

Requirements for all Science Museum computer exhibits

A guide to your contractual obligations and information on how to avoid mistakes made in previous projects.

Developing a computer exhibit for the Science Museum is a tremendous opportunity. The museum has 1.5 million visitors a year and is recognised as a world leader in interactive exhibits. We're a world class institution looking for innovative, exciting, off-the-wall exhibits and have some very experienced people to help you create them.

On the flip side, the diversity of our audience and the nature of museum exhibits makes them rather different from other multimedia projects. We also have a set of needs connected to maintenance, administration and quality control, which make the whole business rather more complicated.

As you may be aware, the museum has recently opened a new wing containing around 200 new computer exhibits. This document therefore tries to set out what we've learnt during this project, together with the things which we thought were obvious but our exhibit contractors didn't until it was too late.

Although you are contractually obliged to follow these recommendations, this document has also been produced to save you pain, so please read it whether you're a programmer, designer or project manager.

Joe Cutting March 27, 2003

Contents

Contact information	2
Exhibit development	2
Exhibit development process	2
Development platform.....	3
Exhibit design requirements	3
Evaluation	5
Robustness	6
Hardware platform	7
Loan computers.....	7
Appendix 1: Deliverables for a software exhibit	8
Software deliverables.....	8
Appendix 2: Robustness of common authoring tools	10

Contact information

The Museum will assign you a contract manager, who should always be your first point of contact. In addition, you may need to contact the people listed below. If you do so, you should always confirm any agreed actions by e-mail and copy the mail to your contract manager.

Interactive Media Unit

Dave Patten (Manager)	020 7942 4864	d.patten@nmsi.ac.uk
Ken Chan	020 7942 4853	k.chan@nmsi.ac.uk
Joe Cutting	020 7942 4866	j.cutting@nmsi.ac.uk
Jo Saull	020 7942 4867	j.saul@nmsi.ac.uk

Head of Visitor Research

Kate Steiner	020 7942 4821	k.steiner@nmsi.ac.uk
---------------------	---------------	----------------------

Exhibit development

Exhibit development process

A typical exhibit development process starts with the Museum setting the initial scope and aims of the project. The contractor then returns with a storyboard for approval by the museum. Exhibit development can now start and the contractor produces a series of prototypes (usually three), which are each evaluated in turn (see the 'Evaluation' section below) and refined to produce the next one. Once the exhibit is finished it is delivered and tested by the Museum for robustness (see the 'Robustness' section below). It is then installed on gallery. There then follows a defects-liability period, during which the contractor is responsible for any problems with the exhibit. Once this is over, final signoff and payment can be made.

Development platform

Museum exhibits are generally written using Director or Web technologies such as HTML and Flash. However, we are open to other development environments, but we strongly recommend that you do not use:

- Low level languages like C or Java. These languages make it very tricky to incorporate the number of changes generally demanded by the evaluation process. Wellcome Wing developers using these languages generally found themselves taking a long time to produce exhibits that tended to be less reliable.
- The latest ('X.0') version of a development environment. These tend to be buggy and unreliable, which is a crucial problem for robustness (see below).

If you think you need to use this type of programming environment, please contact the museum interactives team (Dave Patten or Joe Cutting) before starting development. Remember that whatever environment you choose, it is your responsibility to make sure it is up to the job.

Problems

If you run into a technical problem that you can't solve during development, then do get in touch. The Museum's interactive team have a lot of experience and may have the answer or may be able to put you in touch with another developer working in the same area. However, bear in mind that we have only a small team and are not able to spend much time on each exhibit. If you are using technologies that you are not familiar with, we recommend you enlist a third-party source of expertise and do not rely on the Museum's team.

Exhibit design requirements

The Museum is keen to encourage developers to express their creativity in the design of their exhibits and welcomes innovative designs. However, we have a few requirements that it is essential you stick to if our visitors are to be able to use the exhibit. *The most important of these is that you realise that your exhibit will be running unstaffed, on a stand-alone basis, without any external signage.*

Attractor screens

When not in use, your exhibit should default to an 'attractor' screen. As well as providing an introduction to the exhibit, this screen should work as a screensaver to prevent the image 'burning in' to the screen. Contrary to popular belief, we have experienced burn-in on LCD and plasma screens, as well as CRTs.

If your exhibit is using a touchscreen

- People aren't that accurate when using touchscreens, so active areas on the screen should be at least 25 mm (1 inch) wide and at least 25 mm apart. Accuracy is even lower towards the edges of the screen.
- Touchscreens simulate a mouse by sending the exhibit a mouse click when the screen is touched. The normal Windows standard for mouse operations is to have them execute on 'mouse up' – when the mouse key is released. Touchscreen exhibit software should be written to do everything on 'mouse down', otherwise visitors just keep their fingers on the screen and push harder.
- The cursor should be switched off. You should also include an editable preferences file that allows the cursor to be switched on and off. This means that you won't have to make separate versions for development and delivery.

If your exhibit uses a trackball, note that our visitors find it almost impossible to drag using a trackball. The exhibit should therefore be written so that dragging is not

necessary. Trackball exhibits must show an on-screen cursor.

If your exhibit uses some other interface, such as buttons, joystick, steering wheel etc., it is also essential that you deal with the cursor in an appropriate way.

Timeouts

If no-one touches your exhibit for more than 30 seconds it should reset itself back to the attractor sequence. You should also give the visitor some warning of this, so after 30 seconds without input the exhibit should bring up a message saying something like 'Are you still there? Touch the screen or I will restart in ten seconds.' The period of time before the exhibit resets should also be stored in an editable preferences file.

Sound

Good sound effects can really enhance an exhibit. However, we cannot predict the sound levels in any museum gallery, so you should never make the sound track essential for the visitor's understanding of an exhibit. Therefore, exhibit instructions should always be accompanied by the equivalent text. Your exhibit should also be silent during its attractor state, as sounds like these quickly become irritating and destroy the ambience of the gallery. Finally, do be aware that Windows NT4 handles sounds in a different way from other versions of Windows, so be sure to test any sound effects you have on this platform early in development.

Novel interfaces

The Museum likes to encourage contractors to develop novel interfaces and ways of controlling exhibits. However, there are a few things you should bear in mind:

- The Wellcome Wing had several exhibits with novel interfaces and without exception none of them proved robust enough when first installed. Since the opening, we have improved and reinstalled these interfaces and they are now working well. If you are going to develop novel mechanics for an interface, we strongly recommend you consult the Museum's maintenance expert (Ashley Beugge) during the planning stage.
- A large number of Wellcome Wing contractors were keen to incorporate 'new' ways of using trackballs. However, evaluation showed that visitors found most of these very difficult to use and preferred the standard system of a trackball just controlling a pointer.
- If your exhibit has a video camera, the camera should display images as 'mirror images'; visitors find it impossible to position themselves unless you do this.
- You should also incorporate and document an option to operate the exhibit without using the novel interface (that is using a keyboard and mouse). Similarly, you should provide a way of testing the interface without using the main exhibit software.

Other requirements

The Wellcome Wing has around 200 computer exhibits. Although most of them are now working as intended, we've done some calculations and worked out that only around 6 per cent of the original 'final versions' that were delivered to us are still installed. All the other exhibits have had to have new versions of the software installed, generally due to robustness issues (see below) or content changes (generally typos). On average, each exhibit was installed two-and-a-half times, some as many as five or six times. Multiply this by 200 exhibits and you can see that we did roughly 500 installations.

As it's unlikely that future exhibits will prove to be more reliable, we have some extra requirements to make installation easier:

- All exhibits should have an automatic installation program which allows the exhibit to be installed in the folder of our choice. This should take care of the complete installation process. We have found a **free** installation program called 'install maker', which is available from **www.clickteam.com**. This seems to do everything needed and it takes about ten minutes to make an installer.

- Each exhibit should display its version and date of creation on a splash screen at start-up.

Keeping to these requirements will also save you time, as we will be able to install your exhibit faster and it will be much less likely that you'll waste time hunting for bugs produced by us installing the wrong version of your software.

Evaluation

The major difference between developing a computer exhibit for the Science Museum and general multimedia development is the evaluation process.

What is evaluation?

The general idea is that exhibit developers produce prototypes, which are then tested in the Museum to see how visitors respond to them. These responses are then written up as a report, which is sent to the developer, who changes the exhibit design accordingly. Each exhibit usually goes through three evaluations before final delivery.

The advantage to the museum of this process is that we ensure that our visitors will understand the exhibit. The advantages to you the contractor are that it gives you a chance to experiment with new and different designs and formalises the process by which the museum feeds back comments to you. You know that comments will only be sent a fixed number of times and in a written document that everyone agrees on. Exhibit developers are normally paid by the Museum in stages which are tied in to the delivery of prototypes for evaluation.

What we need from developers for evaluation

Dates for evaluation of your exhibit will be agreed during the drawing-up of the exhibit contract. The evaluation generally takes about a week, with a further week for writing the report. Please be aware that the evaluation schedule is very tight and if you're more than a day late for your evaluation you'll miss your slot and will probably have to wait for a month or so for another one. This will completely throw out your exhibit and payment schedules.

- If you have been lent a computer by the Museum, you should bring the computer to the Museum with your prototype loaded on it and ready to run.
- If you haven't been lent a computer, you should bring in your prototype on a CD. Please note that sending the file over the Internet or on a zip disk is not acceptable.
- If you will need any extra hardware for your prototype to run, please call the interactive media team at least a week before.

Your exhibit prototype will be tested on real visitors to the museum. Many developers find it useful to watch some of the evaluation, which is possible if arranged in advance.

Although we've been evaluating exhibits for many years now, the visitors still surprise us with their reactions. This means that it's particularly important that you *don't spend too much of your development time producing the first prototype*. We recommend the following as a rough guide:

- First prototype. Should just contain the 'core interaction' of the exhibit designed to get the main message of the exhibit across. Graphics should be rough 'placeholders' in case substantial redesign is needed. Paper storyboards alone are not acceptable.
- Second prototype. Will incorporate recommendations from the first evaluation report. At least some sections should incorporate the final graphical style, so that we can confirm that we're happy with it.

- Third prototype. Should incorporate practically all of the exhibit's interaction and content, plus recommendations from other evaluation.
- If you have any queries about what is needed for an evaluation prototype, please call the Head of Visitor Research (Kate Steiner). It is particularly important that you call her if for any reason you feel that your schedule or the technicalities of your exhibit (e.g. extensive live video) will not allow a full evaluation cycle.

What you'll receive after evaluation

Around two weeks after the start of evaluation, you'll receive an evaluation report detailing problems with the prototype and suggestions for improvements. This report will also contain any content or design problems that need attention. Due to the unpredictable nature of evaluation, we recommend that you halt your development schedule for these two weeks to avoid wasting time doing work which then needs to be scrapped. There will usually be a meeting scheduled at the end of the evaluation, where we will discuss the report with you.

Robustness

Museum exhibits are set to automatically start up at 10 am and must then run completely unattended until sometimes as late as 11 pm. How well an exhibit does this is known as its 'robustness'.

Robustness problems are very difficult to diagnose and fix because:

- Sometimes they only turn up after many hours of continuous use and it's very difficult to monitor exhibits once they've been installed on gallery.
- Most robustness problems are due to inherent problems with the tools used to create the exhibit – so the only way to fix the problem is to change the tool.
- Typical new-media products such as prototypes and promotional material do not need to be robust, so you may be unused to dealing with these issues.

Robustness problems generally happen because:

- Some pieces of software have a built-in problem, whereby every time they perform some action they reserve a new bit of memory on the computer. Eventually, the computer runs out of memory and crashes. This is known as a 'memory leak'. For typical office use this is not a big problem, as the program is usually restarted several times a day, so the computer never runs out of memory. However, memory leaks cause big problems for computer exhibits, as the program must run continually for over eight hours. You can test for a memory leak by running the software and getting the computer to display how much free memory it has – if the free memory consistently goes down, then there's a problem.
- Exhibits generally have a number of options and if the exhibit has been insufficiently tested a visitor may find a particular combination of options which either make the exhibit crash or unusable.

To keep robustness problems to a minimum, we recommend that you do the following:

- Make sure the tools you use to make your exhibit are well established and from a reputable source. In particular, be very wary of the latest 'X.0' versions of software. Non-commercial software such as that produced by research institutions should be treated with particular caution.
- If you are planning to use any software that you're unsure about, we strongly recommend you do some robustness testing on it **before** you finalise which technologies you're going to use for your exhibit. In particular, we recommend testing for memory leaks (see above).

- Be sure to test your exhibit thoroughly before final delivery. A small amount of time spent testing in your office can save a lot of time trying to find obscure bugs while on gallery. We recommend that you do two main types of testing:
 - Get someone who is not directly involved in developing the exhibit to sit down and test it for an hour or two to make sure that none of the options produce unexpected results.
 - Create a version of the program which will automatically run through the whole thing, selecting options at random, and leave it running for a long time, such as over a weekend.

Hardware platform

The Museum has a standard hardware platform which all computer exhibits run on. Currently (April 2001) it is:

600 Mhz Pentium III
 128 MB RAM
 20GB hard drive
 Nvidia Riva TNT 1 graphics chipset running at 1024x768 in True Colour
 SoundBlaster AudioPCI 64V sound card
 Intel motherboard and chipset

Running Windows NT 4 Service Pack 6.01a

Our experience with the Wellcome Wing has shown that any non-standard equipment will cause ten times as many problems during installation and maintenance. So even if the actual cost increase is fairly small, we have made it a rule to allow no deviations from this hardware specification. It is your responsibility as an exhibit developer to make sure your exhibit runs on the standard hardware. Please don't ask for more powerful hardware – all of the interactive media team have developed for eight-bit computers and will bore you with the details if you're not careful.

Be particularly careful to design your graphics to the correct resolution (*unlike some nameless Wellcome Wing developers, who had to redo all of their graphics*). Although it is fairly easy to change the resolution of CRT screens, many Museum exhibits now run on LCD screens, plasma screens and projectors, which have fixed resolutions.

For certain exhibits, it may be necessary to add additional hardware (e.g. extra i/o ports) or make changes to system settings. Please ensure that a member of the interactive media staff has been fully informed of any changes you plan to make before you start building your exhibit, as any additional hardware must be approved and purchased by the Museum.

Loan computers

Under certain circumstances, the museum will lend computers to computer exhibit contractors. These are generally exhibit computers which will later be installed on gallery. We lend these computers for two reasons:

- So you can do technical tests on your exhibit to make sure that it is compatible with the standard hardware and system spec.
- So we can evaluate your exhibit in the museum. Contractors who have been lent a computer should bring this computer in with their exhibit preloaded on it for evaluation. The computer will be returned as soon as evaluation is over.

Generally speaking, the Museum's policy is that it expects computer exhibit contractors to have their own PCs for development work. We **strongly** recommend that you do not

rely on using the loan computer as your primary development platform because:

- We will recall all loan equipment for installation at least a month before the exhibit is due to go live. The computer disk will then be completely reformatted ready for use on gallery. *(If you need to make any changes to the system setup for your exhibit, be sure to document them thoroughly – otherwise we'll install your exhibit on a standard system setup.)*
- Ninety-four per cent of computer exhibits in the Wellcome Wing needed changes (generally due to robustness issues) after installation. If you don't have your own PC, you'll have problems making any changes needed.

Problems with loan computers

If you have a technical problem with one of our loan computers, please **do not** attempt to fix it yourself. Contact the museum (Jo Saull) and we will arrange for it either to be fixed or replaced.

Don't be like one Wellcome Wing developer and sit up until 3 am trying to hack the NT password system or, like another, take the computer apart and leave bits rattling around inside it.

Appendix 1: Deliverables for a software exhibit

Software deliverables

All software should be supplied on a PC readable CD-ROM (two copies). You should write the name of the exhibit, plus the date and version on the actual CD – not just the jewel case.

The software should include:

- A fully automatic install version of the software which installs everything required to make the exhibit work. The installer should be flexible enough to allow the software to be installed in any directory or drive. See 'Exhibit design requirements' for more details.
- A complete set of ALL the files needed to run the exhibit software.
- Third-party files – such as Xtras and Quicktime – which should be fully documented, including information such as the version number, codecs used, shareware licensing information, the supplier and any serial numbers or authorisation codes.
- A list of files which must be installed and the appropriate directories.
- A complete set of all the source files needed to author the software, e.g. Director .dir files. We will also need a full specification of the software used to author the exhibit, including full version number, and a complete set of the intermediate files used for producing graphics and sounds. This includes unflattened Photoshop files, vector files and the meshes used for producing 3D graphics.
- The manuals in Word or rich-text format.

Manuals

Some of the things we're asking for in the manuals may seem a bit strange. The thing to remember is that the purpose of these manuals is to act as a complete reference to the exhibit which will still be useful in five years' time. The manual should be the only document you need to read to understand the exhibit.

Hardware documentation

This should include:

- Processor, memory and hard-disc specification.
- Graphics-card specification.
- Display specification.
- Input device(s) used.
- Any additional ports (e.g. parallel, serial) used by the exhibit.
- Any additional non-standard hardware components. Third-party components should be very clearly specified with part number, manufacturer and supplier.
- Circuit diagrams and schematics for any non-standard hardware.

System software documentation

It's particularly important that you highlight any changes made to the standard Museum specification, such as updating the drivers. Please specify:

- Full version name of the system software required by the exhibit.
- Display settings, i.e. resolution and number of colours needed.
- Any external fonts needed.
- Any other specific system settings needed, such as control-panel settings.
- Any registry settings or .ini files which need to be present.

Exhibit software documentation

Please include the following:

- Installation instructions for a clean install. Curiously, many contractors didn't supply these in their first draft – they are the most important bit of the manuals, so please make sure you include them.
- Brief description of all supplied files, including purpose, filetype, creator program and where the file should be installed.
- Description of the application, including a brief description of main routines (e.g. lingo scripts) within the program. This should be at a level that allows experienced programmers in the museum to maintain and update the software.
- List of settings which are museum settable – timeouts, cursor visibility etc.
- Licence numbers of any third-party software or extras, including version numbers, codecs, etc.
- Assignment of copyright, if applicable.
- If a novel interface has been used, you should document how to operate the exhibit using only the keyboard and mouse. You should also document how to test the hardware without having to run the whole exhibit (you may have to write a small test program to do this).

Hardware deliverables

This includes any hardware needed by the exhibit that has been procured by the software contractor as part of their contract. Any equipment supplied by the Museum, including all leads, must be returned (in the same condition as it was delivered to the software contractors).

Appendix 2: Robustness of common authoring tools

Windows NT 4.0 Service Pack 6: We have around 150 exhibits all running on this platform and can fairly confidently say that if all the drivers and hardware are set up correctly then the operating system will not crash. We have logged particular exhibits running consistently for over six weeks without problems.

Director 7.02: Similarly, we have around 130 exhibits written in Director 7.02, which are generally stable. Although it is possible to get Director to crash by using various external components (see below), we have only found one clearly identifiable potential problem – the 'Flash asset bug'. This is a well-known problem with Director 7.02. However, if you do not take steps to avoid it, *it is guaranteed to freeze your exhibit within a few hours, whether you are using any Flash components or not*. If you don't know about this bug, then do ring us and we'll tell you how to avoid it.

Director 8.0: We have five exhibits that run in Director 8.0. Of these, three have occasionally frozen, while the other two are fine. We have not been able to identify the cause of this problem.

Flash 4.0: We have about eight exhibits running in Flash 4.0, which seem to be reliable. However, do note that we have not been able to turn off the cursor in Flash 4.0 (although it may be possible using API calls).

Flash 5.0: We also have about eight exhibits written in Flash 5.0, which also seem to be reliable. The cursor can be switched off.

Visual Basic 6.0: We have two exhibits written in Visual Basic, which seem to be stable.

Quicktime 4.12: We have around ten exhibits that use Quicktime being called from Director. Of these, three or four consistently crash after a period of use. The amount of time it takes to crash seems to be dependent on how many times the Quicktime is played. We have tried comparing codecs and have not managed to narrow down the problem, as exhibits with the same codec display different behaviour. We suspect the problem could be in exhibits that use the Quicktime Sound mixing device to play both Director and Quicktime Sound. Also, sometimes Quicktime exhibits freeze – this can be corrected by altering the Quicktime graphics settings to 'use GDI'. If you use Quicktime in your exhibit, we suggest you do stringent robustness testing before making a final decision.