

Technical Introduction to Producing European Demos

This document gives the developer advice on how to create an effective working demo segment for inclusion on the European Demo Disc.

1. Sit, Think and Plan

Before you begin programming your demo there are a number of important questions that you should consider. For instance, does your game in its present form represent the final game well enough? How much game should you give to the user? Ideally you should include enough to get the user interested, experience some enjoyment, and with any luck have a craving for more.

As you're only showing a small excerpt of your game, include a full product information screen which describes all its features. You should also minimise disc usage as much as possible. There is a limit of approximately 40MB per game. The simplest way to do this is to reduce the number and size of DA tracks and pre-renders as they use huge amounts of disc space.

2. Check Standards

All demos must conform to a standard if they are to be included on a demo disc. This ensures that each game can run without adversely affecting any other program on the disc. If you fail to do this then your demo can't be used. All the time and effort in its creation would have been wasted.

Remember that the standard to use depends on the territory - there are different standards for SCEE, SCEI and SCEA. Each standard describes a number of technical and non- non-technical requirements that your demo should satisfy.

The SCEE standard specifies that all playable demos can be executed in one of two modes. 'Attract Mode' is essentially a rolling demo of your game and 'Interactive Mode', as the name suggests, is a playable demo. In each mode your demo must behave in the correct manner. For instance, in interactive mode your demo must exit if the user reaches the end or fails to complete your demo. Similarly, your demo should exit if the user simply stops playing it (in other words, no input is received) or explicitly wishes to exit using the controller (with any luck this won't be due to boredom).

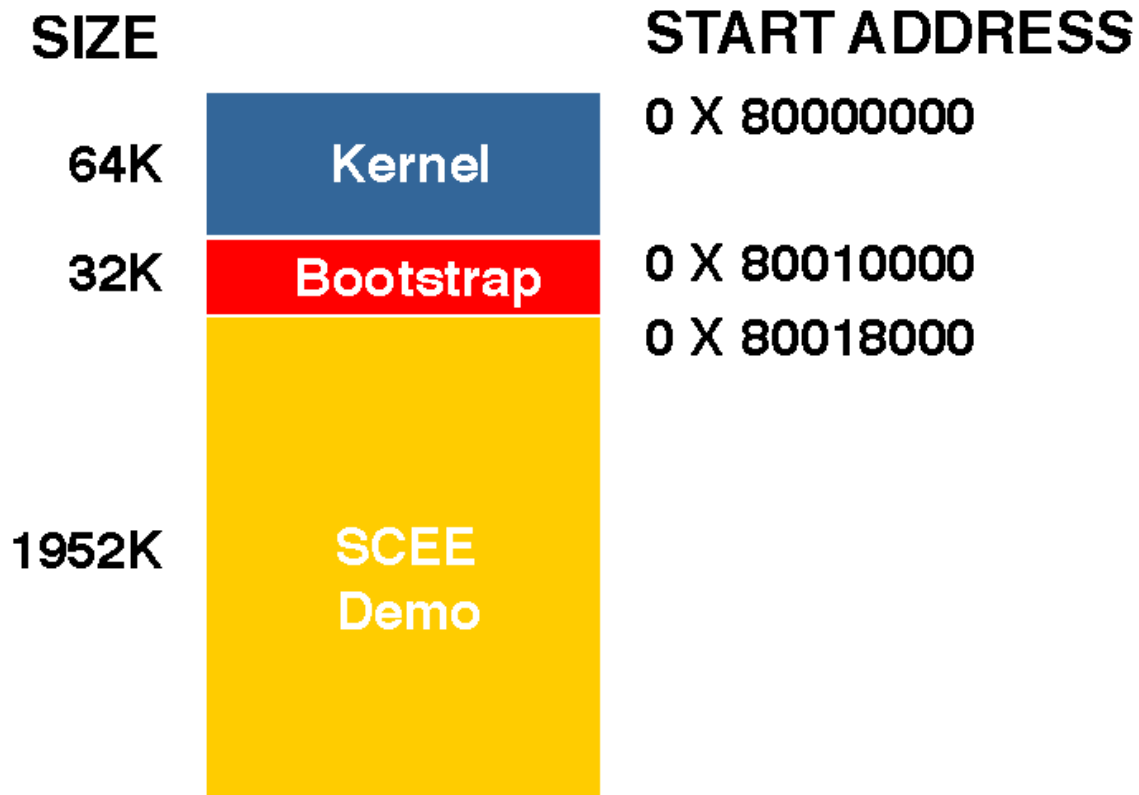
3. Know the Technical Issues

We can examine here some of the technical issues in creating a demo using the European standard. Full information is found in Word for Windows and text format on the accompanying floppy disk, in the file called Demog.doc or Demog.txt (English version) or DemoN.doc or DemoN.txt (Japanese version)

A large part of the programming involved is concerned with getting your game to run with the SCEE bootstrap program. There is a copy on the accompanying floppy disk.

The bootstrap, BS, is essentially the foundation of all SCEE demo discs. Its purpose is to load and execute all the demos on the disc.

As BS remains resident in memory while your demo runs, its memory area must not be corrupted. Simply ORG your demo above kernel and bootstrap memory, that is, 96K from the base of memory instead of the usual 64K. This figure shows the memory layout while a demo is running.



BS and your demo have a parent/child relationship. As a child your demo must be able to accept arguments from the parent and return to it in the same way as a typical C function would. Child programs are slightly different from the normal stand-alone programs you're used to creating. For instance, when compiling your demo you must use None2.lib instead of LibSN.lib. None2.lib library provides a cut down version of the start-up code which caters specifically for child programs.

As the SCEE standard requires a number of demos to be run one after the other without requiring a reset it is important that your game initialises and closes all sub-systems and events used in a clean manner. If you fail to do this the system may become unstable which may cause the entire demo disc to crash!

Your demo must also parse a number of arguments from the parent (that is to say, BS). This is achieved using the argc and argv mechanism that we're all used to. These arguments are used by your demo to find out what mode its been executed in, when to time-out and where its DA tracks are.

The way in which your demo is laid out on the CD is also very important. For instance, your demo should use its own directory. This makes the demo disc structure a lot cleaner and also minimises the possibility of file name clashes. Don't use the root, and make sure that you give your directory an original name. We don't want ten games, all using a directory called Demo!

As PlayStation CDs can have only approximately 40 directories and 30 files within each directory make sure you minimise the number of files as much as possible. Although this restriction doesn't seem too harsh, remember that we are producing multi-title demo discs containing as many as 15 demos. With many games on the disc the number of files and directories soon adds up. If the limit is exceeded then mysterious problems start to happen! Last of all, don't hard code the positions of your DA tracks, use the arguments to find their track numbers instead. This will enable our programmers to set up your demo for different discs.

Important

To operate on the European Demo Disc a program has to coexist with the demo bootstrap program, which is compiled to fit into a 32k area of reserved memory.

This bootstrap program has been produced with an extremely cut down version of the libraries to fit into the tight memory limits. This special library is called BS.LIB.

There is a compatibility problem with the CD file functions in this library and those found in LibCD version 3.5 and higher. Because there is no memory available to fix the special library, all program demos should use LibCD version 3.4. Please note that this is only a problem with the CD library. The other libraries can be later versions.

When time allows, a new version of BS.LIB will be produced and sent to developers. Until then, please use the LibCD version 3.4 when constructing your demo segment.

4. Submit

Once you've created your demo how can you submit it? It's simple. Send us a CD containing all the executables and data files, an interesting in-game screen shot, all the DA and streams. All DA should be supplied as files, don't embed them in DA tracks. Also provide streams in a form where they can be easily reconstructed. In other words, you should send a PC format CD not a PlayStation format CD.

A CCS and/or CTI file describing the directory structures and file types is also extremely useful, as is a short description of any other requirements needed to get your demo to work.

The submission form in the Technical Specification for European Demo Disc (Demog.doc or Demoj.doc) lists these and the other requirements for a correct submission.

5. Finally...

Although the process of programming and submitting a demo seems fairly obvious, about 90% of developers muck it up somewhere along the line. Please, please, please, give us all the necessary information! I don't want to phone you complaining that your demo doesn't work, and then have you explain "that's because you haven't renamed GAME.EXE to ESP.EXE"!

And that's all there is to it. All the sample code and documentation that you'll need for all demo standards is available on the accompanying floppy disk and on the SCEE BBS.

Good luck!

6. Documentation reference

The accompanying floppy disk contains the full code for the SCEE demo shell, with examples of use and various reference Readme files.

The main reference - Technical Specification for European Demo Disc - is called Demog.doc. It is the complete technical reference for the European demo disc. Demog.doc is in English. There is also a version in Japanese, called DemogN.doc. Both versions are found in the Docs sub-directory. There are also text-only versions of these documents in the same sub-directory.

Each other sub-directory of the floppy, such as 'Example' (tatoeba), contains a Readme file which describes the contents of that sub-directory. The original English versions of these files are also found in the Docs sub-directory, with the same names as the Japanese versions.

Written by Vince Diesi, SCEE Developer Support, May 1996.

Updated by Richard Milner, Special Projects, February 1997.

Updated by Dave Virapen, SCEE Developer Support, October 1997