interpretation with common sense. You may be doing things in a particular way for a particular purpose. But even if you are unhappy at looking at the absolute values, it will be interesting to compare the relative values – to look at the correlation of one question as compared to another.</p>
<b>Outcome analysis</b>	Tick this box if you want to display detailed information about each outcome.

### Outcome analysis information section



If you use **Outcome analysis**, then you can choose which of these four statistics to include for each outcome in the question. This makes a sub-table in the report for all the outcomes in the question.

<b>Outcome name</b>	This is the name or ID of the outcome, as set in <i>Question Manager</i> .
<b>Times answered</b>	This is the number of times the outcome occurs.  It's sometimes possible for a question to result in more than one outcome. So the number of outcomes answered can be more than the number of times the question is answered.
<b>Percentage of times answered</b>	This is the number of times the outcome has been answered divided by the total number of times the question has been answered, expressed as a percentage.
<b>Mean for outcome</b>	This looks at the session scores of the participants who chose this outcome, and gives the average percentage score. It's calculated even if the answers being analyzed come from different sessions.  For example if three participants choose an outcome and score 40%, 50% and 90% on the session, then the mean for outcome will be 60%.  This can be a very useful indicator – for example if people who choose an incorrect outcome score highly in the session, then the question might be poorly worded.

The usefulness of the outcome analysis depends on how you set up your outcomes when creating the question. You should make an outcome for each different possibility that you want to report on.

## 5.6 Survey reports

### 5.6.1 Example Survey report

A **Survey report** is a list of results, with one line in the report corresponding to one person taking a session. Unlike the **Question Statistics report**, which focuses on scores to questions, this report focuses more on the answers. Here is an example report fragment:

The screenshot shows a web browser window with the address bar containing the URL: `http://127.0.0.1/enterprise/ersurvey/view.asp?Report_Index=10&Re...`. The browser's menu bar includes File, Edit, View, Favorites, Tools, and Help. The address bar shows the full URL: `http://www.questionmark.com/enterprise/ersurvey/view.asp`. The main content area displays two survey questions:

**Queen Elizabeth II**

Topic:	Example Question Types	
Question type:	Text Match	1. Mountbatten 2. Windsor 3. Windsor 4. Mountbatten 5. Windsor
Mean score:	0.6	
Times presented:	5	
Times answered:	5	

**What is the capital city of the Philippines?**

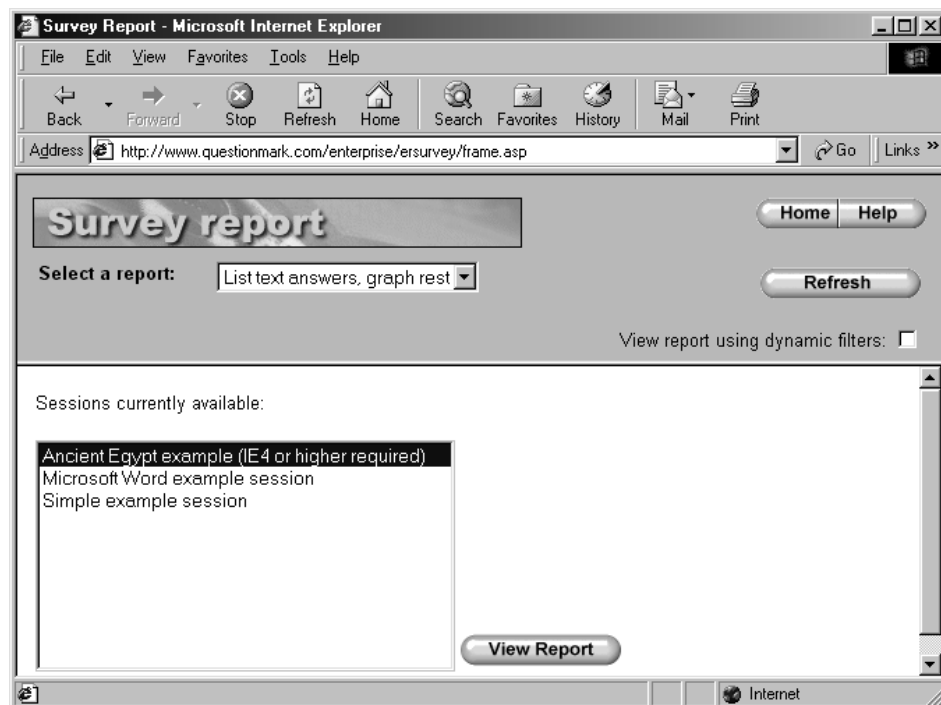
Topic:	Example Question Types	
Question type:	Multiple Choice	0 Manila 60%(3) 1 Jakarta 40%(2) 2 Seoul 10%(0) 3 Taipei 10%(0)
Mean score:	0.6	
Times presented:	5	
Times answered:	5	

The browser's status bar at the bottom shows "Done" and "Internet".

## 5.6.2 Choosing to view a Survey report

Here is how you view a **Survey report**.

1. Click the **Run Report** button for this report type on the home page.
2. Choose the report definition you want to use from those available.
3. You will see a list of available sessions. Choose one by clicking on it. The report will cover all answers to this session.
4. Click the **View Report** button.



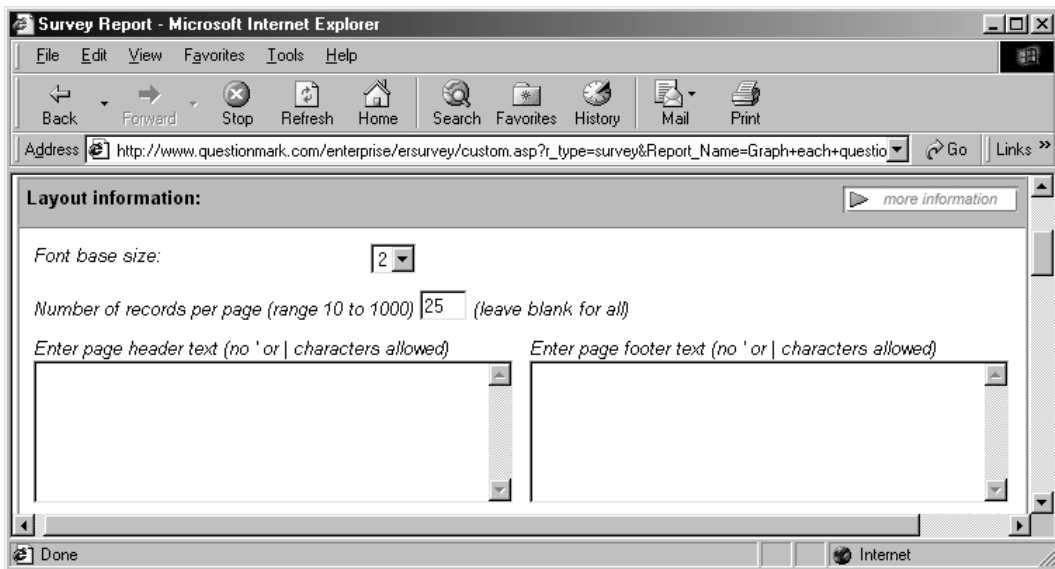
The **Survey report** includes all answered questions in the session. Questions are presented in the order they were first answered. You can also set some dynamic filters to control which answers are included in the report. See section 5.7 for more on filters.

## 5.6.3 Survey report contents

When defining a **Survey report**, you set data in six sections as follows:

### Layout information

This lets you define header and footer text, the font size and the number of records (questions) included on each page. These are defined the same way as in other reports.



The screenshot shows a web browser window titled "Survey Report - Microsoft Internet Explorer". The address bar contains the URL: [http://www.questionmark.com/enterprise/ersurvey/custom.asp?r\\_type=survey&Report\\_Name=Graph+each+questio](http://www.questionmark.com/enterprise/ersurvey/custom.asp?r_type=survey&Report_Name=Graph+each+questio). The main content area is titled "Layout information:" and includes a "more information" link. The form contains the following fields:

- Font base size: A dropdown menu with the value "2" selected.
- Number of records per page (range 10 to 1000): A text input field with the value "25" and the instruction "(leave blank for all)".
- Enter page header text (no ' or | characters allowed): A large text area.
- Enter page footer text (no ' or | characters allowed): A large text area.

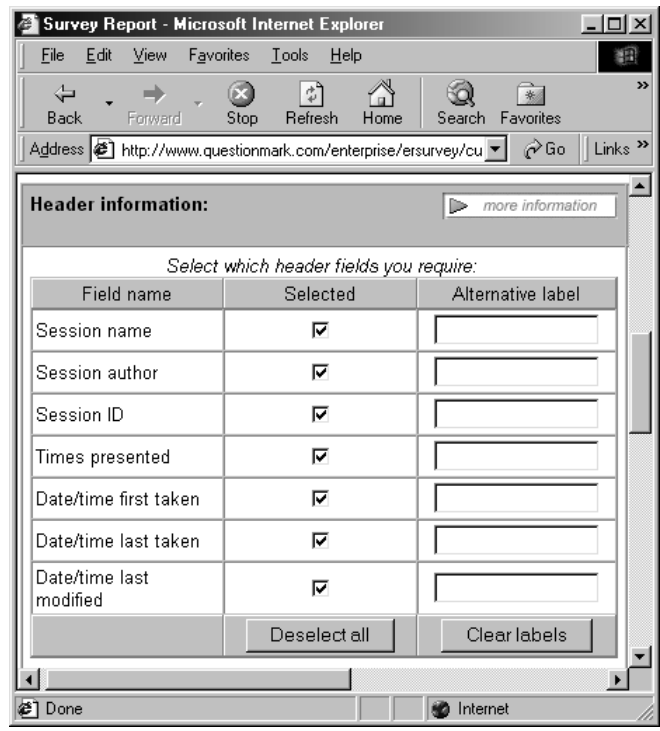
### Header information section

This part of the report definition allows you to define which information is shown at the top part of the report. You can select each field, and define alternative labels in the usual way.

<b>Session name</b>	This is the name of the session, as set in the session database by <i>Session Manager</i> .
<b>Session author</b>	This is the author of the session, the name set in the session database when the session was created in <i>Session Manager</i> .
<b>Session ID</b>	This is the unique numeric identifier for the session.
<b>Times presented</b>	This is the total number of times the session has been presented, ignoring any filters set in the report.
<b>Date/time first taken</b>	This is when the session was first taken, ignoring any filters set in the report.
<b>Date/time last taken</b>	And this is when it was most recently taken, also ignoring any filters set in the report.



<b>Date/time last modified</b>	This is the date and time when the session was last edited in <i>Session Manager</i> , prior to the first set of results being stored in the answer database.
--------------------------------	---



**Filter information section**

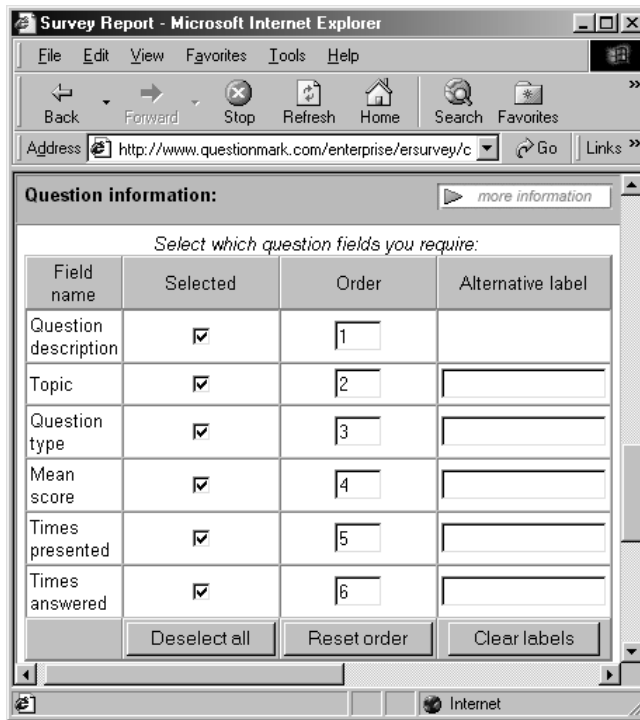
This allows you to set some standard filters that apply to the report. See section 5.7 for more on filtering.

**Question information section**

Each question is identified by its question description, and you can choose to show the following information by each question.

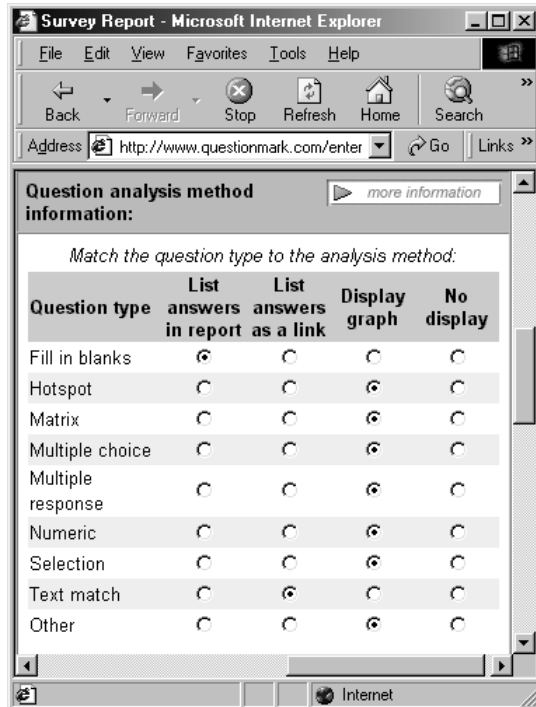
<b>Topic</b>	This is the topic the question is in.
<b>Question type</b>	This is the type of the question, e.g. Multiple Choice.
<b>Mean score</b>	This is the average or mean score for the times the question is answered.

<b>Times presented</b>	This is the number of times (subject to any filters) that the question was presented.
<b>Times answered</b>	This is the number of times (subject to any filters) that the question was answered.



### Question analysis method information section

This section allows you to define how the report shows answers from each question type. For example, you might wish to display a graph for some question types, but list the answers for other question types.



For each question type, you set one of the following:

- **List answers in report** means that you want the unprocessed answers to the question displayed in the report. This is especially appropriate for open-ended text questions, where it's not possible for the computer to assign the answers usefully to outcomes. Answers are listed in the order they were made, earliest first. Using this report option could lead to very large reports if many people have answered your questions.
- **List answers as a link** means that you want the text answers displayed, but you want this done on a separate page for each question. The main report displays a button, which if you click it shows the answers. If you have many responses or long responses, you may need to set this to avoid the report being too large.
- **Display graph** means display a frequency graph of outcomes. This requires you to set up the outcomes usefully when designing the question. Providing you do this, this is often the most useful thing to choose for most questions - in fact all that are not open-ended.
- **No display** means do not display answers to questions of this type in the report.

The standard question types each have their own line for you to set one of the above. The **Other** line refers to all other question types, for example those you define yourselves.

### Outcome information section

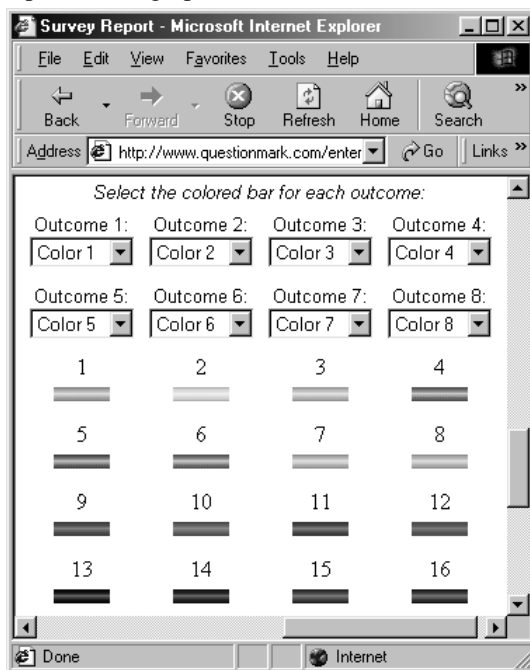
This section defines which colors are used in the graphs if you set **Display graph** above. Each outcome in the list is displayed in the color chosen.

Each outcome in a question is numbered according to its position in the question. Usually the top outcome as shown in the question editor or QML is outcome 1; the next outcome is 2; and so on. You can then assign different colors to different outcomes.

For example, if you divide your outcomes into positive, negative and neutral, you could:

1. Ensure that in all your questions, the positive outcome is first, the negative second and the neutral one last. You must do this before any answers to the questions are written to the answer database.
2. Match outcome 1 with green (color 3), match outcome 2 with red (color 4) and match outcome 3 with black (color 16).
3. Your report will then show positive as green, negative as red and neutral as black.

Sixteen graphics are shipped with *Enterprise Reporter* to make the graphs in survey reports, **bar1.gif** through **bar16.gif**. The colors of these are shown on the screen. Expert users can replace these graphics with their own to achieve alternative effects.



## 5.7 Filtering

### 5.7.1 How filtering works

Unless you specify filters, a report in *Enterprise Reporter* will cover all data in the answer database. Filtering allows you to select criteria so that the report just covers some of the data. You can set the following filters, choosing any combination that you need for your reports:

<b>Participant name</b>	<p>This restricts the report to only include results from one participant.</p> <p>You can type in any name in the filtering box. Participant names are not case sensitive. You leave the participant filtering box blank to not specify a filter.</p> <p>For example, you might make a <b>List report</b> that lists all results from a single participant. Or else, if you have many people using a standard login name, for example <b>Demo</b>, then you can filter on this to see an analysis of the results from this name.</p>
<b>Session name</b>	<p>This means that only results from this one session are included in the report.</p> <p>When setting up this filtering, you can choose from any session that has already been answered in the answer database. To not set a session filter, you select <b>ANY SESSION</b>.</p> <p>For example, if questions in a topic are presented in many sessions, you might choose to set a filter so that you can see a <b>Question Statistics report</b> for only the times they are answered in a particular session. Or in a <b>List report</b>, you might choose to see the results only for one session.</p>
<b>Group</b>	<p>Group filtering allows you to specify a group. The report will only include results from that group.</p> <p>You set this up by selecting from any group that has a set of results in the answer database. To not set a group filter, you select <b>ANY GROUP</b>. You can choose to filter on those participants that are not members of any group by choosing <b>NO GROUP</b>.</p>

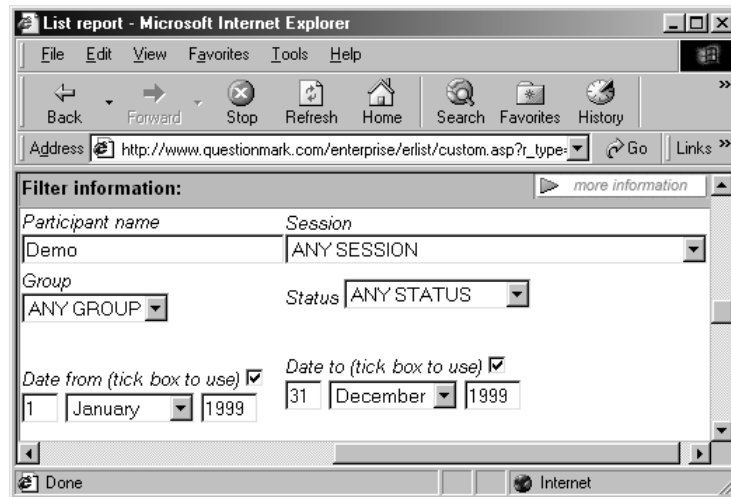
	<p>The group a result is in depends on how it was scheduled in <i>Security Manager</i>, or what the participant typed in if using <b>open.dll</b>. Results can not be assigned to any group, in which case they will not appear if using any group filter.</p> <p>Group filtering is particularly useful to limit results in a report to one group, department or class of participants.</p> <p>If you log on to <i>Enterprise Reporter</i> as a user without access to all groups, then a group filter will automatically apply to all your report selections.</p>
<b>Status</b>	<p>Status filtering lets you look at only a certain category of results. You can choose from:</p> <ul style="list-style-type: none"> <li>• <b>ANY STATUS</b> means do not filter by status.</li> <li>• <b>Recently Started</b> means that the session has started, but no answers have yet been received back.</li> <li>• <b>In Progress</b> means that the session has started, and answers from at least one block have been received. But the score band has not yet been calculated.</li> <li>• <b>Finished Normally</b> means that the session has concluded in the normal way.</li> <li>• <b>Timed Out</b> means that the session time limit was exceeded.</li> <li>• <b>ALL UNFINISHED</b> means either <b>Recently Started</b> or <b>In Progress</b>.</li> </ul> <p>For example, you might want to look at only completed sessions in your report.</p>
<b>Date from and Date to</b>	<p>Date filtering lets you set a start date and/or an end date. The report will only include results from sessions within this date range.</p> <p>If someone starts a session on one date and finishes it on another, the date used for filtering is the starting date.</p> <p>For example, if you want to compare this month's results with last month's results, then you can make the report in two different ways with two different date filters.</p>

Filtering does not apply to participant reports, which are only ever produced on a single set of results. Filtering on date, group and status is available in all report types except the **Participant report**. Filtering by participant is available in the **List**, **Question Statistics** and **Survey** report types. Filtering by session is only available in the **List** and **Question Statistics** reports – it is implied in the other report types already.

## 5.7.2 How to define filtering

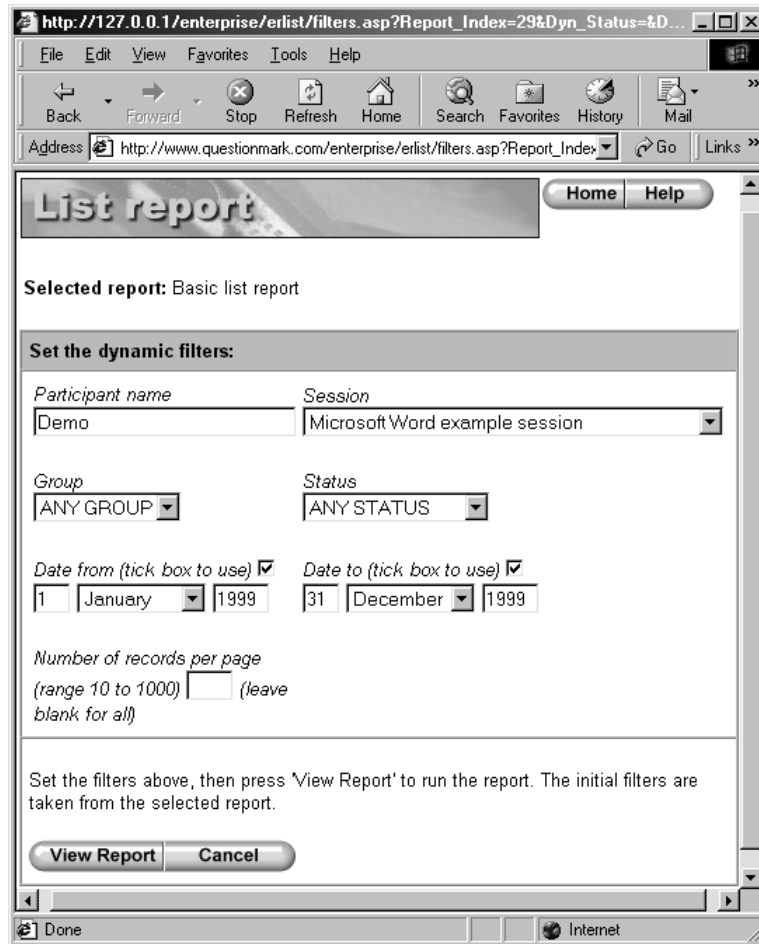
You can set filtering in two places, either when defining the report, or dynamically when viewing it.

When you define a report, you can set some static filters, which always apply when that report definition is used. This is useful if you want to do the same filtering each time the report is run. The screenshot below shows setting such filters up when making the **List report**. In the example, the report will show all results from the participant **Demo** in the calendar year 1999.



When you choose to run each report, you will see an option to **View report using dynamic filters**. If you select this, then when you click the **View Report** button, you are taken to a screen which allows you to set filters online. These filters apply only to the current instance of the report.

For example in the screenshot below, you can choose to modify the filters set statically. This change is not saved to the database, and only applies while you run the report.



When dynamic filtering, you can also set another value for **Number of records per page** in those reports that support this.

Whichever filters you set, they are always shown at the top of each report created with filters, so you have a record of which filters were used.



## 5.8 Letting participants see reports

### 5.8.1 Participant login

Participants can login to *Enterprise Reporter* and see a limited range of reports on their own performance in a session. They can only see participant reports, and only on their own results, not on any other participants. And they cannot choose a report definition type, but are limited to a single preset report definition.

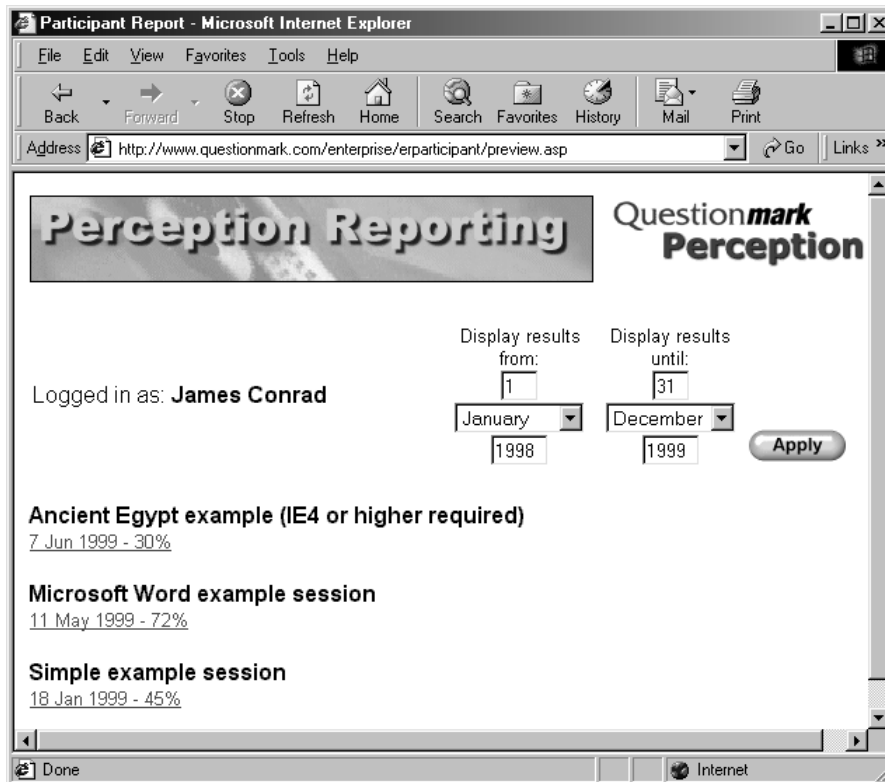
You can prevent participant login by altering the configuration file, **settings.asp** as described in section 2.2.3.

Participants login by pointing their browser to the **plogin.asp** program, for example by accessing the following URL: **http://www.xyzcompany.com/enterprise/plogin.asp**, where you should replace **www.xyzcompany.com** with your organization's URL.



They then see a login screen, into which they must type their name and password as defined in *Security Manager*. This works in a similar way to logging into **perception.dll**. Alternatively, there is an option in **settings.asp** to allow anyone to login as a participant without security.

After logging in, participants see a list of results from sessions they have taken. They can choose to look just at results within certain dates by entering dates at the top of the screen and clicking the **Apply** button.



Clicking on a result shows a **Participant report** on the set of results. Which participant report is used is defined in **settings.asp**, and can be freely changed. Only one report is possible for all participant login on the installation of *Perception Server*.

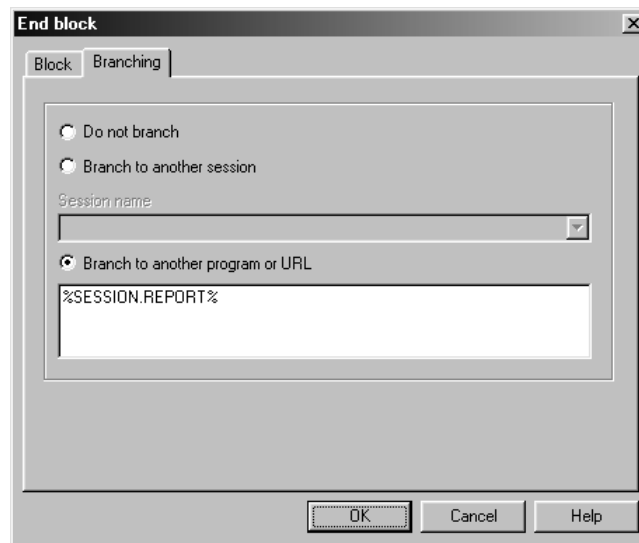
## 5.8.2 Calling a Participant report at the end of a session

As well as or instead of letting participants login to view reports, it's possible to arrange for a **Participant report** to be called securely at the end of a session. This means that after finishing a session, you can display to the participant a report.

This is a good way of presenting summary or diagnostic information to a participant at the end of a session. It supplements the information on the score band page with the full range of information in a **Participant report**, including the topic score breakdown.

You set up this report by using a special variable **%SESSION.REPORT%** in a session. Here is how to do it:

1. Ensure that you have not turned off this facility in **settings.asp**. If you want to prevent these reports being made, then you can set the variable **CF\_pend\_session** to **0**.
2. In *Session Manager*, edit each score band of a session you want to present a **Participant report** from. On the **Branching** tab, click on **Branch to another program or URL**, and enter the wording **%SESSION.REPORT%** as shown in the box below.



3. When a participant runs the session and reaches the score band, he/she will see a link, which if followed will lead to an *Enterprise Reporter* **Participant report**. This doesn't give access to any other part of *Enterprise Reporter*, just the single report.
4. Which report definition is to be used is set in **settings.asp**, but can be changed in the template file for the session if required. You change it by setting the **REPORT\_NAME** template variable to an alternative name. See the templates documentation online for more information on setting templates. You can also modify the text used to present the link to the report by setting something in the template.

The URL to *Enterprise Reporter* is found by using the variable **REPORT\_URL** from the format file. If you install *Perception Server* in a non-standard way, you may need to alter this variable in a template to make this facility work. See the templates documentation for more about this.

## 5.9 Advanced topics

### 5.9.1 Copying report data into spreadsheets

It's often useful to be able to copy report data into spreadsheet programs like *Microsoft Excel* so as to analyze them further. The easiest way to do this is to copy and paste them from your browser into the spreadsheet program.

How exactly this is done varies between different browsers and spreadsheet programs, but it usually works as follows:

1. Display the report in your browser. If using *Internet Explorer*, you can copy from the usual browser screen. If using Netscape *Communicator*, you should choose **View | Page Source** before copying.
2. Use your mouse to select the part of the report you want in the spreadsheet, or use **Ctrl+A** to select the entire report.
4. Use **Ctrl+C** to copy the report contents.
5. Paste the data into your spreadsheet using **Ctrl+V** or the **Paste** menu item.

With *Microsoft Excel '97*, this puts each report element in its own cell, and usually makes it very easy to manipulate the data in the spreadsheet program.

Because the answer database is an open *Access* database, it's possible to read the data directly from the database with some programs, instead of getting it from *Enterprise Reporter*.

### 5.9.2 Clearing the answer database

The answer database starts off blank, with no data in it. You can replace it at any time with a blank database, using the following procedure. This will of course remove all the results in the database. If using *SQL Server* or *Oracle*, see the Perception support site.

1. Check if possible that no sessions are in progress. You may also want to take the server software offline.
2. If it may have important data in it, make a backup copy of the existing answer database, usually file **qmpanswer.mdb** in directory **\perception\databases\**.
3. Use Windows *Explorer* to copy the file **qmpanswer.mdb** from the Perception CD in directory **\uncompressed\databases\** to replace this file.

4. Copying a file from CD usually copies it as read-only. It's important that the answer database is not read-only, so right-mouse click on it within Windows *Explorer*, click **Properties**, and set that it is not to be read-only.

*Enterprise Reporter* only lets you read and report on the answer database, it doesn't let you edit or remove results. If you want to delete or remove results from the answer database, you can do so using a results remover utility which is available on the Perception support site.

### 5.9.3 What happens if data changes

It's best if possible not to change questions and sessions after they have been answered. If you do this, then it's possible for report data to be misleading.

Suppose for example, you have a question which 100 people answer. And then you change it, and 100 more people answer it. Looking at the 200 sets of results could be misleading, if for example the changes make the question easier or harder. It would be better in this instance if possible to make the question a new one, rather than a change to an existing question.

Questions are identified by their question ID. *Enterprise Reporter* considers that questions are the same if they have the same ID, and different if they have different IDs. If you want to change the question so it has a different ID, here is a simple way of generating a question with a new ID in *Question Manager*:

1. Select it in *Question Manager*, and copy it to the clipboard.
2. Paste it into the same topic.
3. Edit the pasted question to make your changes.
4. If you want to, delete the old version of the question.

Similarly sessions are identified by their session ID. And you can make a session with a new ID, by making a new session by copying an old one from the *Session Wizard*.

If you do change sessions and questions after some results have already been entered into the database, *Enterprise Reporter* will do its best to make sense of the data. It will usually take the first instance of the data to be the one it uses.

If you make small changes to the data, for example correcting the text in question content, choices or feedback, this will not usually cause any problems in reporting. But if you adjust scores or add choices, this could cause issues that you would need to check carefully before relying on your data.

## 5.9.4 Changing text and graphics used in reports

Most of the text in reports, including almost all the labels and column headers, can be changed from the report definitions. This is usually done by defining an **Alternative label** to the standard report text.

If you want to change other text in reports, all text in *Enterprise Reporter* is defined in files with names starting with **TX** and with the **.asp** extension. It is possible to find text in these files that you want to change, and replace it with your own text. If you do this, backup your changed files, or make a note of all changes that you make, so that you can redo such changes if a subsequent release of *Enterprise Reporter* updates the files.

Each report type displays a standard graphic at the top of each page. Advanced users who want to change these graphics can do so by replacing the files with alternative graphics. For example each list report contains a graphic in the file **lrlogo.gif** at the top of each report. This file is found on the **enterprise\erlist** directory, and you could replace it with an alternate if you want to. You should make your alternate graphic the same height and width as the original.

If you want to display reports without the **Home** button in them, there are two ways you can do this:

1. For all reports, change **settings.asp** to set **CF\_Disable\_icons** to **1**. This will remove the **Home**, **Help** and **About** buttons from throughout *Enterprise Reporter*.
2. For a particular report, add the characters **&DI=1** to the end of a URL to view a report. For example if you are viewing a participant report and the browser shows the current URL as **www.xyz.com/enterprise/erparticipant/view.asp?iResult=1&Report\_Name=Demo**, then change this to:

**www.xyz.com/enterprise/erparticipant/view.asp?iResult=1&Report\_Name=Demo&DI=1**

You can do this for any of the report types.

## 5.9.5 Setting up Perception for local reporting

*Enterprise Reporter* is designed to be used for online reporting while sessions are being taken, but you can if you wish use it for reporting on a different PC to the webserver. You might want to do this to allow you to make reports on your personal PC at your leisure. Or if you archive the answer database from time to time, then you may want to use *Enterprise Reporter* to look at an old answer database, different to the one currently used on the webserver.

Setting up *Enterprise Reporter* locally is very simple. You should, however, check your license as to whether you are permitted to do this. Usually you will need to purchase a server

license for each separate copy of *Enterprise Reporter*. To set up *Enterprise Reporter* for local reporting:

1. You need a PC with a webserver with ASP and ODBC installed. If you don't have another webserver, on a desktop PC you may find it easiest to use the free *Personal Web Server* from Microsoft.
2. Install the reporting part of the server software.
  - The easiest way to do this is to run the **server.exe** program from the CD.
  - Or if you have it already installed on another computer, copy all the files in the **perceptionenterprise** directory including its subdirectories and the report database (**qmpenterprise.mdb** in directory **\perception\databases**). You will also need to register the **qmpcr.dll**.
3. Make the **enterprise** directory web sharable.
4. Make a copy of the answer database you want to report on, either to a new name in its existing directory, or to a new location.
5. Set up the ODBC **System DSN** for the answer and report databases.
6. Run a browser to access the *Enterprise Reporter* software via the local webserver. The URL usually will be **http://localhost/enterprise/login.asp**.





## 6 Advanced server issues

### 6.1 Perception Integration Protocol

*Perception Integration Protocol* (PIP) is a way of integrating Perception with other applications. PIP is a tool for programmers and system integrators, who want to link Perception with other applications.

This manual only gives a brief overview of using PIP. For full information on PIP, see the support website at:

<http://www.questionmark.com/perception/help/interface/index.html> (North American users)

<http://www.qnmark.com/perception/help/interface/index.html> (Other users)

PIP has two parts to it.

1. It allows sessions to be called securely from other applications. Instead of starting a session via **open.dll** or **perception.dll**, using PIP, you can start a session by making a direct call to **session.dll**, usually from another program.
2. It allows another program to be called at the end of a session, passing back information about the results of the session.

Applications of PIP can use one or both of these mechanisms. So:

- A program can use PIP just to start Perception sessions.

For example if you want to write your own secure front-end to Perception to control who can take which sessions to replace *Security Manager*, **open.dll** and **perception.dll**, you can do this. Once the session had been started, Perception would then run normally.

- Or a program can use PIP to control and track Perception sessions. It will call Perception to deliver a session and then get the results back.

For example, you might schedule someone to take a test in a course or training management system. The system would present the test via PIP, and the results would be stored in the management system's database as well as in Perception's database. Results could then be analyzed and tracked by the management system, and possibly further tests could be scheduled depending on the results of the first test.

- Or you can use PIP solely to call a program at the end of a session. Sessions are started using **open.dll** or **perception.dll** in the usual way, and the PIP mechanism is simply used to do something at session end.

For example, you could use PIP to send an email at the end of a session, or to copy results into an ASCII file or second answer database.

For ordinary users of Perception, the main benefit of PIP is that it allows you to link Perception with other programs. For example, any program that supports the AICC standard for web-based Computer Managed Instruction should be able to work with Perception very simply. Check the support site for a list of programs that work with Perception. At the time of writing this manual, the programs that can call Perception using PIP include Macromedia's *Pathware Attain*, Allen Communication's *Manager's Edge* and Siltan-Bookman's *Registrar*.

For system integrators, PIP offers a powerful and easy route to integrate Perception with other applications and to call other programs at the end of sessions. Full documentation on the range of options and ways of using PIP is on the support site.

For managers of *Perception Server* installations, here is what you need to know about PIP to control its use:

1. Every call to PIP needs a PIP file to be present on the server. PIP files are stored in a directory specified in **perception.ini**, usually **\perception\pip**. If for example, you wanted to prevent use of PIP, you could remove all files from this directory. And if some of your users are using PIP, they will need to be able to put files in this directory.
2. There are three ways of setting for a session to use PIP:
  - The usual way is to call **session.dll** with the **CALL=** parameter, usually passing the name of a PIP file.
  - If you just want to use PIP to call a program at session end, you can use *Session Manager* to make a setting in the branching part of a score band.
  - If you set in **perception.ini** that **Call PIP file=1**, then every session will use the default PIP file on the system. This should only be set by experienced users.

For further information, see the Perception support site.

## 6.2 Maximizing security

Every organization and webserver has its own security requirements, and it is not possible to cover all the ground in a document such as this.

Here are ten points to consider which address some common security issues for users of Perception. Please also check the Perception support site, as new information on security will be posted there.

### 1. Prevent unauthorized file access to the Perception directories

The various Perception files stored on the server are all unencrypted. Reading them could compromise security (e.g. the question database may hold the right answers to questions) and writing to them or deleting them could cause complete disruption to the assessment process.

- You should consider using the facilities of Windows NT to restrict access to these files to authorized people.
- Additionally or alternatively, you may want to consider restricting physical and network access to the webserver PC.
- Remember also to control access to back-up copies of data (e.g. back-up tapes made of the server disks).

### 2. Prevent unauthorized ODBC access to the Perception databases

The Perception databases are accessed as ODBC System DSNs. Depending on how your system is set up, it may be possible to read or write to the databases using ODBC, even if file access is not permitted.

- Putting the databases into *SQL Server* or *Oracle* format will allow you to protect the databases with a password.
- Any program running on the webserver could theoretically access them, so you should control what programs run on your webserver. (For example, letting a technically skilled student run programs on the webserver could permit them to read or write to the databases.)
- On some systems, it might be possible to access the databases from other PCs as well as the webserver itself. You need to set up your system security to control this.

### 3. Make sure your webserver only permits access to the right directories

Webservers are typically quite secure, and only permit access to directories that you have made accessible (web shared). However webservers vary in the way security works, and

you should make sure that you read the documentation that comes with your webserver, as to how to make it secure.

- Make sure that access to the Perception files is not permitted over the web, except to the three shareable directories. If for example, you made the databases directory shareable, then anyone would be able to download the entire question database from the web.
- Limit the sharing to the permissions specified in the installation chapter, as otherwise it's possible there could be security loopholes – for example do not give people read access to the **server** directory.

#### 4. Prevent unauthorized access to taking sessions

You may want to make sure that people can only take sessions if they have been properly scheduled to do so. For example, if you do not prevent it, participants may be able to see and take all sessions on the system using **open.dll**.

- You can disable **open.dll** by setting in **perception.ini** that **Enable open access to sessions** is **0**. This will prevent any access to **open.dll** and this, or else simply removing the **open.dll** file from the system guarantees that no one will be able to run it. To avoid possible mistakes or loopholes, this is the safest route to use.
- If you want to be able to run **open.dll** and **perception.dll** on the same system, you can set in *Session Manager*, that sessions are not to be presented in **open.dll**. Or you can set in **perception.ini** that **open.dll** will only work if the session ID is given on the command line. This should be safe in most situations, but is not as safe. If you are using both DLLs, you may want to disable the **Home** button to prevent possible confusion.
- If many people have access to *Security Manager*, you may also want to set a password to control scheduling each session. You set this password in the **Security** tab of the control information page in *Session Manager*. This will restrict scheduling the session to people who know the password, and will prevent, for example, a user in one department from scheduling another department's sessions.
- Make sure participant passwords are only communicated to the participant and are not easy to guess.
- If you want to prevent someone from taking a session more than once, you can use *Security Manager* to specify the number of times a participant may take a session.

**5. Use the Security Manager passwords and user system well**

*Perception Server* lets you assign names and passwords to people, and using this well will aid security. It's important to choose passwords that people will not guess, so use reasonably long passwords, and change them frequently.

- Change the starting password in *Security Manager* as soon as you install the software. Then change it from time to time.
- Use the facilities in *Security Manager* to give users of *Security Manager* and *Enterprise Reporter* only the capabilities they need. This means that if someone learns or guesses a password, they will only get limited capabilities.

**6. Be aware of possible loopholes with graphics and resources**

All the files in your **qmresources** directory and subdirectories are open to participants or others to access at any time. So even though the contents of questions and sessions can be very carefully guarded, it is much less easy to safeguard your graphics and multimedia files.

- It's often good practice to prevent browsing all directories on a webserver. But it's vital to do this with the **qmresources** directory and its subdirectories. You can check to see if this is set correctly on your webserver by typing in your browser the URL **<http://www.xyzcompany.com/qmresources/>**. You should not see a listing of files. If you do adjust your webserver settings.
- Even with browsing for a directory listing disabled, it is still possible for people to access these files outside sessions. They can do this if they know or can guess the file name. If a graphic contains the right answer to a question, or the question itself, this has implications for your assessment security. If you are concerned, you may want to consider only uploading graphics and other resources to the webserver just before a session is made available to participants, and removing them immediately afterwards.
- Although this can make it harder to manage graphics, it can be worth using names that are harder to guess. For example a file called **v124qx3.jpg** will be harder to guess than **mathsquestion.jpg**.

**7. Make it hard for people to find out or pass on the right answers**

As well as the protection measures above, there are other practical measures you can take to minimize the chance that participants find out the right answers to questions in advance of taking a test, or pass on the correct answers to each other.

- An excellent practice is to randomize which questions are selected in the session, so that each time someone takes a session, it is different. If you have enough questions in your bank to choose from then this means that most questions are different in two

attempts at a session. And so passing on the contents of a session to another person will not help them very much.

- Randomizing the order of the answers (e.g. in multiple choice questions) is also useful. It stops people saying that the right answer to question X is choice Y. It also prevents any possible author bias in where the right answer is.
- The right answers are not included in the HTML sent to the participant, so there is no way that a participant can find out the right answers from looking at what they receive. But take care about giving sessions with feedback in them, because the feedback contains information about the right answers.
- Reports can sometimes contain information about the correct answers to questions. Make sure you use the security facilities of *Enterprise Reporter*, and take care about what happens to reports after they are produced.

#### **8. Do what you can to stop people passing questions to each other**

Because browsers are designed as general-purpose tools, there is a danger that participants can print out questions and answers, or copy and paste them to other applications. This is a potential problem with any computer delivered test, but particularly so with web tests, because the participant may be remote and the way browsers work makes it easy to do.

- The only complete protection against this is to have a person monitor or watch the participant take the test. This is called proctoring or invigilation.
- Alternatively or additionally if you have control over the PC on which tests are delivered, you may be able to set it up with a specially configured browser that stops people from printing and doing other things you don't want. See the Perception support site for more information on this.
- You can make a setting in **login.format** which will open all sessions in a "kiosk" style window without the usual menu commands. This isn't totally secure, i.e. there are ways round it, but it can be a useful security measure.
- If you put questions in many short blocks, or use question by question delivery, this can make it harder to print all the questions. The most vulnerable situation is when you put many questions in a block without any protection on the browser; it's then extremely easy to print them all.
- Although you may be able to minimize this risk, once a participant has seen a question, you should recognize that its secrecy may be compromised.

**9. Think laterally about possible security loopholes**

It doesn't help much having strong security in one place and weaker security in another place. Think around all the issues in your system and organization and try to find any weak places. Here are a few things to consider:

- Look at the security of files on machines being used by authors. If a secure exam is being prepared, make sure that people cannot access it during preparation on the author's machine (or in his/her back-up media).
- If your server permits FTP, check that the Perception files are not included in the FTP area; or if they are that appropriate security is maintained on them.
- As well as trying to prevent security breaches, you should also keep an eye on the server to detect them after they have happened. One obvious thing to check for is an attempt to find out passwords by repeatedly trying all possibilities. Check the *Perception Server* logs regularly to see if any unusual activity is happening. Look at the answer database with *Enterprise Reporter* to see that it makes sense. And if your webserver has any logging mechanism, you may wish to use this as well to check for any security breaches.
- Most breaches of computer security occur because of loopholes outside of the computer system; people write their passwords down, for instance, or use simple passwords that can easily be guessed, such as their partner's name. One of the most lucrative techniques used by computer hackers is to look through rubbish bins for pieces of paper, which give essential information such as user names and passwords.

**10. Some extra things you can do if you are very concerned**

If security is very important to you, then you could consider some of the following measures:

- You could use Windows NT passwords for some of the Perception DLLs as well as their own passwords. For example, you could protect **security.dll** with both its own password and a Windows NT one.
- Change the server key in your **perception.ini** to a different value. This key is used in generating numeric ids within the system. Changing it minimizes the possibility of someone cracking the system, and accessing sessions they should not.
- If you are concerned about security, you may want to disable the about command by changing **perception.ini**. Please only do this once you have the system working satisfactorily.

- Using *SQL Server* or *Oracle* will give you a higher level of security than keeping the databases in *Access* format.

One other related issue is how to check that a participant is the person he or she claims to be. By setting up a defined list of participants with names and passwords, you prevent someone from impersonating someone else without their permission. In order to login as someone, you need to know his or her password.

But this does not stop someone from giving their password to someone else, or someone getting a friend to sit by them at the computer helping them answer the questions. For a high stakes exam, the only completely reliable way is to proctor or invigilate the session. Have someone physically check that the person is who they say they are, and that they log in on the correct computer with the correct name. If you are testing remotely, a common practice is to let local managers or training officers have the passwords for entry and get them to log the person in. That way you can be reasonably sure of identity.

If you are testing over the web and you can't proctor, then there are still lots of useful measures you can take.

- Firstly use the facilities of *Security Manager* to set names and passwords for all participants. Give each participant a unique password to log onto Perception with. This is the most valuable measure and will mean that no-one can impersonate someone else without their permission.
- Then consider setting a time and date limit within which an important test can be taken. For example, in an educational setting, if all students in a class must take a test at the same time, there is less scope for an able person to assist or impersonate a less able one, as both must do the test simultaneously.
- Remember that Perception stores in the answer database and makes available in reports the IP address, and, usually, the host name of the computer used by the participant. The log file **perception.log** also records IP addresses. Participants may not realize that they can be tracked by these means, and scanning these looking for anomalies or duplicates can be valuable, if you know where participants are supposed to be.
- Lastly, you can try inserting some questions of a personal nature part-way through a test. Ask for the participant's mother's maiden name or some other piece of personal data that only the real participant is likely to know.



## 6.3 Obtaining participant name from Windows NT

If you use Windows NT security for storing and validating people's names, it is possible to link *Perception Server* up with this list of names. This means that if people are already logged into Windows NT, they may not need to re-login to Perception.

You do this by passing the **?NTUSER** parameter to **open.dll** or **perception.dll**. If you pass this parameter, then the software ignores any **NAME=** parameter, and will never ask the participant their name in the usual way. Instead, the software will try to get the name by the browser communicating with Windows NT. If the name cannot be obtained from Windows NT, then access will be refused.

In order for this to work, you will need to set up both the webserver and the participant browser to authenticate in this way. For the highest level of security, you will want to set up your webserver to use only **Challenge/Response** security for the **qm** directory. You should not permit **Allow Anonymous Access**. If participants are using the *Internet Explorer* browser on Windows 95, 98 or NT PCs, and if this is set to automatic logon the browser will automatically send the user's login name. If they are using Netscape browsers or non-Windows computers, the browser will display a dialog box asking them for their name and password before continuing.

Some further information on how to set this up is on the Perception support site.

### Using ?NTUSER with perception.dll

For fully secure use, you should use the **?NTUSER** parameter with **perception.dll**. You need to follow these steps:

1. You must input all the participant names into the security database, either by importing them or typing them in. It doesn't matter what passwords are set in the security database, as these will be ignored.
2. You should schedule participants to sessions in the usual way.
3. When **perception.dll** is called with the **?NTUSER** parameter, **perception.dll** will check with Windows NT for the name of the currently logged in user. And then will check to see if this name is in the security database.
4. If the name is found and confirmed, the list of sessions scheduled to the participant will be offered in the usual way.

Although this does require you to enter participant names into the security database in advance, this does provide a way of securely identifying participants using the Windows NT login procedure. It also prevents participant passwords from being transmitted or stored without encryption.

### Using ?NTUSER with open.dll

If you are using **open.dll**, this facility will only work if you have purchased a license for an unlimited number of participants. Otherwise you cannot use the **?NTUSER** parameter with **open.dll**.

If you do use it, then **open.dll** will read the name from Windows NT and use this as the participant name. Note that there is a security loophole in this process. Someone could call **open.dll** without the **?NTUSER** parameter and type in whatever name they want. So using **?NTUSER** with **open.dll** is more of a convenience than a security aid.

Thus if all your users are already authenticated with Windows NT, then you can pass this information into **open.dll** using the **?NTUSER** parameter, for example by calling **http://www.xyzcompany.com/open.dll?NTUSER**. But an unscrupulous user could call **open.dll** without the parameter and enter a different name.

## 6.4 Maximizing the speed of Perception Server

*Perception Server*, like other programs that run on a webserver, puts much more load on the webserver than the simple operations of delivering HTML pages. Here are ten areas to look at in optimizing the performance of *Perception Server*, particularly if you have many participants taking sessions at the same time.

### 1. Make sure your webserver is fast and well specified

Obviously, the faster your webserver is, the faster it will run *Perception Server*.

- Because of rapid changes in hardware, it's not practical in this manual to specify a recommended configuration, but you are recommended to have a fast processor, lots of RAM memory and a fast disk drive. Adding extra memory is probably the most effective improvement that can be made to an existing machine.
- Although some users of *Perception Server* do manage to use it with *Personal Web Server* under Windows 95 or 98, you will get much better speed and robustness with Windows NT Server running IIS 4.
- As with any web application, speed problems can be due to connection issues between server and client. The fastest server in the world is still limited by bandwidth and network bottlenecks.
- You may be able to adjust or "tune" your installation of IIS or your use of ODBC to maximize performance. Check the *Perception* support site and your webserver documentation for more information about this.

## 2. Set up Perception well

There are several **perception.ini** settings that have an effect on speed.

- Make sure that **Keep database open** is set to 1. Failing to set this will seriously slow database access.
- Make sure that **Disable statistics calculation** is set to 1. Failing to set this will slow down writing answers.
- You should usually set **Preprocess formats** to 1. This will slightly slow down the first access to a block after **session.dll** is initialized, but will speed up subsequent accesses.
- Review the setting for **Template cache size**. You can see when templates are loaded and unloaded from the cache by checking the log file; if this happens very frequently, increase the number of templates cached.
- Make sure the log file doesn't get too large, by periodically archiving it or deleting it. In extreme cases, you can turn off logging by setting **Disable logging** to 1, but this is not usually recommended as the log file is a useful resource.
- Check that database logging in **perception.ini** is not on. And check that ODBC tracing in *Control Panel* is not turned on. Either of these will slow down the server.
- Although this may well not be possible for you, consider turning off full answer saving, by setting only summary answer data to be saved in *Session Manager*. This will store only results for each session, not for each question, so may not be appropriate for all users. But it will speed up the system.

## 3. If possible, stagger your start and end times for sessions

Unlike a conventional desktop testing system, the webserver only processes material at the start and end of each block. This means that if large numbers of people start a test at the same time, this will put a large load on the server, which may give a slow response time. While a block of questions is being answered all the work is performed on the participant computer, and the server is idle. Then when the block is being marked, all the work is done by the server, and the participant computer is waiting.

So to maximize the throughput of the system, and to minimize the load on the webserver, you would ideally want people to take sessions at varying or random times. Everyone starting a test at the same time puts the maximum load on the webserver. There are often good security or pedagogical reasons for wanting to start sessions at the same time, but if it is possible to arrange staggered or variably timed access to sessions, you will get better performance. You may find that even a slight delay will help – for example if you have

lots of students in a room taking a test at the same time, have someone walk round the room tapping people on the shoulder to let them start.

#### **4. Consider carefully the number of questions in your blocks**

It's important to get a good balance in the number of questions in your blocks. Too many questions in a block can slow things down, as can too few.

The more questions there are in a block, the longer it takes the software to construct the block to send to the participant, and the longer it takes to mark and process the answers when they come back. Particularly if a lot of people are starting a session at the same time, having a big block to start with can slow things down.

Suppose for example, 100 people are taking an hour long session with 100 questions in it, all starting about the same time. And suppose on this server, with these questions, it takes two seconds for the webserver to prepare a block of 100 questions and half a second to prepare a block of 10 questions.

- If your session has a single block of 100 questions, then it will take a total of 200 seconds to make the blocks for all 100 participants. And some participants will wait 3 minutes to get their first questions. The webserver will then stand idle until the end of the test, when it will get very busy again.
- If your session is divided into 10 blocks, each of 10 questions, then it will take only 50 seconds to make the first blocks, which means a much smaller waiting time for participants. Participants will tend to move at a different speed through the remaining blocks, meaning much less congestion on the webserver.
- On the other hand, if you had 100 blocks each of one question, you might put a heavier load on the webserver, because there is an overhead to create and mark each block. And so the webserver could become very busy, dealing with a total of 10,000 small interactions in the hour.

There may be educational or other operational reasons that dictate the number of questions in each block. But if you have the option to change this, then you might increase your webserver's productivity by changing this. Look particularly at sessions that have more than 50 questions in a block or that have many blocks with less than 10 questions in a block.

#### **5. Consider using SQL Server or Oracle**

In general, putting the databases into *SQL Server* or *Oracle* format should give improved speed and robustness, especially if you have a large number of simultaneous connections. *Access* is not designed for a very large number of simultaneous connections, and if you have more than 50 participants taking sessions at one time, you should seriously consider moving to a more robust database engine.

**6. Answer the session once before your participants get to it**

The first time answers to each session and each question are recorded in the answer database takes a bit longer, as the initial details must be recorded. If you are going to have a lot of people start a session at the same time, take the session yourself once first (or perhaps more than once if questions are selected at random) to register the questions and session in the answer database.

This will also initialize the DLLs if necessary and load the templates into memory.

**7. Compact the databases**

If you are using the standard *Access* databases, then you may find that compacting the Perception databases will improve performance. You need to do the compacting before uploading the databases to the webserver, or while **session.dll** is offline - DO NOT compact the databases while sessions are in progress. You can compact the question database from *Question Manager* and the session database from *Session Manager*. To compact the answer and security databases, you need to use *Access 97*.

**8. Review your use of graphics and multimedia**

Large graphics or multimedia files will obviously take time to download. And having a lot of graphics, for example a graphic in each choice in a question, can also put a big load on the webserver.

It can sometimes be worth putting graphics or multimedia files on a different server to speed up their delivery.

**9. Check what else is running on the webserver**

Other programs or services running on the webserver will impact on how fast Perception can run. Ensure that no superfluous programs are running on the webserver, including screen savers - a webserver should never have a screen saver program running. Running *Enterprise Reporter* at the same time as participants take sessions will also impact performance. Producing reports can use up a lot of processor power.

**10. Consider multiple servers**

If all else fails, you may want to look at having more than one server delivering Perception sessions, either writing to the same databases, or else with different databases.

Please check the support site for further and more up-to-date information on optimizing performance.

## 6.5 Server variables

Server variables are best illustrated with an example.

Supposing you want to mention someone's name in a session, for example in the introduction wording to a session. Obviously, the name is different each time the session is run, so you can't put it in the wording itself. What you can do is put in the special code `%SESSION.USER%`, which is interpreted as the participant name.

So if for example, you put in your introduction wording:

**Welcome %SESSION.USER%. This test is about genetics.**

Then if a participant logs in to the session as participant **John Doe**, then they will see the wording:

**Welcome John Doe. This test is about genetics.**

Server variables are used within the template and format system for controlling the way questions and sessions are generated. And they can also be used anywhere within your questions or sessions, where you want Perception to display the current value for something. They are also useful when calling other programs from Perception.

All server variables start and end with the % character. The server software considers all words that begin and end with % without spaces in between as being possible server variables. If you want to use some text like this, without it being considered a server variable, use `%%`. For example, if you put this in a question:

**What does the %%SESSION.USER%% variable mean?**

Then it will be displayed as:

**What does the %SESSION.USER% variable mean?**

These variables apply throughout any session:

<code>%SESSION.TITLE%</code>	The name of the session as defined in the session database.
<code>%SESSION.ID%</code>	The unique numeric ID identifying this session in the session database.
<code>%SESSION.URL%</code>	The URL of the server software being used, e.g. <b><code>www.xyzcompany.com/qm/session.dll</code></b>
<code>%SESSION.TIME%</code>	The current time.
<code>%SESSION.DATE%</code>	The current date.

<b>%SESSION.USER%</b>	The current participant name, or blank if not set.
<b>%SESSION.USERDETAILS%</b>	This returns the content of the participant details field from the security database. If this field is empty or if the participant entered via <b>open.dll</b> a blank string will be returned.
<b>%SESSION.GROUP%</b>	The group of the current participant, or blank if not set.
<b>%SESSION.RESULT%</b>	This is the index of the current session's results in the result table of the answer database. It's useful if you are calling another program, and want that program to be able to access full details of the session in the answer database. If answers are not being saved, this is zero.
<b>%SESSION.ACCESS%</b>	This returns the current access ID being used in a session, which is an internal number used to identify the session in progress.
<b>%SESSION.OF%</b>	The total number of blocks in the session.
<b>%SESSION.SCORE%</b>	The score for the session so far.
<b>%SESSION.MAX%</b>	The maximum score for the session so far.
<b>%SESSION.PERCENT%</b>	<b>%SESSION.SCORE%</b> as a percentage of <b>%SESSION.MAX%</b> . If the maximum score is zero, it too is zero.
<b>%SESSION.TIMED%</b>	This returns <b>%TIMED%</b> if the session has a time limit and <b>%NOTTIMED%</b> if it doesn't. Its main purpose is to be used within the format files.
<b>%SESSION.TIMELIMIT%</b>	The time limit of the session in seconds, or 0 if no limit.
<b>%SESSION.TIMEELAPSED%</b>	This is the number of seconds elapsed since the session started, as the server measures it.
<b>%SESSION.TIMELEFT%</b>	This is the number of seconds left until the time limit is reached, as the server measures it. If there is no time limit, it is -1.
<b>%SERVER.GRAPHICS%</b>	The web path to the server subdirectory in which graphics are found.
<b>%SERVER.VERSION%</b>	Version and build number of the server software.
<b>%SERVER.DATE%</b>	Date the server software was created (by Question Mark).

These apply only when displaying the score band at the end of the session.

<b>%SESSION.SCOREBAND%</b>	This gives the name of the score band that applies at the end of the session.
<b>%SESSION.REPORT%</b>	This gives the URL to display a participant report, as described in section 5.8.2.

And these apply to the current block within the session.

<b>%SESSION.NUMBER%</b>	The number of the current block within the session.
<b>%SESSION.INTRO%</b>	In a block, the introductory text. In a score band, the final page text.
<b>%SESSION.FEEDBACK%</b>	The entire feedback currently being presented.
<b>%BLOCK.TITLE%</b>	The name of the current block.
<b>%BLOCK.SCORE%</b>	The score for the current block.
<b>%BLOCK.MAX%</b>	The maximum score for the current block.
<b>%BLOCK.PERCENT%</b>	<b>%BLOCK.SCORE%</b> as a percentage of <b>%BLOCK.MAX%</b> .
<b>%BLOCK.INTRO%</b>	The introductory text for the current block.
<b>%BLOCK.NUMQUESTIONS%</b>	This is the number of questions in a block. Explanation questions are not counted.
<b>%BLOCK.NUMBER%</b>	This is the number of the current block in the session.

These objects relate to the current question.

<b>%QUESTION.DESCRPTION%</b>	The question description for this question.
<b>%QUESTION.TOPIC%</b>	The topic for this question.
<b>%QUESTION.ID%</b>	The unique numeric ID for this question.
<b>%QUESTION.QTYPE%</b>	The QML question type of the question, e.g. MR or MC.
<b>%QUESTION.NUMBER%</b>	This is the number of the question in the current block, as the participant sees it. Explanation questions are not counted.
<b>%QUESTION.ANSWERED%</b>	Useful when delivering feedback. If this question has been answered then it will yield <b>%ANSWERED%</b> , otherwise will yield <b>%NOTANSWERED%</b> .
<b>%QUESTION.SCORE%</b>	The score for the question.
<b>%QUESTION.MAX%</b>	The maximum possible score for the question.
<b>%QUESTION.OUTCOME%</b>	Feedback text for the outcome(s) that have been reached.



<b>%QUESTION.RESPONSE%</b>	This gives the text (truncated) of any answers given to the question.
<b>%CONTENT.URL%</b>	The URL of the current content element (e.g. when using graphics, this will be the file name).
<b>%CONTENT.ATTRIBUTES%</b>	This is for use with image and multimedia content. It returns a formatted string of attributes in the form <code>ATTRIBUTE="VALUE"</code> , e.g. <code>WIDTH="100" ALT="Picture"</code> .
<b>%CHOICE.ID%</b>	The number of the current choice.
<b>%CHOICE.VALUE%</b>	If answered, the value of the answer. If this is a text choice, this will yield any text the participant entered.
<b>%CHOICE.WIDTH%</b>	The width of the choice if defined, else the definition of <b>%WIDTH%</b> .
<b>%CHOICE.HEIGHT%</b>	The height of the choice if defined, else the definition of <b>%HEIGHT%</b> .
<b>%CHOICE.WIDTH%</b>	The maximum number of characters of the choice if defined, else the definition of <b>%MAXCHARS%</b> .
<b>%CHOICE.CHECKED%</b>	If a choice (e.g. a radio button) was chosen this will yield <b>%CHECKED%</b> . If not chosen, it will yield <b>%NOTCHECKED%</b> .
<b>%CHOICE.SELECTED%</b>	If a choice (e.g. a selection option) was chosen this will yield <b>%SELECTED%</b> . If not selected, it will yield <b>%NOTSELECTED%</b> .

The **QUESTION**, **ANSWER** and **CHOICE** objects can also be called to return the value of any attribute defined in the QML for the question simply by using the name of the attribute as the object property. For example, if a question in QML starts:

```
<QUESTION DESCRIPTION="This is a question" TOPIC="My First Topic" ID="7199072612921553"
TYPE="text/plain" TEMPERATURE="Very Hot">
```

a call to **%QUESTION.TEMPERATURE%** will give the value **Very Hot**.

This facility is there to enable advanced users to incorporate information from other systems or to extend existing question types or create new ones.

## 6.6 The Perception log file format

The server DLLs create and write to log and error files to record what they have done. These files are ASCII files that can be read with *Notepad* or any text editor, and contain very valuable information for troubleshooting.

Whenever the server DLLs find a problem, they write full details of the error in an error file, called **perception.err**. This information is often more detailed than the information displayed

on screen, and if you have a problem that puzzles you, you should always check the error file. This file is stored in the directory `\perception\logs\`, or else in whatever directory is defined as the **Error directory** in `perception.ini`.

And whenever the server DLLs do anything significant, they write a line of details in a log file, called **perception.log**. This file is stored in the directory `\perception\logs\`, or else in whatever directory is defined as the **Log directory** in `perception.ini`. The log file can be useful for working out what has happened, or for an audit trail.

The log file records each time:

- A DLL is reset, or an attempt is made to reset but is not permitted.
- The server is taken offline or online or an attempt is made to do this but is not permitted.
- Someone tries to enter *Security Manager*, with the name they use.
- Someone starts a session, with the participant name, group name and session name.
- Someone submits a block, with the participant name, group name, session name, block number and block score. If the session has expired, this is noted in the log file.
- Someone finishes a session, with the participant name, group name, session name and score for the session.
- A PIP call is allowed, or a PIP call is denied for any reason.
- A template file is loaded for the first time, or when it is unloaded from the cache because the cache is full.

All log file entries start with the date and time, including time in milliseconds, and the IP address of the participant. For example, if someone takes a two-block session, the log entries might look like the following:

**04/04/1999 14:00:03:0550 & 194.159.30.136 & John Doe & Ohio & Started Safety Quiz progress ID 2326529845400298**

**04/04/1999 14:02:32:1460 & 194.159.30.136 & John Doe & Ohio & Completed Started Safety Quiz block 1 score 5 out of 8**

**04/04/1999 14:04:56:5670 & 194.159.30.136 & John Doe & Ohio & Completed Started Safety Quiz block 2 score 7 out of 8**

**04/04/1999 14:04:58:8912 & 194.159.30.136 & John Doe & Ohio & Safety Quiz score 12 out of 16**

This shows that **John Doe** of group **Ohio** started the **Safety Quiz**, scored **5** points in the first block and **7** points in the second block, scoring a total score of **12** out of **16**. The time, date and IP address are also shown.

If you set **Log database activity** to **1** in **perception.ini**, then the log file will also contain information about all database activity, but this is only recommended if there is a problem in this area.

If using PIP, you can set **Log PIP data** to **1** in **perception.ini**. This will log all PIP activity, and make a separate log file **pip.log** containing this.

Log files can get quite large. You can delete the log file at any time, and the server software will start a new one. You may wish to archive log files at regular intervals, e.g. once a week or once a month, and keep old log files in case there is a query about use of the server that you need to resolve.



# Appendix A

## Answer database format

### A.1 Overview

The supplied answer database (**qmpanswer.mdb**) is in Microsoft *Access* '97 format, but since the server software only accesses it via ODBC, you can convert it to a different format if you need to.

The database contains five tables:

- The results table (**A\_Result**) is the master table containing one record for each time a person takes a session. Whenever someone does a test or survey, a record is added to the result table saying who they are and what they did. If 10 participants take 2 sessions each, 20 records will be added to this table.
- The answer table (**A\_Answer**) contains one record for each time someone answers a question. If 10 participants take 2 sessions, each of 10 questions, 200 records will be added to this table.
- The session table (**A\_Session**) contains one record for each session that has been asked. If 10 participants take 2 sessions, there will be 2 records added to this table.
- The question table (**A\_Question**) contains a record for each question asked to any participant. If 10 participants take 2 sessions, each of 10 fixed questions, 20 records (one for each question) will be added to this table. But if one of the sessions selects its 10 questions from a bank of 50, then a record will be added to this table once for each unique question in the answers (up to 60 records in this example).
- The outcome table (**A\_Outcome**) contains one record for each possible outcome in a question (so it is a sub-table to the question table). A question can have any number of outcomes. If 10 participants take 2 sessions, each of 10 fixed questions, and each question has 3 outcomes, 30 records will be added to this table.

The answer database is updated at the start of every session, and after the participant submits each block. If only summary information is written to the answer file, then just tables **A\_Result** and **A\_Session** are filled. Tables **A\_Answer**, **A\_Question** and **A\_Outcome** are only written when full information is saved.

## A.2 Results table

Table **A\_Result** contains one record for each participant who has started a session.

Field Name	Description	Type
<b>Result_Index</b>	Index to result record within database.	Numeric
<b>When_Started</b>	When the session was started.	Date
<b>Session_Index</b>	Reference to table <b>A_Session</b> .	Numeric
<b>Session_Last_Modified</b>	When session in the session database was last modified. If the session changes between different times it is presented, you can see this by comparing this field across different records.	Date
<b>Password</b>	Not used.	Text 10
<b>Write_Answer_Data</b>	Whether <b>Full data</b> is saved for this set of results. If not, tables <b>A_Answer</b> , <b>A_Question</b> and <b>A_Outcome</b> are not filled in.	Yes/No
<b>Participant</b>	Participant name.	Text 50
<b>Member_Group</b>	The <b>Group</b> a participant is considered to be in for the purposes of this session. If session started by <b>open.dll</b> , this is whatever the participant entered in login. If session was started by <b>perception.dll</b> , this is whatever was set when the session was scheduled in <i>Security Manager</i> .  This field can be used in <i>Enterprise Reporter</i> to control which reports a user can see.	Text 50
<b>Participant_Details</b>	Participant details as set in the security database. This is blank if the session was called from <b>open.dll</b> .	Text 50
<b>Hostname</b>	The hostname of the user's computer, if known. If the hostname is not known, the IP address is copied to this field.	Text 50
<b>IP_Address</b>	IP address of participant's computer as advised to the server, when the session starts.	Text 20
<b>Signature</b>	Text "signature" code written so the validity of the record can be checked.	Text 20
<b>Still_Going</b>	Whether session still active.	Yes/No
<b>Status</b>	Status of session: <b>0 – Recently started.</b> First block sent but no replies received yet. <b>1 – In progress.</b> Replies received from one or more blocks but not finished. <b>2 – Finished normally.</b>	Numeric

	<b>3 – Timed out</b> (time limit expired).	
<b>Feedback</b>	Whether feedback has been given to the participant during the session at all. <b>0</b> means none given. <b>1</b> means some given.	Numeric
<b>Number_Sections</b>	Number of question blocks for which replies have been received.	Numeric
<b>When_Finished</b>	If the session is finished, the date and time when it finished.	Date
<b>Max_Score</b>	Maximum score of all questions in blocks answered.	Numeric
<b>Total_Score</b>	Sum of scores in all questions in blocks answered.	Numeric
<b>Special_1 to Special_10</b>	These fields can be set to the answers or outcomes of particular questions, if this is set in <i>Session Manager</i> .	Text 50
<b>Time_Taken</b>	If the session is finished, time taken in seconds from start to end of session.	Numeric
<b>Score_Band_Title</b>	If the session is finished, title of score band at end of session.	Text 50
<b>Score_Band_Number</b>	If the session is finished, the number of the score band in the session definition. (This allows access to the full score band text by reading the session database.)	Numeric
<b>Percentage_Score</b>	If the session is finished, the percentage score achieved.	Numeric

### A.3 Answers table

Table **A\_Answer** contains one record for each question in a block that was answered, including for questions in that block that were unanswered.

<b>Field Name</b>	<b>Description</b>	<b>Type</b>
<b>Answer_Index</b>	Identifier of answer record in database.	Numeric
<b>Result_Index</b>	Reference to table <b>A_Result</b> .	Numeric
<b>Question_Index</b>	Reference to table <b>A_Question</b> .	Numeric
<b>Block_Number</b>	Number of block question is in.	Numeric

<b>Question_Number</b>	Number of the question in the block as presented to participant.	Numeric
<b>Signature</b>	Text "signature" code written so the validity of the record can be checked.	Text 20
<b>Times_Answered</b>	Number of times answered. <b>0</b> means presented only but not answered. <b>1</b> means answered once. Other values are reserved for future use.	Numeric
<b>Max_Score</b>	Maximum score achievable for question.	Numeric
<b>Actual_Score</b>	Score for the question.	Numeric
<b>Know_Time_Taken</b>	Not used.	Yes/No
<b>Time_Taken</b>	Not used.	Numeric
<b>Number_Outcomes</b>	Number of outcomes selected (often 1).	Numeric
<b>Outcome_Number</b>	Number of first outcome selected (reference to <b>A_Outcome</b> ).	Numeric
<b>Outcome_Exponential</b>	If more than one outcome selected, the sum of $2^{n-1}$ for n from 1 to number of outcomes. For example, if 9 outcomes, and 2 <sup>nd</sup> , 5 <sup>th</sup> and 9 <sup>th</sup> selected, this number is $2^1 + 2^4 + 2^8 = 274$ .  This examines only the first 32 outcomes, and is unreliable if the question has more than 32 outcomes.	Numeric
<b>Answer_Full</b>	Raw answer in full. What this is will depend on question type. (In a multiple choice question it is the choice text; in a text answer question it is the answer entered; in a multiple response question, it is the answers chosen concatenated separated by a colon; and so on.) Truncated at 16,000 characters.	Memo
<b>Answer_Truncated</b>	Raw answer (duplicate of above field but truncated to first 50 characters).	Text 50



## A.4 Session table

Table **A\_Session** contains a record for each session presented. The record is usually written once when the session is first taken, and is not updated when the session is taken again, even if the session details change.

Fuller information on the session can be obtained by reading the session database.

Field Name	Description	Type
<b>Session_Index</b>	Identifier of session record in answer database.	Numeric
<b>Session_Database_Name</b>	Not used.	Text 50
<b>Session_ID</b>	ID of session definition in session database.	Text 20
<b>Session_Name</b>	Session name.	Text 50
<b>Session_Author</b>	Name of author of session.	Text 50
<b>When_Modified</b>	When session last modified.	Date
<b>Whether_Time_Limit</b>	Whether there is a time limit for the session.	Yes/No
<b>Time_Limit</b>	If so, what it is (in minutes).	Numeric
<b>Number_Sections</b>	Number of possible question blocks in session.	Numeric
<b>Number_Started</b> and other fields up to <b>Average_Time</b>	There are 11 additional fields in this table, which can hold statistics on the session results.  These were used in version 1, but are not normally written to in version 2 of Perception for efficiency reasons. If you want compatibility with version 1 and to use these fields, then you should set <b>Disable statistics calculation = 0</b> in <b>perception.ini</b> . These fields may not be supported in future versions of Perception.	
<b>Last_Updated</b>	This is the time and date this record was last updated, usually the time the record was created.	Date

## A.5 Question table

Table **A\_Question** contains one record for each question presented to a participant. Further information (e.g. the question wording) is available by reading the full question definition in the QML in the question database.

The record is usually written once when the question is first taken, and is not usually updated when the question is taken again, even if the details change. The only exception to this is that field **Number\_Outcomes** is changed if additional outcomes are added to the question.

Field Name	Description	Type
<b>Question_Index</b>	Identifier of question within database.	Numeric
<b>Question_Database_Name</b>	Not used.	Text 50
<b>Question_ID</b>	ID of question in database.	Text 20
<b>When_Modified</b>	Not used.	Date
<b>Question_Author</b>	Not used.	Text 50
<b>Question_Description</b>	Question description.	Text 200
<b>Question_Type</b>	Question type.	Text 10
<b>Topic</b>	Topic question is in.	Text 255
<b>Number_Outcomes</b>	Number of distinct outcomes question has.	Numeric
<b>Times_Presented</b> and other fields up to <b>Standard_Deviation</b>	There are 9 additional fields in this table, which can hold statistics on the question results.  These were used in version 1, but are not normally written to in version 2 of Perception for efficiency reasons. If you want compatibility with version 1 and to use these fields, then you should set <b>Disable statistics calculation = 0</b> in <b>perception.ini</b> . These fields may not be supported in future versions of Perception.	
<b>Last_Updated</b>	This is the time and date this record was last updated, usually the time the record was created.	Date

## A.6 Outcome table

Table **A\_Outcome** contains one record for each outcome of each question.

Field Name	Description	Type
Question_Index	Reference to <b>A_Question</b> .	Numeric
Outcome_Number	Number of outcome, 1 for 1 <sup>st</sup> , 2 for 2 <sup>nd</sup> and so on.  These numbers will usually correspond to the order of the outcomes in the question, unless the question is changed after its details are first stored in the answer database.	Numeric
Outcome_Name	The name of the outcome.	Text 50
Feedback	Feedback associated with outcome, blank if none. This is set when the record is created, and not updated if it changes because the question is edited.	Memo
Times_Selected	This field is not normally updated. If you set <b>Disable statistics calculation = 0</b> in <b>perception.ini</b> , then it is updated to show the number of times the outcome is selected for compatibility with the version 1 software. This field may not be supported in future versions of the software.	Numeric



# Appendix B

## Security database format

### B.1 Overview

The supplied security database (**qmpsecurity.mdb**) is in Microsoft *Access* '97 format, but since it is only accessed by the server software with ODBC, you can convert it to a different format if you need to.

It contains the following tables:

1. Table **G\_Participant** with one record per participant.
2. Table **G\_Group** with one record per group.
3. Table **G\_Membership** which defines which participants are members of which groups.
4. Table **G\_Schedule** with one record per schedule.
5. Table **G\_User** which defines the *Security Manager* and *Enterprise Reporter* users.

You can view or edit the database by hand using Microsoft *Access*, or else read and write to it programmatically. If you write to the database yourself, it is your responsibility to do so correctly. Putting incorrect information into the database is likely to lead to program failure when the Perception programs read the database.

### B.2 Participant table

Table **G\_Participant** contains one record per participant. The license file can limit the number of participants held in the security database. If so, then if you add too many participants into the table, the DLLs will refuse to run.

Field Name	Description	Type
<b>Participant_Index</b>	Identifier of participant within database.	Numeric
<b>Participant_Name</b>	Name of participant.	Text 50
<b>Participant_Password</b>	Password of participant (not encoded).	Text 20
<b>Participant_Details</b>	Details for participant.	Text 50

<b>Checksum</b>	Not used.	Text 20
<b>Properties</b>	Not used.	Numeric

## B.3 Group table

Table **G\_Group** contains one record per group.

Field Name	Description	Type
<b>Group_Index</b>	Identifier of group within database.	Numeric
<b>Group_Name</b>	Name of group.	Text 50
<b>Checksum</b>	Not used.	Text 20
<b>Properties</b>	Not used.	Numeric

## B.4 Membership table

Table **G\_Membership** is used to store group membership and ownership. There is one record for each participant who belongs to each group, and also one record for each user who owns a group.

Field Name	Description	Type
<b>Membership_Index</b>	Index field.	Numeric
<b>Group_Index</b>	Reference to <b>G_Group</b> .	Numeric
<b>Participant_Index</b>	Reference to <b>G_Participant</b> . Or zero if this record refers to a user.	Numeric
<b>Checksum</b>	Not used.	Text 20
<b>Properties</b>	Reference to <b>G_User</b> . Or zero if this record refers to a participant. Will not be zero if <b>Participant_Index</b> is zero.	Numeric

## B.5 Schedule table

Table **G\_Schedule** contains one record for each instance of a participant or group being scheduled to a session.

Field Name	Description	Type
<b>Schedule_Index</b>	Identifier of schedule within database.	Numeric
<b>Session_ID</b>	ID of session within the session database. Only the first 20 characters are usually used.	Text 50
<b>Session_Name</b>	Name of session in session database.	Text 50
<b>Session_Database</b>	Not used. The session database used is not defined in the security database.	Text 50
<b>Participant_Index</b>	Reference to <b>G_Participant</b> . Zero if a group is scheduled.	Numeric
<b>Group_Index</b>	Reference to <b>G_Group</b> .  If <b>Participant_Index</b> is zero, then this field is the group that is scheduled.  If <b>Participant_Index</b> is set, then this field is used to show the group the participant is scheduled as a member of. If it's zero, then the participant is scheduled as a member of no group.	Numeric
<b>Schedule_Name</b>	Descriptive name of the schedule.	Text 50
<b>Restrict_Times</b>	Whether the schedule is only available in a certain time period.	Yes/No
<b>Schedule_Starts</b>	If <b>Restrict_Times</b> is set, when the schedule starts being available.	Date
<b>Schedule_Stops</b>	If <b>Restrict_Times</b> is set, when it stops being available.	Date
<b>Max_Attempts</b>	If <b>Restrict_Attempts</b> is set, this is the number of attempts left.	Numeric
<b>Restrict_Attempts</b>	Whether only a limited number of attempts permitted.	Yes/No
<b>Checksum</b>	Not used.	Text 20
<b>Properties</b>	Not used.	Numeric

## B.6 User table

The **G\_User** table holds *Security Manager* and *Enterprise Reporter* users. The first record in this table holds the master user, usually called **Manager**.

Field Name	Description	Type
<b>User_Index</b>	Index field.	Numeric
<b>User_Name</b>	User name.	Text 50
<b>User_Password</b>	The password for the name. This password is encrypted by a “one-way” algorithm and cannot be decoded, even by Question Mark.	Text 50
<b>User_Status</b>	Used to define the capabilities of the user.	Numeric
<b>Properties</b>	Used to define the capabilities of the user.	Numeric



# Appendix C

## Question database format

### C.1 Overview

The question database is in Microsoft *Access* '97 (.MDB) format, but uses a .question file extension. Question databases are usually managed by the *Question Manager* program.

The question database holds a bank of questions, hierarchically organized by topic. There are two tables, one for topics (**Q\_Topic**) and one for questions (**Q\_Question**).

You can view or edit the database by hand using Microsoft *Access*, or else read and write to it programmatically. If you write to the database with any application other than *Question Manager*, it is your responsibility to do so correctly. Putting incorrect information into the database is likely to lead to program failure when the Perception programs read the database.

### C.2 Topic table

The **Q\_Topic** table contains one record for each topic and subtopic in the database.

Field Name	Description	Type
<b>Topic_ID</b>	Numeric identifier for each topic. This is referenced from table <b>Q_Question</b> . Numbers are randomly generated, and unique within the database.	Numeric
<b>Topic_Name</b>	Actual name of topic. Topic names are stored with backslash characters (\) between each element of the name. For example, <b>Topic\Subtopic\Subtopic</b> . The database may not contain two topics with the same name.  The topic name for each question is also stored inside the QML for a question, so if you change a topic name from outside <i>Question Manager</i> , you should also change the name inside all the questions that reference it.	Text 255

Records are sorted in alphabetical order of topic name. For example, if there is a database with a topic **Geography**, with 2 subtopics, **Borders** and **Capital Cities** and the subtopic **Borders** has a further subtopic **America**, then the following topics will be stored in the database in the following order:

- **Geography**
- **Geography\Borders**
- **Geography\Borders\America**
- **Geography\Capital Cities**

### C.3 Question table

The **Q\_Question** table contains one record for each question in the database. The information about the question itself is held in the QML language inside field **QML\_Data**, and the other information is just used to index and categorize the question.

See the *Authoring manual* for a full description of the QML language.

Field name	Description	Type
<b>Question_ID</b>	Unique ID for each question. This is a string of 16 random digits used to uniquely identify a question. Duplicates should not be permitted. This ID should match the question ID inside the field <b>QML_Data</b> .	Text 20
<b>Topic_ID</b>	Reference to <b>Topic_ID</b> in table <b>Q_Topic</b> . The topic name is also duplicated inside the QML.	Numeric
<b>Description</b>	Question description. This is taken from QML, truncated if necessary.	Text 200
<b>Difficulty</b>	This field is not used at present, and should be set to zero.	Numeric
<b>Question_Type</b>	The question type, taken from the QML.	Text 10
<b>Author</b>	Name of person who authored the question, as set in <i>Question Manager</i> . This can be safely changed by editing the database with Microsoft <i>Access</i> .	Text 50
<b>Maximum_Score</b>	Maximum score achievable for the question, constructed from the QML. This value is only used for indexing, and display purposes. The server software ignores it, and calculates the maximum score directly from the QML.	Numeric

<b>Modified_Date</b>	Date and time the question was last modified in <i>Question Manager</i> .	Date
<b>Encoded</b>	How the question data is encoded. At present, the only supported value is 0 for unencrypted question data.	Numeric
<b>QML_Data</b>	Raw unprocessed QML data, containing the question data in ASCII form. See the separate QML specification for full information on this. Maximum length 30,000 characters.	Memo
<b>Lock_Question</b>	Whether question is currently open in <i>Question Manager</i> .  When a question is opened in <i>Question Manager</i> , this is set, and when it is closed, this is unset. If the value is set, then this may mean that the question is currently being edited in <i>Question Manager</i> , and you should not change it.	Yes/No



# Appendix D

## Session database format

### D.1 Overview

The session database is in Microsoft *Access* '97 (.MDB) format, but uses a .session file extension. Session databases are usually managed by the *Session Manager* program.

The session database holds one or more sessions. Each session is entirely independent, and there is no hierarchy or other grouping of sessions.

In the question database, the “meat” of the data is held in an unstructured ASCII QML field and the rest of the database fields are primarily an index to the data. But in the session database, almost all the session data is held in database fields. This means it is more easily accessible, but also the session database structure is more complicated. There are six tables in the database:

- **S\_Header** contains control blocks, one record for each session in the database.
- **S\_Index** has a list of all records in the **S\_Question**, **S\_Item**, **S\_Jump** and **S\_End** tables.
- **S\_Question** contains question blocks, one record for each question block in each session in the database. There are one or more question blocks for each session.
- **S\_Item** contains the questions within a question block. Each question block contains one or more records in **S\_Item**.
- **S\_Jump** contains jump blocks, one record for each jump in each session in the database. There can be zero, one or more jump blocks in any session.
- **S\_End** contains score bands, one record for each score band in each session in the database. There are one or more score bands for each session.

Each table contains all the relevant records for all the sessions in the database. Which records belong to which session is defined within table **S\_Index**.

You can view or edit the database by hand using Microsoft *Access*, or else read and write to it programmatically. If you write to the database with any application other than *Session Manager*, it is your responsibility to do so correctly. Putting incorrect information into the database is likely to lead to program failure when the Perception programs read the database.

Several fields in the session database are unused. These are set to zero if numeric fields or left blank if text fields. They may be used in future versions.

## D.2 Header table

The **S\_Header** table contains one record for each session in the database. For example, if a session database contains three sessions, then it will contain three records in this table. The table contains the session name and ID, and the information set within the control block in *Session Manager*.

Field name	Description	Type
<b>Session_Name</b>	Name of the current session.	Text 50
<b>Session_ID</b>	Unique ID for each session. This is a string of 16 random digits used to uniquely identify a session. Duplicates should not be permitted.	Text 20
<b>Author</b>	Name of the person who authored the session. Set in <i>Session Manager</i> options dialog. You can safely change this field in <i>Microsoft Access</i> , if you wish to change the author name.	Text 50
<b>Checksum</b>	Not used.	Numeric
<b>Version</b>	Version of <i>Session Manager</i> software used to create the session.	Text 6
<b>Serial_Number</b>	Not used.	Text 10
<b>Password_Protect_Answer_DB</b>	Not used.	Yes/no
<b>Answer_DB_Password</b>	Not used.	Text 10
<b>Save_Answers</b>	Whether answers for the session to be saved in answer database.	Yes/No
<b>Name_Of_Answer_DB</b>	Not used.	Text 50
<b>Open_Session</b>	Whether or not session can be presented with <b>open.dll</b> . If not, can only be presented with <b>perception.dll</b> or via <i>PIP</i> .	Yes/No
<b>Session_Password</b>	Used to hold the session password, which protects access to the session in <i>Security Manager</i> and when publishing. If blank, no password is set.	Text 10
<b>Save_Block_Data</b>	Not used.	Yes/No
<b>Save_Answer_Data</b>	If <b>Save_Answers</b> is set, then should all the tables in the answer database be filled? If <b>Yes</b> , they are. If <b>No</b> , then the only ones filled are the results and session information tables, i.e. there is no detailed information stored about the answers and outcomes for each question.	Yes/No
<b>Number_Of_Questions_To_Present</b>	Not used.	Numeric

<b>Session_Timed</b>	Whether the session has a time limit.	Yes/No
<b>Time_Limit</b>	If <b>Session_Timed</b> is set, this is the time limit of session in minutes. From 1 to 4320 (3 days).	Numeric
<b>Template_Name</b>	Name of the default template file to apply to the session. This file (e.g. <b>example.template</b> ) can be over-ridden for each individual block.  The path for templates is defined outside the database (in <b>Session Manager.ini</b> or <b>perception.ini</b> ).	Text 50
<b>When_Feedback</b>	When is feedback to take place: <b>0</b> means no feedback at all. <b>1</b> means feedback after each block. <b>3</b> means feedback at the end of the session. Other values are reserved.	Numeric
<b>End_Feedback</b>	If <b>When_Feedback</b> is not <b>0</b> , then this field defines which questions feedback is to be displayed for: <b>0</b> means for all questions. <b>1</b> means wrong and unanswered questions. <b>2</b> means wrong questions. <b>3</b> means right and wrong (but not unanswered) questions.	Numeric
<b>Exclude_Unscored</b>	If <b>When_Feedback</b> is not <b>0</b> , then this defines whether questions with a maximum score of <b>0</b> be excluded from feedback. Often such questions are unmarked and should not be given feedback.	Yes/No
<b>Allow_Escape_Between_Blocks</b>	Not used.	Yes/No
<b>Modified_Date</b>	Date and time the session was last modified in <i>Session Manager</i> .	Date
<b>Lock_Session</b>	Whether or not the session is currently open in <i>Session Manager</i> .	Yes/No

## D.3 Index table

The **S\_Index** table indexes the question, jump and score bands and the references to questions within question blocks.

Field name	Description	Type
<b>Session_ID</b>	Unique reference for the session, matches with the <b>Session_ID</b> field in table <b>S_Header</b> .	Text 20
<b>Block_Type</b>	Type of block:	Numeric

	<b>1</b> means question block. References <b>S_Question</b> . <b>2</b> means reference to questions in a question block. References <b>S_Item</b> . <b>3</b> means jump block. References <b>S_Jump</b> . <b>4</b> means score band. References <b>S_End</b> .	
<b>Block_Order</b>	The order of the block in the session. This matches the order in which the blocks are displayed in <i>Session Manager</i> .	Numeric
<b>Block_ID</b>	Reference to the field of the same name in <b>S_Question</b> , <b>S_Item</b> , <b>S_Jump</b> or <b>S_End</b> .	Numeric

In order to compute the contents of a session, *Session Manager* and the server software look at the control block in **S\_Header** and then search for all records in **S\_Index** with a matching **Session\_ID**. These records are then sorted in ascending **Block\_Order**, and this defines the order and place of each element in the *Session Manager* tree and the order the server software delivers blocks.

All references to score bands must be after references to all other blocks. Each question block must be followed by one or more references to **S\_Item**, and these can appear only immediately after a question block, not in any other placing. All records in the subsidiary tables are only linked to their place in a session by their entry in this table. If you are creating entries in this table, it is vital that you do so with extreme care.

## D.4 Question block table

The **S\_Question** table contains one record for each question block, defining the header information for the block. The actual questions are defined in table **S\_Item**.

Field name	Description	Type
<b>Block_ID</b>	Unique number of each record in this table. Used to refer to the record from <b>S_Index</b> .	Numeric
<b>Block_Name</b>	Name of the question block, e.g. "Part 2." This can be displayed to the participant as the title or header of the document containing the questions in the block.	Text 50
<b>Introduction_Text</b>	Contains introductory text, which is usually displayed at the head of the block when presented to the participant. Maximum 16,000 characters, interpreted as HTML.	Memo
<b>Feedback</b>	Whether feedback is on for this block. If set, then the settings in <b>S_Header</b> take effect. If set off, then no feedback is presented even if feedback is set in <b>S_Header</b> .	Yes/No
<b>Shuffle questions</b>	Whether the questions within this block should be presented in randomly shuffled order (in addition to any randomization within	Yes/No



	the question references).	
<b>Use_Template_File</b>	Whether a different template file to the one defined in <b>S_Header</b> is to be used when presenting this block.	Yes/No
<b>Template_Name</b>	If so, name of template file.	Text 50
<b>Session_Time_Limit_Applies</b>	Not used.	Yes/No
<b>Block_Time_Limit</b>	Not used.	Yes/No
<b>Time_Limit</b>	Not used.	Number

## D.5 Question reference table

The **S\_Item** table contains a list of definitions of questions within a question block. Each record refers to one of these three possibilities:

- All questions in a topic.
- A number of questions from a topic chosen at random.
- A single question in a topic.

Several instances of **S\_Item** can follow (that is to say be indexed from **S\_Index**) one instance of **S\_Question**. So for example to include 3 specific questions in a block, you would have 3 instances of **S\_Item** each referring to a specific question.

Field name	Description	Type
<b>Block_ID</b>	Unique number of each record in this table. Used to refer to the record from <b>S_Index</b> .	Numeric
<b>How_Are_Questions_To_Be_Picked</b>	How questions are to be picked: <b>0</b> represents randomly. <b>1</b> represents all from a topic. <b>2</b> represents a single question.	Numeric
<b>Include_Sub_Topics</b>	If picking randomly or all, whether to include questions in subtopics in the selection.	Yes/No
<b>Topic_ID</b>	Topic ID in question database that question(s) come from. This refers to table <b>Q_Topic</b> in the question database.	Numeric
<b>Number_Of_Questions</b>	If picking randomly, number to pick.	Numeric

<b>Single_Question_ID</b>	If picking a single question, question ID in the question database of question to pick. This refers to table <b>Q_Question</b> in the question database.	Text 20
<b>Special_Question</b>	Whether results of question to be stored in a special way in the answer database.	Yes/No
<b>Field_Number</b>	If so, which field (1-10).	Numeric
<b>Special_Field_Contents</b>	And if so, what information is to be stored in the relevant field: <b>0</b> means the answer. <b>1</b> means the outcome.	Numeric
<b>Name_Of_Question_DB</b>	Not used.	Text 255

## D.6 Jump table

The **S\_Jump** table holds one record per jump.

<b>Field name</b>	<b>Description</b>	<b>Type</b>
<b>Block_ID</b>	Unique number of each record in this table. Used to refer to the record from <b>S_Index</b> .	Numeric
<b>Block_Name</b>	Name of the jump block for reference purposes.	Text 50
<b>Condition</b>	The condition on which the jump is to be made. Valid conditions include: <b>ALWAYS</b> <b>OUTCOME = "name"</b> <b>NOT OUTCOME = "name"</b> One of <b>Value = n</b> , <b>NOT Value = n</b> , <b>Value &lt; n</b> or <b>Value &gt; n</b> , where <b>n</b> is a number and <b>Value</b> is one of <b>PERCENTSCORE</b> , <b>POINTSCORE</b> , <b>PERCENTBLOCK</b> or <b>POINTSBLOCK</b> .	Memo
<b>Destination</b>	Not used.	Number
<b>Jump_To_Block_ID</b>	This is the jump destination. A value of <b>0</b> means jump to the end of session. Any other value is the block ID of a question block to jump to (a reference to <b>Block_ID</b> in <b>S_Question</b> ).	Number
<b>Jump_To_Session_ID</b>	Not used.	Text 20
<b>Other_Jump</b>	Not used.	Memo

## D.7 Score band table

The **S\_End** table contains one record for each score band of each session.

Field name	Description	Type
<b>Block_ID</b>	Unique number of each record in this table. Used to refer to the record from <b>S_Index</b> .	Numeric
<b>Block_Name</b>	Name of the score band. This can be used to label the block with a message that is stored in the answer database with the numeric score, for example, "Pass" or "Fail."	Text 50
<b>Lower_Score_Band_Value</b>	The lower, percentage, limit of the score band, e.g. 0%.	Numeric
<b>Upper_Score_Band_Value</b>	The upper, percentage, limit of the score band, e.g. 20%.	Numeric
<b>Message</b>	Score band message presented to the participant at the end of the session for achieving this score. Maximum 16,000 characters, interpreted as HTML.	Memo
<b>Is_Session_ID_Jump</b>	Whether a jump to another session is to happen if this score is achieved.	Yes/No
<b>Jump_To_Session_ID</b>	If yes, session ID of session in this session database to jump to. This refers to the field <b>Session_ID</b> in table <b>S_Header</b> .	Text 20
<b>Is_Other_Jump</b>	Whether another kind of jump to take place if this score is achieved.	Yes/No
<b>Other_Jump</b>	If yes, place to jump to, usually a URL.	Memo

In order to determine which score band is used, the percentage score for the session is calculated by computing the score achieved and dividing it by the maximum score for all questions in all blocks presented to the participant. This value is then rounded to become an integer number from 0 to 100. (In the rounding, any fractional value of 0.5 or higher rounds up, any fractional value up to 0.49999... rounds down.)

The score bands are then evaluated in the order in the index table, and the first one where the score is greater than or equal to the **Lower\_Score\_Band\_Value** and less than or equal to the **Upper\_Score\_Band\_Value** becomes the score band chosen for the session. If none are found, then no score band is used. For example, if **Lower\_Score\_Band\_Value** is 10 and **Upper\_Score\_Band\_Value** is 20, then any percentage score of 10 through 20 meets this score band (or in fact before rounding, any percentage score of 9.5 through 20.49999).



# Appendix E

## E.1 Report database format

The *Enterprise Reporter* report definition database is called by default **qmpenterprise.mdb**. It contains the report definitions for *Enterprise Reporter*. Some initial report definitions are shipped with the software. Users of *Enterprise Reporter* can edit or delete the shipped definitions, and create more definitions.

There are five tables in the database. Each record in a table corresponds to a single report definition.

- Table **R\_List** contains one record for each list report definition.
- Table **R\_Overview** contains one record for each session overview report definition.
- Table **R\_Participant** contains one record for each participant report definition.
- Table **R\_Question** contains one record for each question statistics report definition.
- Table **R\_Survey** contains one record for each survey report definition.

If additional report types are added to *Enterprise Reporter*, they will be added by making additional tables.

Each table has a field **REPORT\_INDEX** which is the index number of the report definition. Unlike the other databases used by Perception, the other fields in the database are not documented in this manual. The reports database is an *Enterprise Reporter* configuration file, and it's only designed for internal use by *Enterprise Reporter*.

Each table is independent, without any relationship to other tables. Except that table **R\_List** contains a field **PARTICIPANT\_REPORT\_INDEX**, which contains an index number of a participant report that is referred to in that list report.



# Index

- About function 22 , 34, 98
- Active Server Pages See ASP
- Administration 73-74, 91-92
- AICC 152
- Answer database 14, 93-94, 146-147, 171-177
- ASP 26, 38
- Assigning See Scheduling
- Attempts allowed 71, 83-84
- Automatic restart 23, 64, 66
  
- Branching at end of session 62-63
- Branching within a session See Jumping within a session
- Browsers 6, 38, 60-61, 63-64 , 72, 93
  
- Cleaning up progress files 23, 43
- Command line parameters 50-51, 52-53
- Continue button 55
- Correlation 129-130
  
- Database drivers 6, 27, 31
- Databases 14, 29, 171-195
- Date limits See Scheduling
- Detailed information See Participant link
- Difficulty 128-129
- Discrimination See Correlation
- Domino See Lotus Domino
- DSN names See ODBC
- Dynamic filtering See Report filters
  
- End block See Score band
- Enterprise Reporter 16, 24-26, 38, 93-149
- Error file See perception.err
  
- Facility See Difficulty
- Feedback 55, 56, 110
- Filtering 139-142
- Final page See Score band
- Format files 14-15, 18, 32, 33 42, 53 58
- FTP 46

Graphics	See Resources
Groups	52, 53 54, 69, 70, 77, 79, 81, 84-85, 89, 95, 106
Home button	21, 63
IIS	5, 26
Importing participants	88-91
INI file	See perception.ini
Initializing	See resetting
Installation	5-40, 148-149
Installing multiple copies	30-31
Integration	See Perception Integration Protocol
Internet Explorer	See Browsers
ISAPI	26, 38
Java	15, 19, 47
Jumping at end of session	See Branching at end of session
Jumping within a session	56
Kiosk mode	64
License file	5, 11-12, 14, 17, 33, 42, 70, 89, 149, 160
Limiting attempts	See Attempts allowed
List of sessions	See Login screens
List reports	98, 116-124
Logging	15-16, 17, 20, 167-169
login.format	15, 18, 42, 53-55, 65, 156
Lotus Domino	26
main.format	See Format files
manager.format	15, 18, 42, 76
Manual restart	23, 65-66
Mean for outcome	131
Microsoft DLLs	16, 35-36
mfc42.dll	See Microsoft DLLs
msvcrt.dll	See Microsoft DLLs
Multimedia	See Resources
Name of participant	See Participant name
Netscape Navigator	See Browsers
ODBC	6, 7, 19, 24, 27-29, 38, 39-40, 45, 46, 149, 153
ODBC tracing	39-40
Offline	92
open.dll	13, 22, 51-53



Optimization	20-21, 160-163
Oracle	31-32
Parameters	See Command line parameters
Participant link	116, 123
Participant name	49-50, 67-68, 69, 80
Participant reports	97, 100-111, 123
Participant viewing of reports	25, 143-145
Participants	69, 79-80, 143-144
Passwords	
perception.dll	13, 49-51
perception.err	16, 17, 35, 38, 44, 167-168
perception.ini	13, 17-24, 30, 34, 39, 42
Perception Integration Protocol	15, 18, 20, 24, 151-152, 169
perception.log	See Logging
Perception support site	See Support site
Permissions	10-11
Personal Web Server	See PWS
PIP	See Perception Integration Protocol
Profiles of users	76-77
Progress files	16, 18, 33, 43, 44, 55, 64
Publishing	44-48
PWS	5, 26
qmper.dll	36-37
Question by question delivery	59-61
Question database	14, 48, 183-185
Question Statistics reports	98, 124-131
Refresh button	57, 64
Registering qmper.dll	See qmper.dll
Reload button	See Refresh button
Report database	14, 93, 99, 195
Reporter	See Enterprise Reporter
Report filters	See Filtering
Report Manager	99-100
Resetting DLLs	42-44, 92
Resources	15, 19, 47, 155
Restarting	See Automatic restart or Manual restart
Running the software	49-53, 72-75
Schedule name	50, 83
Scheduling	70, 83-87
Scoring	

Security	24-25, 48, 73-75, 153-158
Security database	14, 68-71, 89, 179-182
security.dll	See Security Manager
Security Manager	13, 67-92
Server configuration file	See perception.ini
Server key	23, 157
Server URL	21-22, 27
Server variables	164-167
Session database	14, 47-48, 82, 187-193
session.dll	13, 21, 55-56
Session ID	47, 51
Session Manager	31, 45, 45, 46, 62
Session Overview reports	98, 112-115
settings.asp	16, 24-26, 143, 145, 148
Speed of software	See Optimization
Spreadsheets	146
SQL Server	31-32
Students	See Participants
Support site	3
Survey reports	98, 132-138
System DSN	See ODBC
Technical support	20, 35, 58
Template caching	20-21, 42-43, 161
Template files	14-15, 18, 47, 58, 60
text.format	See Format files
Time limits	61, 84
Timing leeway	23, 61
Topics	48, 111, 125
Tracing	See ODBC tracing
Troubleshooting	34-35
UNIX	26
Unfinished sessions	See restarting
Upgrading	32-34
Users	68-69, 76-78
Variables	See Server variables
Version 1	21, 32-34, 175, 176
Webservers	5, 26-27
Web sharing	7-9, 18-19, 30, 33, 38, 149, 153
Windows NT authentication	159-160