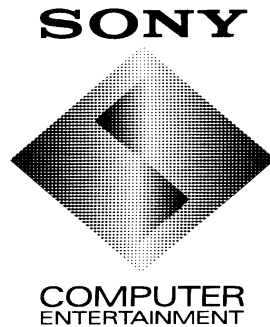


Technical Specification for European Demo Disc for PlayStation

Specification for Playable Game Segments



**Sony Computer Entertainment Europe
Waverley House, 7-12 Noel Street
West Soho, London W1**

Author: Allan J Murphy

25 April 1995

Updated 18 August 1995 (allan)

Updated 23 October 1995 (pholman)

Updated 10 November 1995 (Vince)

Updated 19 September 1996 (Richard Milner)

Updated 08 October 1997 (Dave Virapen)

Introduction

Overview

This document sets out the requirements for generating a playable game section to be included in the European version of the PlayStation demo disc. This document covers both the general requirements for the section and the technical details of building the section so that it is suitable for use in the demo disc. It is important that development teams working on these segments follow the technical guidelines laid down in this document, as failure to follow them will cause the demo disc not to work properly. If a segment does not follow the guidelines properly, it will not be included in the disc. Please make sure that the appropriate development staff are aware of the restrictions and guidelines set out below, otherwise the effort of building a demo segment will be wasted. This document also includes a questionnaire covering some general aspects of the playable segment, which we use to get some idea of how we can lay out the demo CD. It is important that you return this questionnaire to SCEE, because we have to plan allocation of space on the CD in advance.

The Demo Disc Concept

The demo disc concept, first introduced at PlayStation launch in Japan, provides an excellent mechanism for advertising and promoting both the PlayStation hardware platform and forthcoming or current releases from third party products. In Japan, the demo discs are not bundled with the PlayStation machine, but are just used as a rolling demo in retail outlets stocking the machine. Sony does the same thing in Europe, but also bundles the demo disc with the machine.

We also produce a variety of demo discs for magazines in Europe (Official PlayStation magazine), for bundling with games, and to send to registered PlayStation users. In the next year we expect to produce about 50 different titles containing games demos, which will be widely distributed across the SCEE territory to an expected total of 500,000 discs per month. These discs will be seen by dedicated users and their friends, as well as casual browsers in retail outlets of the hardware, software and magazines.

Having a playable section of your game in development on these discs means that even people who have not bought a PlayStation will see your game running in retail outlets; people who buy a PlayStation and no games with it will have only the demo disc to play with (so that you have the opportunity to get people hooked on your product). The demo disc will be updated on a roughly three-monthly basis, and any demos submitted will also be used on magazine cover-mount discs, so it is worth bearing in mind that you can submit playable sections at any time and expect them to be widely seen by users.

The European demo disc contains music videos, film trailers, pre-rendered trailers for games in development, some PlayStation technical demos, and around 10 playable sections of games which are forthcoming or current releases. The playable sections are intended for products whose release is imminent. If you want to advertise a product which is too early in development for a playable level to be built, you can submit a pre-rendered sequence advertising your game. Get in touch with SCEE about this option.

General Requirements

This section describes some of the general requirements for your playable section.

Playable Section Inclusion Process

- **Product at near-finished stage**

Your game must be at a stage where you are happy that it represents the final game well enough to be included; SCEE reserve the right to reject a game section because it is too unfinished (this is not unreasonable - an unfinished game demo does neither you nor us any favours). The level / section / track or whatever must be in a state where you can actually 'play' the game. This is really a qualitative judgement which will be made by both SCEE and the developer. In some cases it may be possible to submit a demo which is not actually playable, but provides some other suitable interaction for the user, and shows off the game in action. However, in this case, the demo must exhibit the abilities of the PlayStation - it should not be something that could easily be mistaken for 'pre-render'.

- **Developer requests game demo to be included on disc**

At this point, SCEE judge whether the game is suitable and ready to be included. There are issues other than quality to be considered here - there is no point in having a demo disc with eight football games on it, for example, or all the slots on the disc may already be taken. Your product must have had concept approval from SCEE (get in touch with your account manager at SCEE if this has not been done), but the product does not have to have passed QA.

- **Fill in questionnaire and return to SCEE**

This questionnaire gives us some idea of what your game is like. This helps us to work out how to lay out the disc, what other products we can and cannot include, and any special requirements for your playable section. A particular concern is streaming based games; these are usually too space consuming to include, unfortunately.

- **Submit early demo version**

The next stage is to submit an early version of the playable section. This is really just so that some testing can be done to check that the program that co-ordinates the whole demo disc when it is actually running works fine with your code. This is really just a first try at putting together the disc; at this stage the demo only needs to be functionally similar to the final thing, it does not need to be fully featured at all. It may even be NTSC if necessary (the European demo disc will of course be PAL). In other words, don't panic, just submit any recent demo configured as described in the technical section below (the configuration should not be a lengthy process at all, a couple of hours at the very most).

- **Demo production**

Now SCEE leave you alone to produce the playable section. Obviously, as it is in everyone's interest to produce the best demo disc possible, and a product being included on the disc is considered a major title, SCEE will be prepared to offer more technical (or otherwise) support than normal; the staff constructing the demo disc will be available to provide extra help if needed.

- **Final version testing and disc construction**

At this stage you submit your final version of the playable demo. Please check your demo with the technical checklist at the end of this document. It will most likely be necessary for the staff involved in the demo disc at SCEE and the development staff to exchange information in order to get the whole thing working (in other words, everyone makes mistakes, we may have to run through a couple of revisions of the SCEE demo disc driver program and your playable demo in order to get everything working). Unless your playable section is very small indeed, the best delivery medium is probably CD.

- **Mastering and distribution**

A final master disc is burnt, and then the discs are pressed and packaged with the PlayStation machine, and also distributed to retailers to provide a demo for the machine. Magazine cover-mounts are delivered monthly to the magazine publishers.

Production Requirements

- **Rolling demo**

The demo disc will run in two distinct modes. If there are no controllers attached at power on, the demo will go into attract mode; it will do the same thing if no controller keys are touched for a set amount of time. In this mode, the demo disc program will randomly select items to run, and so at some point your playable demo will be chosen and run. Because no-one is controlling the game, two things are important - firstly, that the playable section times out eventually and returns to the demo disc program (the time-out is specified below in the technical section), and secondly, that your playable section does something on its own if no-one is playing (even if it isn't that interesting - the main thing is that the screen is not static with nothing happening). It is also important that if someone picks up the pad and hits a key while your playable demo is in attract mode, they can actually control something or take over the game, or at least stop the demo and select another.

The other mode will be interactive mode, where someone is actually holding a pad and messing about with the demo. In this mode, the game demo should also terminate, under two circumstances. One is where the player completes a level / track / section (i.e. has reached the end of the playable section); the other is where the player has clearly put down the pad and wandered off to get their parents to buy a PlayStation, in which case the game should time out.

The essential thing is that the demo disc can be used as a rolling demo. **If your playable section does not time out one way or another, then the demo disc will stick in your demo (Although this might seem desirable to you, it will not please SCEE or the other developers who have playable demos on the disc !).** Your playable demo will be 'told' which mode it has been started in, and also how long its time-out is (this is so that we can configure the demo disc time-outs without having you rebuild everything every time we make a change).

- **Section Complete**

Other than the termination conditions described above (time outs etc.), the playable section should stop and return control to the demo disc program under these circumstances:

- *The player finishes the level / section / track*

As described above. This is the 'end' of the playable game. For example, on one Japanese demo disc, the playable demo of Toshinden allowed the player to fight one opponent, then returned control to the demo disc. This is to make it slightly unpleasant to play the same game a lot of times in a row, to encourage the consumer to buy the product, not just play the demo disc.

- *The player is 'killed' (loses all his lives, fails the level, or whatever)*

The playable demo should not allow 'continues' or let the player start a new game; the way the player starts a new game is to select your demo again from the demo disc program. If your game is extremely addictive, the player will get fed up waiting for it to load....and so may buy the actual game. So, in a Galaxians type game, for example, you might give the player 3 lives only; once they are gone, the demo disc program re-appears.

- *When the player hits the square 'select' key on the controller pad*

Currently, the demo disc is only going to support games which use 1 or 2 controller pads. It is important that the demo can be stopped by the player, and also important that all the playable demos are stopped the same way, and so the convention adopted is that the 'select' key stops the game and returns control to the demo disc program.

- **Full Product Information**

You may want to include an end screen with information about what is missing from the demo, and what features are in the full game. This is so that people don't think they have the whole game on the demo disc, and to encourage them to buy the final game.

- **Section Content**

To give a rough idea of how much game to include, some of the previous demo discs have included: 1 single circuit lap of Ridge Racer Revolution on the basic track; 1 fight (two bouts) against 1 random opponent in Tekken 2; an easy level of Tomb Raider, and so on. The objective is to give the user a good feeling of the game-play he can expect from the full game. You don't need to include a set percentage, just the features which make your game play differently (and better) than others.

- **Titles and Instructions**

Each playable demo will have one title screen (which will be displayed by the demo disc program to give the player some idea what kind of game it is) and one screen with a quick key guide (i.e. mini-instructions). These screens will be constructed by SCEE for you. To do this we need to know what the keys used in your demo are, and you may want to submit a screen shot of the demo for us to use in the title screen (otherwise we will grab one anyway).

- **Pre-render**

The playable section should not contain a lot of pre-render (unless this is part of the gameplay, of course - say, a game with a pre-rendered background with gameplay overlaid) - it is intended to give the user an idea of the game play. As a result, long intro or cut sequences should be left out (because they are both expensive in space and non-interactive). There should also not be a large number of static screens for the user to skip through.

- **Disc Usage**

You are encouraged to keep your CD usage to a minimum. Some games, by their nature, use a lot of space (for example, streaming based games), and so this is difficult to do. However, the space allocated to playable demos on the demo disc is 400 Mb. This works out about 40 Mb per game. This figure is flexible - if there is a game which only uses 3 Mb, then there is 37 Mb more to go around; however, a maximum limit of 40 Mb should be assumed. From this figure, it should be obvious that having 5 DA audio tracks for your demo is going to take up too much space. If you have DA audio, consider cutting it down or looping a smaller section. If you have a lot of streaming or data that is absolutely essential to your demo (over 40 Mb), get in touch with SCEE and we will try to arrange more space.

- **Loading & Program Start-up**

You should try to keep your loading and start-up time to a minimum. If your playable section takes some time to load information from CD, make sure that some feedback is given to the player, so that they know the demo has not crashed - for example, some simple music, a title screen, a game or company logo, or a loading message.

Technical Specification

Overview

The way the demo disc works from a technical point of view is as follows. When the PlayStation boots up at power on, the demo disc program (DDP) is loaded from CD into main RAM and runs. The core of this program lives between 0x80010000 and 0x80018000 in main RAM. In other words, the DDP core occupies 32K bytes of RAM right above the PlayStation kernel's 64K of RAM, which your code should not touch under any circumstances.

The main menu program then allows the user to choose from the variety of different things on the demo disc. It will load its own data in RAM above 0x80018000, but it will never depend on that data being there. Once the user has chosen your playable demo, your executable (.EXE) will be loaded from CD; the BSS segment will be cleared, and your program will be run. While your program is running, the DDP's core will still be in RAM - in other words, your program must not write to memory between 0x80001000 and 0x80018000, and your program must be linked with an org address of 0x80018000 or above.

Since you are probably used to not touching memory inside the kernel's space, it is anticipated that altering your code to avoid corrupting the DDP will probably only require you to relink your code with an org address 32K bytes higher than before. Hopefully, 32K of RAM less will not make too much of a difference to your game code. If you are pushed for RAM, you may consider checking the size of the stack you are using - the default is 32K, which is pretty big. The DDP keeps its own small stack inside its 32K, so you don't need to worry about corrupting its stack; your stack (in standard configuration) is in the top 32K of memory.

Essentially the DDP pages data it needs (including your code and data) in and out of the memory from 0x80018000 to 0x80200000. With the DDP, the main RAM looks like:

0x80000000 - 0x80010000	PlayStation Kernel RAM space
0x80010000 - 0x80018000	DDP core program and stack
0x80018000 - 0x80....	Your demo code and data
0x801f8000 - 0x801fffff	Your stack (assuming the default size and position assigned by libsn)

Your demo will have its own directory on the CD, which will contain the data files used by your demo and any other information it needs, with the exception of any DA audio, which will be a separate track on the CD. You are encouraged to keep the number of files you use to a minimum, as some other pieces of code on the disc may use SCE Japan's libcd.lib, which restricts the numbers of files on the CD to around 40 directories with about 30 files in each. If your application uses a number of files anywhere near this limit, you will probably find that it cannot be included on the disc because it will cause other programs to fail (libcd will start failing seeks to files if the total number of files on the disc is greater than libcd's limit).

The DDP runs your program using the kernel call Exec(). As such, your playable demo must be a standard PlayStation .EXE file on CD; the DDP needs the information in the first CD sector of the .EXE to figure out where to put your code, how much BSS to clear and where to jump to run the code. In order that your playable demo will return control to the DDP properly, you have to link your code with none2.lib, a replacement for libsn.lib which replaces the standard libsn start-up code. This start-up module does not clear the bss and set up the heap, because if your playable demo does this, it may foul the DDP. The only problem is that code linked with none2.lib will not run in its own right (obviously). The best way to develop the code is to work without none2.lib until the demo is ready, then test the version linked with none2.obj using a program which calls Load() and Exec() to execute your code, and

then checks that the sub-program returns properly (i.e. your demo is not fouling the parent program). A simple harness version of the DDP will be available from SCEE for your testing.

In addition to the memory and start-up restrictions described above, your program must do its initialisation, and close down, in the following way. This small fragment of code is essentially a harness for a program that will return properly to the DDP and also leave the various PlayStation subsystems in a usable state.

```
#ifdef LINKED_NONE2                /* If we have linked none2.obj */
int main( int argc, char** argv) /* DDP will pass argc, argv to you. */
#else
int main()                        /* Plain old main instead. */
#endif
{
    /* PSX doesn't like argc, argv in a main prog */

    ResetCallback();              /* Clear all of the CD callbacks. */
    CdInit();                     /* Re-initialise the CD subsystem. */
    PadInit(0);                   /* Initialise the pads. */
    ResetGraph(0);                /* Cold boot the GPU. */
    SetGraphDebug(0);             /* Turn GPU debugging off. */

    /* Now you can do any other startup you need to */
    /* And the code from here on is your own */

    .
    .
    .
    .
    .

    /* Little Johnny has been killed by the giant spiders from Mars, so.. */
    /* This is the end of the program now. */

    StopCallback();               /* Stop the CD callbacks. */
    PadStop();                    /* Stop pad reading. */
    ResetGraph(3);                /* GPU warm reset */
    return (0);                  /* This is necessary too. */
}
```

The DDP will pass the standard C variables `argc` and `argv` to your program. However, `argv` will not be a ragged array of characters, it will actually just point to an array of integers; `argc` will be four, as your demo will be passed four pieces of information. They are, the mode of the demo (attract or interactive), the time-out you will use to stop the playable demo, the track index of the game's first DA track, and number of DA tracks the game uses. If the game does not use DA arguments three and four can be ignored. If on the other hand your game does use DA please remember to inform us, using the questionnaire, the number of DA tracks used. The first track index, and number of tracks arguments will be used by the game to find out where its tracks have been placed on the final demo CD. We will ensure that the DA tracks are stored sequentially. Unlike the previous demo standard, the game will always be passed four arguments. This replaces the old standard which used a variable number of arguments depending on the number of DA tracks used.

Another slight change from the previous standard is the demo time-out. In the instance where the game is run as an interactive demo it should return if no input has been received for the number of seconds specified by the timeout. Alternatively if the game is run in attract mode (i.e. as a roling demo) the time-out represents the number of seconds before the demo should return. Please note that the time-out is always a positive integer.

Once the arguments have been processed use a `printf` to display them. By doing this we can check that the correct arguments are being passed to you're demo segment.

You can obtain the mode, time-out and DA information using the following code:

```
#define INTERACTIVE          0
#define ATTRACT              1

#ifdef LINKED_NONE2          /* If we have linked none2.obj */
int main( int argc, char** argv) /* DDP will pass argc, argv to you. */
/* Argv is actually the address of an */
/* array of integers. */

#else
int main()                  /* Plain old main instead. */
#endif                     /* PSX doesn't like argc, argv on its own */
{
    int timeout, mode;      /* Demo mode and timeout in seconds. */
    int start_track,       /* First DA track and number of DA tracks. */
    num_tracks;

    /* All the usual startup stuff etc. */

#ifdef LINKED_NONE2
    mode = ((int *)argv)[0];
    timeout = ((int *)argv)[1];
    start_track = ((int *)argv)[2];
    num_tracks = ((int *)argv)[3];
    printf("mode = %d, timeout = %d, start_track = %d, num_tracks = %d\n",
           mode,
           timeout,
           start_track,
           num_tracks);
#endif

    /* And on with the action */
}
```

Obviously, since your playable demo will be running on a European PlayStation, the demo must run in PAL mode (your PlayStation development kits are PAL-switchable - there is a hardware jumper, and also a software switch - please consult the example code on the SCEE support BBS, and contact technical support if you need more help). Since we are showing off the PAL machine's abilities in Europe, we will not accept letter-boxed NTSC products.

If you have any additional technical questions, or require further assistance from SCEE, please get in touch with your account manager, or the technical support group on technical matters.

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 fix into the tight memory limits. This special library is called BS.LIB (old libs 3.4 version) or new_bs.lib (compatible with all libs)

There is shouldn't be a compatibility problem with the CD file functions in this library and those found in LibCD version 3.5 and higher as there was with the old bs.lib.

If You have any problems with this new bs then let me know (Dave V) and I'll try to patch it if I have the time but I have tested it and it works fine here.

Software required

All the software and information you will need to create demo segments can be found on the SCEE BBS in the **Demo** section.

If you do not have a password for the BBS, please contact SCEE Developer Support, by phone or fax, to be assigned one.

Tech Support Contact Details

Phone: +44 171 447 1648

Fax: +44 171 390 4324

Email: ps_developer@interactive.sony.com

BBS: +44 171 390 4327 / 4328 / 4329

SCEE Demo Disc Playable Section Questionnaire

Game Title and Product Code:
Publisher:
Developer:
Project Supervisor and his contact details:

1 Style of Game (e.g. shoot-em-up, racing, etc.):
2 Type of Player Interaction (full control or otherwise):
3 Demo Complete Mechanism (e.g. 1 track, 1 level, 1 battle, etc.):
Controller / Peripheral Support (final game & this demo):

Estimated Demo Completion Date:
Size of EXE on CD (bytes):
Size of Data on CD (bytes):
Total Number of Files on CD:
Audio Type and Size in bytes on CD (and number of DA tracks):

Estimated Final Release Date:
Any Other Comments or Issues:

Playable Section Technical Checklist

- Demo orged at 0x80018000 or above
- Demo linked with none2.lib (and not libsn.lib as well)
- Demo does not write to memory in range 0x80010000 -> 0x80018000
- Demo does startup and closedown as specified above
- Demo reads demo mode properly
- Demo behaves according to the mode
- Demo can be quit in either mode with 'select' key at any point
- In interactive mode, demo plays properly; but times out when user does nothing
- In interactive mode, demo ignores timeout value
- In interactive mode, demo quits before you can play again
- In attract mode, demo quits after the number of seconds specified by the timeout parameter
- In attract mode, the player can either take over, or the demo will quit with the select key
- Demo closes down the subsystems as specified above
- Demo removes any callbacks and event handlers before closing down
- Demo is not in the middle of DMA when closing down
- Demo clears the reverb buffer in SPU RAM before closing down
- Rendered material is submitted in a form that can be easily reconstructed
 - (e.g. interleaving definition and source files; the raw STR on a CD is not easily retrievable)
- DA audio is submitted both as a DA track and also as a file on an XA disc
- Any specific CD layout is adequately described (e.g. files that must be sequential on CD)
- Demo has been tested with the harness BS from the SCEE BBS
- Demo will run more than 10 times in a row when run with the harness
- Demo does not depend on the state or contents of RAM (other than BSS), data cache, VRAM, SPU RAM, CD sector buffer, or I cache at startup
- Stack pointer has been set properly in your .EXE header to the value that you require (by you)
- Printf() inserted at the start of main() showing arguments passed to the demo
 - Printf() inserted at the end of main() to mark end of demo execution