

# CD-i Development in the 21st Century

Date: 2021-02-25

Author: Luc Rooijackers

## 0. Contents

<b>1. INTRODUCTION</b> .....	<b>3</b>
1.1. ABOUT THE AUTHOR .....	3
1.2. HOME-BREW CD-I TITLES.....	3
1.3. CD-I DEVELOPMENT.....	4
1.4. READER PREREQUISITES .....	4
<b>2. SPECIFICATIONS</b> .....	<b>5</b>
2.1. CD-I FULL FUNCTIONAL SPECIFICATION .....	6
2.2. OS-9 TECHNICAL MANUAL .....	8
2.3. OS-9 TECHNICAL I/O MANUAL .....	10
<b>3. SYSTEM DOCUMENTATION</b> .....	<b>11</b>
3.1. PIM TECHNICAL NOTES .....	11
3.2. OTHER TECHNICAL NOTES.....	12
3.3. THE INTERACTIVE ENGINEER.....	12
3.4. TECHNICAL DOCUMENTATION FOR CDI 605 USERS .....	13
3.5. SCC68070 USER MANUALS .....	14
3.6. OS-9 INSIGHTS .....	16
<b>4. TOOL DOCUMENTATION</b> .....	<b>17</b>
4.1. USING PROFESSIONAL OS-9.....	17
4.2. USING OS-9/INTERNET .....	17
4.3. OS-9 C COMPILER MANUALS.....	18
4.4. OS-9 ASSEMBLER/LINKER MANUALS .....	19
4.5. CD-I SYSTEM LIBRARY TECHNICAL REFERENCE MANUAL .....	20
4.6. OS-9 IFF DOCUMENTATION.....	20
4.7. DISC BUILDING AND EMULATION DOCUMENTATION .....	21
4.8. OS-9 VERSION 2.4 RELEASE NOTES .....	21
4.9. OS-9 C COMPILER VERSION 3.2 RELEASE NOTES.....	21
<b>5. DEVELOPMENT TOOLS</b> .....	<b>22</b>
5.1. TOOLKITS .....	22
5.2. COMPILERS AND ASSEMBLERS .....	23
5.3. ASSET CONVERSION .....	24
5.4. DISC BUILDING AND EMULATION .....	25

<b>6. SUPPORT MATERIAL</b> .....	<b>26</b>
6.1. OS-9 SYSTEM DISK .....	26
6.2. 605 DRIVER UPDATE DISK .....	26
6.3. BASE-CASE TOOLS DISK .....	27
6.4. FINAL PRODUCTION DELIVERY DISC .....	27
6.5. OPTIMAGE PUBLIC DOMAIN SOFTWARE DISKS .....	28
6.6. CDISC 4 FREWARE CD-I .....	28
<b>7. VERSION HISTORY</b> .....	<b>29</b>

## 1. Introduction

The *Compact Disc Interactive* system (CD-i or CD-I for short) is a CD-based interactive multimedia playback system jointly developed by Philips and Sony. CD-i player units were commercially sold from 1991 until late 1998; the last commercial CD-i title came out in 1999. A more extensive description can be found here: [https://en.wikipedia.org/wiki/Philips\\_CD-i](https://en.wikipedia.org/wiki/Philips_CD-i).

Developing software for the CD-i system was never easy; documentation and development tools were only available under strict non-disclosure and license agreements. This means that it is now extremely hard to find those resources, seriously hindering new home-brew development.

This document attempts to provide a general introduction to the aspiring CD-i home-brew developer, with special attention to obtaining the necessary development resources.

### 1.1. About the author

From 1993 to 1999 the author of this document developed CD-i software at *SPC Vision* (also known as *The Vision Factory*) which was based in the Netherlands. For the consumer market the company developed a number of well-known game titles and several reference titles.

Where relevant I will mention the specific document and tool versions that we used at SPC. Most of these (or very close ones) are available online or obtainable from a suitable source.

Years ago I wrote a short history of SPC Vision that is still online at <http://cdidev.tripod.com>.

### 1.2. Home-brew CD-i titles

The first major home-brew CD-i title, *Frog Feast*, was developed by Charles Doty in 2005 and shows that is still possible to become a CD-i developer, even starting without any system-specific technical knowledge. Charles documented some of his progress on his *CD-I Development Central* website and also released a simplified version of the Frog Feast sources there.

The original of that website is no longer online; a mirror of the last version can be found at <http://icdia.co.uk/cdidevresearch/> and includes a simplified Frog Feast sources download.

The system initialization part of Frog Feast was based on an extended version of the *CD-i Stub* program from CD-i Fan, source can be downloaded from <http://cdiemu.org/cdistub/>.

Based on the Frog Feast sources, 2018 saw the release of another home-brew CD-i title, *Super Quartet*, by online user *gameblabla*. Disc image files and full sources are on *The world of CD-i* at <https://www.theworldofcdi.com/open-source/CD-i-titles-development/super-quartet/>.

*Gameblabla* has collected many related files on his website <https://gameblabla.nl/files/cdi/>.

### 1.3. CD-i development

Originally CD-i development was a cumbersome process requiring special (very expensive) hardware. Nowadays the special hardware can for most purposes be replaced with software, in particular CD-i emulation software (see below), which can make development easier and faster. As in the past, however, final validation and testing of a CD-i title requires burning an actual CD and using it with the actual hardware.

Along with the aging field of hardware CD-i players there are now also software CD-i players such as *CD-i Emulator* (<http://www.cdiemu.org>), *MAME* (<http://mamedev.org>) and *CD-ice*. These programs emulate in software what the original CD-i players did in hardware; they (except for CD-ice) run the original CD-i system software ROMs inside the emulated machine.

CD-ice was the first software CD-i player. It was written by Pete Dabs who used his source code for the CD-i game *Rise of the Robots* to re-implement required parts of the CD-i system software from scratch. Only demo versions of CD-ice were ever released and the project has no current online presence, although the demo's can still be downloaded from various websites. However, exactly the fact that CD-ice does *not* use the original CD-i system software makes it somewhat unsuitable for CD-i development, because of the compatibility risks involved.

Software CD-i players typically run CD-i discs from image files, thereby eliminating the costliest and most cumbersome part of the original development process (getting CD-i hardware to treat a CD-i disc image file as an actual CD-i disc, which required a special authoring CD-i player connected to a hardware CD emulation device). Software players typically also have deployment and debugging features such as the ability to load software, inspect memory, perform traces and single-step the entire machine that contemporary CD-i developers could only dream of.

However, most of the original CD-i software development tools are nowadays showing their age, leading to special requirements on the host platform or necessitating the use of emulation to run those programs. To offset this, typical PC hardware is much faster today than it was at the times which can make CD-i development a breeze compared to what it was originally.

### 1.4. Reader prerequisites

In order to understand this document, some knowledge of software development tools and processes is required, but in-depth technical knowledge is not necessary.

If you want to actually develop for the CD-i system using the resources described in this document, you will need some knowledge of 68000 assembly language and/or the C or C++ programming languages. Some knowledge of digitized images, audio and optionally video can also be helpful as producing or at least converting such assets is usually a necessary part of the CD-i authoring process.

## 2. Specifications

The CD-i system is specified in the *CD-i Full Functional Specification*, usually called the *Green Book* and abbreviated *FFGB*. The specification is essentially a software API standard; there are no hardware interface descriptions (except the basic processor architecture) in the Green Book or its references. The specified audio and video data formats require hardware assistance to correctly render (at least they did when the specification was initially developed), but no actual device registers are specified in the Green Book. Several different hardware implementations exist and low-level device details often vary between versions of those implementations.

Every CD-i player runs the CD-RTOS 1.1 operating system, incorporating most of the OS-9/68000 version 2.4 embedded operating system from Microware. As described by the Green Book:

FFGB Chapter VI, *Program-related Data Representations*:

*The program related data is the data to be read and processed by a microprocessor of the 68000 MPU family.*

*The 68000 MPU family includes several similar, but not identical, microprocessors. To be compatible with all players, the object code on a CD-I disc must be written for the microprocessor with features which are common to all (e.g., to the 68000 itself). This instruction set is defined in the 'M68000 16/32-Bit Microprocessor Programmer's Reference Manual', published by Motorola (document number M68000UM).*

FFGB Chapter VII, *CD-RTOS (Compact Disc Real-time Operating System)*:

*All application programs issue requests for system services through the CD-RTOS kernel.*

*The "OS-9 System Calls" part of A VII.1 contains detailed explanations of all the service requests available in CD-RTOS.*

FFGB Appendix VII.1, *CD-RTOS Technical Manual*:

*This appendix refers to the OS-9/68000 Technical Manual<sup>4</sup>. Only certain parts of the OS-9/68000 Technical Manual are pertinent to the CD-RTOS Base Case System.*

*All of these chapters of the OS-9/68000 Technical Manual are completely a part of the CD-RTOS Base Case System with the exception of all but one of the file managers referred to in the OS-9 Input/Output System chapter.*

<sup>4</sup> *This manual is revision J and reflects version 2.4 of the OS-9/68000 4 Operating System.*

All development for the CD-i system must ultimately be based on the Green Book and the OS-9 technical manuals it incorporates by reference, so let's take a deeper look at these documents.

## 2.1. CD-i Full Functional Specification

The first version of the Green Book was released in the late eighties by Philips and Sony. It was updated and clarified several times and significantly extended once by the addition of the Full Motion Extension for MPEG audio/video playback, resulting in the final May 1994 version.

The Green Book was originally not freely available and had to be licensed from Philips under non-disclosure agreements, but since at least 2011 the May 1994 version is available here:

[https://www.lscdweb.com/data/downloadables/2/8/cdi\\_may94\\_r2.pdf](https://www.lscdweb.com/data/downloadables/2/8/cdi_may94_r2.pdf)

This version of the Green Book and several related documents are also available from the ICDIA website at <http://www.icdia.co.uk/docs/funcspeg.html>.

The May 1994 version of the Green Book contains the following sections:

TOC	Table of Contents
Chapter I	Introduction
Chapter II	CD-I Disc Format
Chapter III	Data Retrieval Structure
Chapter IV	Audio Data Representation
Chapter V	Video Real Time Data Representation
Chapter VI	Program Related Data Representation
Chapter VII	Compact Disc Real Time Operating System
Chapter VIII	Base Case System
Chapter IX	Full Motion Extension
Appendix I	Glossary of Terms
Appendix II	CD-I Disc Format/Subheader Values
Appendix V	Video Real Time Data Representation
	V.1 CCIR Level Assignment
	V.2 Guidelines for 525/625 Image Interchange
Appendix VII	Compact Disc Real Time Operating System
	VII.1 CD-RTOS Technical Manual
	VII.2 CD-I Peripherals

These chapters and appendices specify the low-level interfaces available to CD-i titles, including all the nitty-gritty details of the CD-i disc format and the system-provided services (system calls). The CD-i specific system calls are specified in Chapter VII using their 68000 machine language bindings; the standard OS-9 system calls are specified in the referenced *OS-9 Technical Manual*.

Chapters I to VII specify the functionality of a CD-i player including a number of possible extensions of the basic functionality (e.g., additional video modes or resolutions).

Chapter VIII specifies the so-called *Base Case System* which has the minimum functionality required for all CD-i players. It specifies a 16-bit data bus version of a 68000 family micro-processor with two RAM banks of minimum 512KB each and an X-Y pointing device as the only guaranteed human input device. Although a keyboard API is specified in the Green Book, no Base Case CD-i player is required to have a keyboard; in fact CD-i keyboard peripherals were not widely available for consumer players until the introduction of the *CD-i KeyControl* in 1996.

Chapter IX specifies the *Full Motion Extension* of the Base Case System; the MPEG data formats and system services described in this chapter are not available on all CD-i players. Most of the Philips players could be upgraded to the extended version by inserting a *Digital Video Cartridge* or *DVC* which was usually sold separately, later player models have it built-in.

In released CD-i players, the Full Motion Extension is normally accompanied by a 1MB memory extension as specified by a player implementation recommendation that is available from the ICDIA website at <http://www.icdia.co.uk/docs/funcspeg.html>.

Note that no current software CD-i player fully supports the Full Motion Extension. There are various reasons for this, including MPEG patent restrictions and development complexity.

The original paper Green Book comes in a thick three-ring dark green binder on 6" x 8" paper.

At SPC we worked for many years with bound A4 copies of the October 1992 version and a ring-bound A4 copy of the March 1993 version of Chapter IX, both of which differ in only minor details from the final May 1994 version of the Green Book.

Later we acquired two "original" Green Book binders containing the September 1990 version, but we never used those during development.

It is important for a CD-i developer to have a basic understanding of the available audio and video data formats and the available CD-RTOS system services, but completely reading and comprehending the Green Book is usually not required.



## 2.2. OS-9 Technical Manual

The Green Book directly describes the CD-i specific services of the CD-RTOS operating system, but refers to the *OS-9 Technical Manual*, version 2.4, revision J for the “regular” OS-9 services.

The OS-9 Technical Manual contains two main sections, *OS-9 Technical Overview* and *OS-9 System Calls*. Each section contains a number of chapters, some of which are specified by the Green Book as not applicable to CD-RTOS:

### *OS-9 Technical Overview*

- Chapter 1 - System Overview
- Chapter 2 - The Kernel (*except Sysgo and Extension Modules*)
- Chapter 3 - OS-9 Input/Output System
- Chapter 4 - Interprocess Communications
- Chapter 5 - Trap Handlers
- Chapter 6 - The Math Module
- ~~Chapter 7 - RBF File System~~ (*not a part of CD-RTOS*)
- Appendix A - Example Code (*except Sysgo and RBF/SCF/SBF device descriptor*)
- Appendix B - Path Descriptors and Device Descriptors (*only Pipeman path descriptor*)
- Error Codes

### *OS-9 System Calls*

- Chapter 1 - User-State System Calls
- Chapter 2 - I/O System Calls
- Chapter 3 - System-State System Calls

The OS-9 documentation was typically not freely available and had to be purchased from Microware. However, Revision K of the *OS-9 Technical Manual* can be downloaded from:

<http://www.icdia.co.uk/microware/tech/>

Revision J, differently formatted but with mostly identical content, is part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>



The OS-9 Technical Manual specifies many low-level interfaces of the OS-9 operating system, including all the provided system services (system calls).

The *OS-9 Technical Overview* section contains a high-level technical overview of the OS-9 system; part of it is recommended reading for any CD-i developer.

Chapters 1 to 3 describe the basic components of the OS-9 system and cover subjects such as memory organization, process structure, system call concepts and the I/O philosophy. It is required reading for any low-level CD-i developer.

Chapters 4 to 6 describe more esoteric parts of the OS-9 system; a CD-i developer will not typically need these although they are fully a part of CD-RTOS.

Chapter 7 describes the RBF File System that is not part of the CD-i Base Case system but available on all CD-i players with floppy drives; a CD-i developer would only need this when using OS-9 formatted floppies.

Note: Using PC formatted floppies requires a different file manager, PCF, that is not documented in standard OS-9 documentation but conceptually equivalent to RBF.

Appendices A & B are mostly not part of CD-RTOS and typically not relevant for CD-i developers.

The list of Error Codes is extremely relevant for low-level debugging, but there is also a command-line tool “*er*” that can be used for looking up error numbers returned by system calls.

The *OS-9 System Calls* section contains the nitty-gritty details of all OS-9 system calls; they are specified using their 68000 machine language bindings.

Chapters 1 & 2 are relevant for CD-i development, although CD-i applications typically use the I/O system calls in very stylized ways to call CD-RTOS services as specified in the Green Book.

Chapter 3 is not typically relevant for a CD-i developer because it requires using system state which is not the default environment of a CD-i application. However, some OS-9 services are only available using system state and it is possible to enter this state from a CD-i application so these services can be used when necessary.

Some shortcuts to using system state are possible by exploiting the fact that CD-RTOS does not utilize memory protection and hence all OS-9 system data structures are directly accessible to a CD-i application program, although doing this was generally frowned upon by Philips.

At SPC we worked with Revision J of this manual in three-ring binders with 6” x 8” paper.

### 2.3. OS-9 Technical I/O Manual

The introduction of the *OS-9 Technical Manual* contains the following sentence:

The ***OS-9 Technical Manual*** is designed for you to use conjunction with the ***OS-9 Technical I/O Manual***.

The OS-9 Technical I/O Manual contains the following chapters:

Chapter 1 - The OS-9 Unified Input/Output System

~~Chapter 2 - Random Block File Manager (RBF) (*not a part of CD-RTOS*)~~

Chapter 3 - Sequential Character File Manager (SCF)

~~Chapter 4 - Sequential Block File Manager (SBF) (*not a part of CD-RTOS*)~~

As with the OS-9 Technical Manual, this manual had to be purchased from Microware. However, Revision C of the *OS-9 Technical I/O Manual* can be downloaded from:

<http://www.icdia.co.uk/microware/techio/>

Revision C, differently formatted but with identical content, is also part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>

Knowledge of the OS-9 Technical I/O Manual is not necessary for most CD-i development, and half of it is not pertinent to CD-RTOS anyway. But if you need them, the very low-level details are there and are fully supported on all CD-i systems.

At SPC we for several years worked without this manual; later we acquired Revision J in a three-ring binder with 6" x 8" paper.

### 3. System documentation

In addition to the CD-i specifications as described in the previous section, there exists a wealth of additional technical documentation for the CD-i and OS-9 systems. This section describes some of those documents.

#### 3.1. PIM Technical Notes

This is a collection of near a hundred small notes from Philips Interactive Media about CD-i development and CD-i player quirks. A small subset of these documents is currently available online on the ICDIA website but is expected to grow over time:

<http://www.icdia.co.uk/notes/>

Many of these notes are not extremely important for CD-i development but some offer essential insights. Skimming at least the abstracts of these Technical Notes is highly recommended.

Any serious low-level CD-i developer should be familiar with at least the following notes:

- TN #041 - CD-I Application Library/Trap Handler Usage
- TN #046 - Restrictions on the Use of Load CLUT Color Instructions
- TN #047 - Real-Time Code Loading
- TN #049 - Performance Considerations in CD-RTOS
- TN #057.1 - Initializing a Player's Configurable Parameters (revised)
- TN #061 - Software Sprites in CD-I
- TN #062 - Living with Reality: Remaining Bugs in CD-RTOS 1.1
- TN #063 - Display Synchronization in CD-I
- TN #069 - Technical Status of the Philips 910 CD-I Player
- TN #076 - Differences between the Philips 18x and 605 Development Platforms
- TN #085.1 - Ensuring Title Compatibility Across Players
- TN #087 - Memory Allocation in CD-RTOS
- TN #101 - Extension Memory in CD-i

A number of Philips IMS TSA Application Notes are included in the PIM Technical Notes series; these are more like tutorials and are much longer than the typical Technical Note. TSA-001 to TSA-010 have been assigned Technical Note numbers #091 to #100, some notable ones:

- TN #092 - Audio Considerations in CD-I (*TSA-002*)
- TN #093 - Image Formats for Compatibility (*TSA-003*)
- TN #094 - Various Time Bases in CD-I (*TSA-004*)
- TN #096 - Emulator and Emulation (*TSA-006*)
- TN #097 - The Full Motion System for CD-I (*TSA-007*)
- TN #098 - Introduction to Programming the FMV System (*TSA-008*)

At SPC we worked with two large hardcopy binders containing A4 copies of these notes.

### 3.2. Other technical notes

Philips often distributed technical notes that were not formally a part of the PIM series described above. A number of these notes is available from the ICDIA website at:

<http://www.icdia.co.uk/docs/>

The following ones are well worth reading:

- MONO II Player Status
- CD-i Nullmodem Cable Specification
- Application Note 'CD-i Ready' disc
- Technical Info Pointing Devices
- Technical Info Keyboards

At SPC we used all of these and some more, often related to specific technical issues.

### 3.3. The Interactive Engineer

*The Interactive Engineer* was a bimonthly publication from PIMC in Belgium for CD-i developers. It ran for several years; the issues from March/July 1996 onwards are available on ICDIA at:

<http://www.icdia.co.uk/iengineer/>

The subjects varied wildly and ranged from announcements of and commercials for (new) services through authoring software issues to detailed technical issues. Some articles about important subjects for the low-level CD-i developer:

Direct Line Control Tables access:

- Access to the Line Control Tables*, Volume 5, No. 3, March/July 1996
- Direct Line Control Table access*, Volume 5, No. 4, July/August 1996

Direct soundmap access:

- Using soundmaps in Balboa*, Volume 5, No. 3, March/July 1996
- More on Balboa and soundmaps*, Volume 5, No. 4, July/August 1996

CD-i player types:

- CD-i player types*, Volume 5, No. 5, September/October 1996
- OOPS*, Volume 5, No. 6, November/December 1996

DV cartridge types and issues:

- Which DV cartridge is in a 605?*, Volume 5, No. 5, September/October 1996
- The CSD file and DV cartridges*, Volume 5, No. 5, September/October 1996
- DV plane shift - 605 IMPEG DVC*, Volume 5, No. 6, November/December 1996

Full disclosure: Three of the articles above were written by me.

### 3.4. Technical Documentation for CDI 605(T) users

This document contains technical information about the CDI 6XX series of Philips professional players, specifically focused on the CDI 605 and 605T players.

Two versions of this document exist, one for the CDI 605 and one for the newer CDI 605T player with software release 1.3 (which was also available for original CDI 605 players). Both versions are available on ICDIA at [www.icdia.co.uk/techdocs/](http://www.icdia.co.uk/techdocs/).

Both versions have the same chapters but their version numbers differ:

Chapter	Version (605)	Version (605T)
1.0 Contents	1.0	1.1
2.0 Introduction	1.0	1.1
3.0 Csd/csdinit	1.1	1.2
4.0 Download routine	1.0	1.1
5.0 Initial process sysgo	1.1	1.2
6.0 Load application and system button	1.1	1.2
7.0 Technical characteristics of the player	1.0	1.1
8.0 RS232 port specification	1.0	1.1
9.0 Input devices	1.0	1.1
10.0 Floppy specification	1.0	1.1
11.0 Extension	1.0	1.1
12.0 CD-I 605(T) tools menu	1.0	1.1
13.0 Application notes ( <i>empty</i> )	1.0	1.1

In most cases the differences are minor or nonexistent apart from section version, header and product name updates.

Most of this information is specifically intended for the development of applications using the Philips professional players, but some of it is applicable to all Philips-designed CD-i players. This includes the generic startup procedures of chapters 3.0, 4.0 and 5.0, the miniDIN port specifications of chapter 8.0 and the input device protocol specifications of chapter 9.0.

The documents *Technical Info Pointing Devices* and *Technical Info Keyboards* described in an earlier section appear to be partial compilations of information from chapters 8.0 and 9.0 of these documents intended for developers of third-party pointing devices and keyboards.

Many of the chapters contain detailed hardware specifications to the level of chip types and their memory addresses as well as I/O signal specifications and connector pin layouts. This information is all specific to the 605(T) player and its extension board and is not useful for developing applications intended to be usable on consumer CD-i players.

At SPC we specifically used the Download routine (chapter 4.0) for debugging on consumer players and the CD-I 605(T) tools menu (chapter 12.0) for debugging on the CD-i 605 player, the other information was mostly useful background for the reason described above.

### 3.5. Processor User Manuals

The Green book specifies a 16-bit bus version of the 68000 processor family as the main processor of a CD-i player, but notes in FFGB VI.2.2 *Instruction Set Summary*:

*The 68000 MPU family includes several similar, but not identical, microprocessors. To be compatible with all players, the object code on a CD-I disc must be written for the microprocessor with features which are common to all (e.g., to the 68000 itself). This instruction set is defined in the 'M68000 16/32-Bit Microprocessor Programmer's Reference Manual' published by Motorola (document number M68000UM).*

*One incompatibility exists in the 68000 family. The 68000 MPU instruction set allows a 'move from sr' instruction to be executed by user state programs. Other processors regard this as a privilege violation and provide the 'move from ccr' instruction instead. Unfortunately, 'move from ccr' is not a legal 68000 instruction.*

*To overcome this difference, application programs may not use the 'move from sr' instruction. Instead, the 'move from ccr' instruction must be used. This instruction is emulated by the CD-RTOS kernel on players which use the 68000 (or other) MPU. Since emulation is slow, it is recommended that the 'scc' instruction be used to save the state of the condition code register.*

*CD-RTOS provides a service request (F\$SetSys) to examine system global variables. One of these variables (D\_MPUTyp) is initialized by the system to contain the MPU actually present on a particular CD-I player. Programs wishing to exploit extended MPU features may inspect this variable and use processor specific routines to take advantage of the native instruction set. However, it is always mandatory to provide functionally equivalent routines which use only the instructions and addressing modes available on a 68000 MPU. The application must take care of the compatibility problems resulting from the difference in performance when executed on the Base Case System.*

All Philips CD-i players except the CDI 370 have used a SCC68070 chip, which is a Philips reimplement of the 68000 architecture by Signetics (a.k.a. Philips Semiconductors). The Sony IVO players use an actual Motorola 68000 chip while the Kyocera player uses a Toshiba TMP68305 chip which has a Motorola 68000 core; the I2M CD-i boards use a Motorola 68341 chip which has a CPU32 core (basically a Motorola 68020 with some restrictions).

One of the problems with these chip differences is that the relative instruction timings differ widely between the Motorola and Signetics implementations, although their “average” performance is somewhat comparable given appropriate chip clock frequencies (the 68070 needs about twice the clock frequency of a Motorola chip, due to a different machine cycle).

Early on at SPC we acquired chip documentation of the SCC68070 and we used the specified instruction timings from this to better optimize our assembly language (esp. the pixel blitting code generated both manually and by the sprite compiler) for the 68070 as the most common CD-i main processor. Our code would still run on the others, it just wouldn't be as optimized.

The actual manuals we used were two bound A5 booklets titled:

*SCC68070 User Manual 1991, Part 1 – Hardware*  
*SCC68070 User Manual 1990, Part 2 – Software*

Digital editions of some versions of these documents are available online as indicated below, although some are incomplete or contain small errors in certain device register specifications.

Documentation for all the above-mentioned main processor chips is available on ICDIA at [www.icdia.co.uk/docs/](http://www.icdia.co.uk/docs/), but note the following from FFGB VI.2.2 *Instruction Set Summary*:

*The 68000 MPU family includes several similar, but not identical, microprocessors. To be compatible with all players, the object code on a CD-I disc must be written for the microprocessor with features which are common to all (e.g., to the 68000 itself). This instruction set is defined in the 'M68000 16/32-Bit Microprocessor Programmer's Reference Manual', published by Motorola (document number M68000UM).*

*One incompatibility exists in the 68000 family. The 68000 MPU instruction set allows a 'move from sr' instruction to be executed by user state programs. Other processors regard this as a privilege violation and provide the 'move from ccr' instruction instead. Unfortunately, 'move from ccr' is not a legal 68000 instruction.*

*To overcome this difference, application programs may not use the 'move from sr' instruction. Instead, the 'move from ccr' instruction must be used. This instruction is emulated by the CD-RTOS kernel on players which use the 68000 (or other) MPU. Since emulation is slow, it is recommended that the 'scc' instruction be used to save the state of the condition code register.*

*CD-RTOS provides a service request (F\$SetSys) to examine system global variables. One of these variables (D\_MPUPTyp) is initialized by the system to contain the MPU actually present on a particular CD-I player. Programs wishing to exploit extended MPU features may inspect this variable and use processor specific routines to take advantage of the native instruction set.*

A typical CD-i developer does not need to know the detailed instruction timing details.

Using the rest of the contents of these documents is not allowed for CD-i development as it mostly describes the on-chip peripherals of the various processor chips and controlling these is the exclusive purview of the system-supplied device drivers.

### 3.6. OS-9 Insights

The book *OS-9 Insights* by Peter Dibble contains a wealth of information about OS-9, including extensive high-level discussion of its internals. It includes sources for a number of OS-9 programs, file managers and drivers, including a prototype version of the “PC File Manager” that is included in all CD-i players with floppy drives.

The second edition of this book describes OS-9/68000 version 2.4 and as such is almost fully applicable to CD-RTOS. Paper copies are very hard to find nowadays since the book has long been out of print. A digital version can be downloaded from:

<http://rab.ict.pwr.wroc.pl/dydaktyka/supwa/os9/MWARE/os9insights/Books/ed2/>

The 3.0 edition of this book was updated for OS-9/68000 version 3.0 and thus contains some material not applicable to CD-RTOS, but most of the high-level descriptions are still relevant. It is still possible to find paper copies but a digital version can be downloaded from:

<http://rab.ict.pwr.wroc.pl/dydaktyka/supwa/os9/MWARE/os9insights/Books/ed3/>

We never had this book at SPC while developing for CD-i, I acquired it personally much later.

### 3.7. The OS-9 Guru

The book *The OS-9 Guru* by Paul S. Dayan is the authoritative reference for the inner workings of OS-9 2.4. This book gives the inside information about how the operating system works, in complete and authoritative detail. And it describes how the features of the operating system should be used in applications programs.

This book was written for OS-9 version 2.4 (equivalent to CD-RTOS), but also describes changes and improvements to the operating system from version 2.2 onwards.

The book been out of print for many years but can occasionally be found for sale online.

We also never had this book at SPC and though it provides an amazing amount of low-level insight the information is not generally needed for CD-i development.



## 4. Tool documentation

This section describes some manuals and documents that describe OS-9 and CD-i developer tools; some of the tools themselves are described in later sections.

### 4.1. Using Professional OS-9

This manual describes using OS-9 as a general-purpose operating system, mostly from the command line. The shell module and most of the commands described in this manual are not normally available on consumer CD-i players.

Some professional CD-i players include the shell and many commands in ROM, and there is also an *OS-9 System Disk* floppy that contains most of them. The floppy can only be used with professional CD-i players that include a floppy drive and requires a serial terminal.

The shell and some commands are also included on the MediaMogul discs that are intended for use on the CDI 180 and CDI 605 authoring players; this requires a serial keyboard or terminal.

As with other OS-9 documentation, this manual had to be purchased from Microware. However, Revision D of *Using Professional OS-9* can be downloaded from:

<http://www.icdia.co.uk/microware/using/>

Revision D, although differently titled and formatted, is also part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>

At SPC we worked with Revision D in a three-ring binder with 6" x 8" paper that came with the OS-9 System Disk floppy, but we seldom needed it except for debugging or maintenance tasks.

### 4.2. Using OS-9/Internet

This manual describes OS-9 Internet commands and C library extensions that support the use of such programs as ftp and telnet. These functions are not available on consumer CD-i players.

There exist extension boards for the CDI 182 authoring player unit and the CDI 60x player that add Internet functionality; these boards have the commands and their support modules in ROM.

Revision B of this manual is part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>

At SPC we almost never used the Internet functionality of our 605 players.

### 4.3. OS-9 C Compiler manuals

These manuals describe the OS-9 C Compiler system from Microware. A large part of these manuals is the section *The C Standard library* that describes all of the compiler-supplied library functions available for C language programs. This includes C language bindings for most of the general OS-9 system calls.

This manual had to be purchased from Microware and was typically delivered with the Microware C Compiler software package.

The *OS-9 C Language User Manual* is part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>

This manual consists of the following sections:

*OS-9 C Compiler and OS-9 C Functions*, Revision I, March 1991  
*OS-9/68000 Source Level Debugger*, Revision B, July 1988 (SrcDbg)

As SPC we used an earlier revision of the first document in a three-ring binder with 6" x 8" paper, we did not initially have the Source Level Debugger document:

*OS-9/68000 C Compiler Users's Manual*, Revision H, October 1989

These revisions are differently formatted but contain mostly the same information.

As part of UniBridge we later acquired a three-ring binder with 6" x 8" paper containing the following documents:

*OS-9/68000 Source Level Debugger User Manual*, Revision B, July 1988 (SrcDbg)  
*OS-9/68000 System State Debugger User's Manual*, Revision A, February 1987 (sysdbg)

We never used either of these debuggers in CD-i development.

Knowledge of the C compiler manual contents is necessary for any CD-i title developer using the C language, especially the section on the standard library.

#### 4.4. OS-9 Assembler/Linker manuals

These manuals describe the OS-9 Assembler/Linker system from Microware that forms the back-end of the OS-9 C Compiler. The manual includes documentation for all the assembler directives and linker options and also describes the Relocatable Object File Format (ROFF) used by these tools in considerable detail.

This manual had to be purchased from Microware and was typically also delivered with the Microware C Compiler software package.

Revision H of the *OS-9 Assembler/Linker User Manual* is part of the Allen-Bradley set of Microware documents that is available online from several sources, one of these is:

<http://www.icdia.co.uk/microware/>

At SPC we used an earlier revision of this document in a three-ring binder with 6" x 8" paper that also contained documentation for the User-State Debugger:

*OS-9/68000 Assembler, Linker and Debugger*, Revision G, October 1989  
*OS-9/68000 User-State Debugger (debug)*

These revisions are again differently formatted but contain mostly the same information.

We never used the Microware-supplied debugger, preferring our own in-application debugger.

Some knowledge of the contents of these manuals is necessary for any CD-i title developer using the C language; developers using assembly language need to know most of it.

#### 4.5. CD-I System Library Technical Reference Manual

This manual describes the C language bindings for the CD-i specific system services described in the Green Book and the C language bindings for a few more esoteric OS-9 system calls.

This manual had to be purchased from Microware and was typically also delivered with the Microware C Compiler software package.

A scanned PDF version of Revision E is available from:

<http://www.icdia.co.uk/microware/>

At SPC we worked with revision E of this manual in a three-ring binder with 6" x 8" paper.

The contents of this manual are required for any CD-i title developer using the C language who wants to use CD-i specific system services.

*Note:* The implementation of the `srqcmem` function described in this manual contains a bug when the size parameter is -1; see PIM Technical Note #87 *Memory Allocation in CD-RTOS* for more information.

#### 4.6. OS-9 IFF Documentation

These documents describe a support library and the file format for CD-I IFF files, which is the format typically used by CD-i authoring tools for representing assets (images, audio, video).

The following documents are available:

*OS-9/IFF Support Library C Functions*

*CD-I IFF Specification: Version 0.99 6 January 1989*

*A Quick Introduction to IFF*

*"EA IFF 85" Standard For Interchange Format Files: August 1, 1987*

*Audio Interchange File Format: AIFF*

Only the first two are specific to CD-i, the others are generic IFF documents from Electronic Arts.

A bundled PDF version of these five documents is included on the CDISC 4 Freeware CD-I and is also available from ICDIA at <http://www.icdia.co.uk/authoring/>.

In-depth knowledge of the contents of these documents is not necessary unless you want to understand the low-level details of the asset file formats.

At SPC we worked with a printed version of this document.

#### **4.7. Disc building and emulation documentation**

These documents describe the Microware CD-i disc image and CD-i real-time-file building tools *Master* and *Green*, the disc image editing tool *cdedit* and how to use these tools with the hardware CD-I Emulator device.

At SPC we worked for many years with the following soft-ring bound 6" x 8" thin paper manuals that matched the Windows / DOS version we were using:

CD-I Emulator *for PC*, Revision B, January 1993  
CD-I Disc Building *for PC*, Revision B, January 1993  
CD-I Disc Building Tutorial *for PC*, Revision B, June 1991

These manuals describe the Windows graphical versions of Master and Green but we mostly used the command-line version of *master* that used a 368 DOS extender like the *cdedit* tool.

We later acquired Sun and Macintosh versions of these manuals in white three-ring binders with 6" x 8" paper describing newer versions of these tools including support for full-motion video.

The Sun versions of these manuals are available from ICDIA at [www.icdia.co.uk/authoring/](http://www.icdia.co.uk/authoring/).

#### **4.8. OS-9 Version 2.4 Release Notes**

This document describes the changes that were made to OS-9 from version 2.3 to version 2.4 and mostly consists of a long list of corrections and enhancements.

This document is available from ICDIA at [www.icdia.co.uk/microware/](http://www.icdia.co.uk/microware/).

At SPC we only acquired this document in the later years and never used it for CD-i development.

#### **4.9. OS-9 C Compiler Version 3.2 Release Notes**

This document describes the changes that were made to the OS-9 C compiler from version 3.1 to version 3.2, which include changes in distribution packaging and edition numbers of all executables.

This document is available from ICDIA at [www.icdia.co.uk/microware/](http://www.icdia.co.uk/microware/).

At SPC we only acquired this document in the later years and never used it for CD-i development.

## 5. Development tools

A number of high-level tools exist for development of CD-i titles, e.g. MediaMogul, HAL9000; some of these even support multi-platform development.

In staying with the low-level approach of the previous sections we will ignore such tools here, instead focusing on low-level development using the C and/or C++ programming languages and, if necessary, 68000 machine language. All the high-level tools are ultimately based on these foundations anyway, so everything that can be done with a high-level tool is also possible with low-level development, while the reverse is not necessarily true.

Also, better performance can generally be obtained with low-level development, which is not unimportant given the relatively resource-constrained nature of the CD-i platform.

In this section we will mainly focus on the “standard” tools from Microware (the creators of the OS-9 operating system on which CD-RTOS is based) and OptImage (a joint-venture between Microware and Philips), but there were additional vendors of low-level CD-i development tools.

### 5.1. Toolkits

Even when doing low-level development toolkits can be used, Balboa being one of the most well-known. While such a toolkit can be very useful, it is the feeling of the author that these can also be a significant barrier to entry because of the additional complexity involved. The low-level system services specified by the Green Book are not that complex although sometimes tedious to use. Consequently, we will ignore such toolkits here.

At SPC we only ever used a small part of Balboa and only in our first CD-i title. After that we evolved our own little toolkit that did not abstract away the low-level details but provided some additional functionality not directly provided by the underlying platform or packaged up existing functionality in a more convenient way.

## 5.2. Compilers and assemblers

For low-level development, tools such as a C/C++ compiler and/or an assembler and a linker are needed to produce program binaries that can be run under CD-RTOS.

Microware supplies a complete OS-9 C compiler / assembler / linker suite in both native versions (running on OS-9) and cross versions (running on other platforms, e.g. DOS/Windows, MacOS and various Unix flavors), intended for use with OS-9/68000 version 2.3 and newer.

There are several versions of the Microware C compiler, for CD-i development version 3.2 was typically used (either native or cross) as it came with the required CD-RTOS include files and linker libraries. This compiler implements a superset of the C language as described by the first edition of Kernighan & Ritchie that predates the ANSI C standardization effort.

Microware also supplies the Ultra C compiler that supports ANSI C and C++ but this compiler is intended for use with OS-9 3.0 or newer and hence not relevant for CD-i development.

The native version 3.2 compiler can be run either on an OS-9 system (a professional CD-i player could be used for this) or using an OS-9 emulator such as os9exec.

The DOS/Windows cross version 3.2 compiler can be run under modern 32-bit Windows versions (it does not run on 64-bit Windows) and emulators such as vDOS or DOSBox.

There are also various ports of the GNU C/C++ compiler to OS-9, both native and cross:

- gcc version 1.39 was ported by Atsushi Seyama and maintained up to 1.42.2 by Stephan Paschedag; these versions utilize the Microware r68 assembler and l68 linker.
- gcc version 2.5.6 was ported again and maintained up to at least 2.7.2 by Stephan Paschedag, still utilizing the Microware r68 assembler and l68 linker.
- gcc version 2.6.2 was ported again by Walter Hunt, partially based on the work of Stephan Paschedag. This version implements full bfd support for OS-9 resulting not only in a working gcc/g++ but also the GNU assembler and linker and most of the GNU binutils tools.

Sources and native binaries for these gcc ports can be found online on various OS-9 websites, including the following URL: <ftp://os9archive.rtsi.com/OS9/OSK/GCC/>.

More recently some work was done on porting gcc 2.7.2 to Windows using Cygwin based on the work by Walter Hunt, but no usable version has yet been released, as reported on CDinteractive here: <http://www.cdinteractive.co.uk/forums/cdinteractive/viewtopic.php?t=2629>

At SPC we used the Microware version 3.2 PC cross compiler until around 1995 when we started from Stephan's 2.5.6 port to build a go32-based DOS version of gcc 2.5.7 and 2.5.8, still utilizing the Microware assembler and linker. From that point we started developing in C++ instead of C. Brief mention of this: [http://www.verycomputer.com/74\\_9441881ea148c795\\_1.htm#p2](http://www.verycomputer.com/74_9441881ea148c795_1.htm#p2)

### 5.3. Asset conversion

The CD-i system uses its own special formats for images, audio and video. The in-memory and on-disc formats are documented in the Green Book, the authoring file formats are documented in the OS-9 IFF documentation.

During development, files in these formats must be created from the more common formats typically used in contemporary systems (e.g. WAV for audio, BMP for images).

Conversion of audio from PCM to ADPCM format is complex and relatively slow.

Conversion of images to CD-i format is relatively easy except for conversion to DYUV.

OptImage supplied a number of conversion tools in two bundles:

Audio Conversion Tools  
Image Conversion Tools

For Windows, these were combined into the “ACU Shell” and “ICU Shell” programs. These programs can still be run under modern 32-bit Windows versions (not on 64-bit Windows).

At SPC we usually used our own tools for most image conversions, including compiling images as sprites; our tools were usually scriptable from the command line which was much easier with large amounts of images or when changes were made (which happened frequently). For audio conversions we typically used the ACU Shell as these were far less in number, or we had the audio studio do it.

For Frog Feast, Charles Doty built his own image conversion tools which are included in his released sources.



#### 5.4. Disc building and emulation

The finished CD-i program binaries must be combined with the asset and possibly other data files into a *disc image* file that will eventually be burned or pressed to a CD. However, it would be very cumbersome to have to burn CDs all the time during development, so given the necessary hardware and/or software, there is a shortcut.

Originally, a hardware device confusingly called a *CD-I Emulator* would read the disc image file from a SCSI hard disk and *emulate* a physical CD drive for the CD-i player by supplying it with the exact digital signals that would have been produced by an actual CD drive.

Nowadays, a *software* CD-i emulator program will read the same image file and emulate a complete CD-i player inside the host operating system.

There used to be a variety of disc building tools, *master* (also called *vcdmaster*) from OptImage probably being the most common one. Like the compilers, these tools used to come in versions for a variety of platforms and were typically script-driven.

Even high-level authoring tools such as MediaMogul often use the standard tools for building the final disc image, in some cases they can preview the title from a hard disk.

Using a CD-I Emulator device with an additional SCSI connection allows disc images to be burned to a CD using a Philips CDD 521 or a Yamaha burner (these were expensive at the time). Nowadays CD-i disc images can be burned by many tools on a PC CD-ROM burner.

OptImage supplied the disc building and emulation / burning tools in two bundles:

- Disc Building Tools
- Disc Burning & Emulation Tool

The DOS/Windows versions of these tools can still be run under modern 32-bit Windows versions (not on 64-bit Windows).

As SPC we normally used the command-line enabled *master* tool from OptImage, typically followed by many program file replacements using the *cdedit* tool during debugging. Often we also created the real-time files for the disc using our own tools.

## 6. Support material

This section lists some additional material that Philips used to make available to support CD-i developers. Most came on floppy disks, often together with documentation that has been described in previous sections of this document.

The use of this material is not essential for developing a CD-i title, but some of it can be useful.

### 6.1. OS-9 System Diskette

This 3.5" floppy disk contains the shell and most of the general purpose OS-9 commands that are missing from consumer CD-i player ROMs. It is intended specifically for the Philips CDI 605 player, but also be used with the CDI 182 authoring player unit.

This disk was distributed with the *605 Driver Update Disk* and the following documents:

*OS-9 System Disk for the CD-I 605 Player*  
*CD-I 605 Player Driver Update*  
*uMacs Reference Manual*  
*Using Professional OS-9*

A disk image for this disk is available from ICDIA at [www.icdia.co.uk/sw\\_disk/](http://www.icdia.co.uk/sw_disk/).

At SPC did not use this disk much directly because 605 players have many of the OS-9 commands in ROM. We did however use it to create an OS-9 CD-i disc that could be used to get an OS-9 shell prompt on consumer players, which was sometimes useful for debugging.

### 6.2. 605 Driver Update

This 3.5" floppy disk contains driver updates for the Philips CDI 605 player; see *CD-I 605 Player Driver Update*. This disk was distributed with the *OS-9 System Disk* and contained driver updates for SCSI hard disks and SCSI tape drives as well as an updated version of the OS-9 Internet Support Package.

This update was mostly intended for MediaMogul users because they used SCSI hard disks and tape drives directly connected to their 605 players.

A disk image for this disk is available from ICDIA at [www.icdia.co.uk/sw\\_disk/](http://www.icdia.co.uk/sw_disk/).

At SPC we almost never used this disk, the ROM drivers were good enough for the very few cases that we needed them.

### 6.3. Base-Case Tools Disk

This 3.5" floppy disk contains a number of OS-9 tools that can be used to test a CD-i application for compliance with the Base Case System restrictions. Some of them overlap in functionality with the CDI 605 players tools button.

Passing these compliance tests was required for Philips-sponsored CD-i titles, as part of ensuring that your CD-i title would run on all CD-i players. They would still be tested on "current" players but most of them provided a little more than the required Base Case functionality (especially memory).

These tools are included in the OS9/fromPIMA directory of the *CDISC 4 Freeware* disc described below for which a disc image is available from ICDIA at [http://www.icdia.co.uk/sw\\_disc](http://www.icdia.co.uk/sw_disc).

At SPC we generally used these tools just before releasing discs for testing.

### 6.4. Final Production Delivery Disc

This CD-i contains a number of standard assets (text/audio/video) that Philips Media required to be used in any Philips-sponsored CD-i title as described in the *CD-i Final Production Delivery Specification*. It also includes a version of the Base Cases tests described above.

The text files are language-localized versions of various standard messages that Philips Media required for each title ("Your disc may be dirty", "All rights reserved", "You need a DVC").

The audio files are language-localized versions of the standard audio warning required to be used in the leadin of every CD-i disc ("Warning: This disc is not intended for use on your CD-Audio player. Continued listening may damage your equipment.").

The video files are the well-known Philips Media "bumper" animations at the start of all Philips Media titles, in various resolutions and for Base Case systems as well for Full Motion Video capable systems. Sample source code for playing these is included on the disc as well.

A disc image for this disc is available from ICDIA at [http://www.icdia.co.uk/sw\\_disc](http://www.icdia.co.uk/sw_disc).

At SPC we used the texts and audio files of this disc as required by Philips and we based our "bumper" animations on the supplied sample source code.

## **6.5. OptImage Public Domain Software Disks**

These two 3.5" floppy disks contain a host of public domain OS-9 programs, most of these are also on the *CDISC 4 Freeware* disc (see below).

At SPC we did not really use these disks much because we weren't using the OS-9 shell prompt.

## **6.6. CDISC 4 Freeware CD-i**

This CD-i disc contains a wealth of tools and information, often including sources. It was originally distributed to CD-i developers attending the 4<sup>th</sup> CDISC developer conference in 1993.

A disc image for this CD-i is available for download from:

[http://www.icdia.co.uk/sw\\_disc](http://www.icdia.co.uk/sw_disc)

At SPC we used some of the sources and tools from this disc in our daily work, but much of it was not directly relevant to our development process even though it made for interesting browsing.

## 7. Version history

### Version 2019-03-31:

- First release for the developer section of *The world of CD-i*.

### Version 2021-02-25:

- Corrections and updates with many new links.