CD-I Authoring Tools

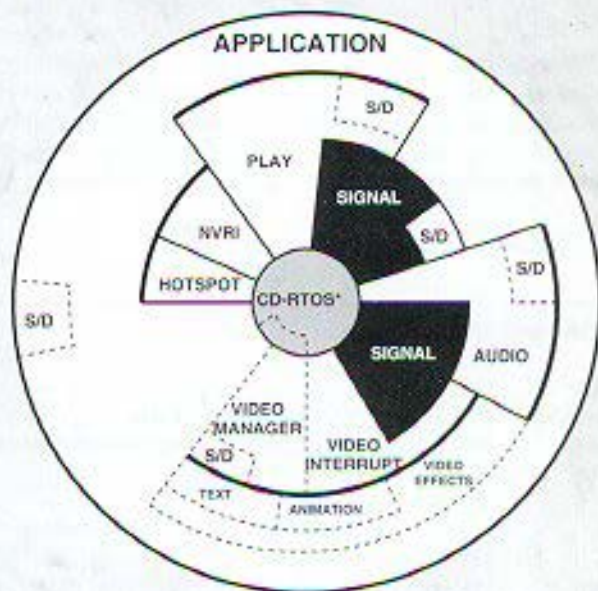# The Balboa Runtime Environment

for SUN® Workstations, Macintosh II®, or PC compatible computers

Until recently, the production of CD-I titles was a substantial undertaking, requiring the talents of software specialists who approached each title as a unique custom encoding task with rigid specifications.

A new solution for CD-I programming

- An API (Application Programming Interface) toolbox of ready to use CD-I runtime libraries
- An interface between the programmer and CD-RTOS (the realtime operating environment of CD-I hardware) allows production teams to concentrate on the more creative aspects of a CD-I title
- Second generation product covering the full spectrum of CD-I programming: audio, image, cursor interaction and realtime play management
- Using the hierarchical action facilities, the process of designing for an interactive environment is made easier

PHILIPS

Balboa Architecture:
- Engine for dispatching functions when they are called
- Engine for processing signals
- Modular architecture of 16 managers which allows more than one logical program to run at any given time

Balboa Managers:
- 16 individual managers can handle a whole range of functions from playing events to creating interactive buttons and switches
- Simplifies operation of both high and low level aspects of CD-I, including audio and visual assets, video interrupts for synchronised effects, timers and dealing with errors

Balboa Flexibility:
- Easier way of dealing with the principal data structures used for CD-I programming, with flexible, transparent working environment

- Buffer to CD-RTOS
- High degree of code abstraction, instantly familiar to C programmers
- Picture Object function for manipulation of an entire container of video information
- Experienced CD-I programmers can write directly for CD-RTOS and combine those functions with Balboa functions

Balboa and Interactivity:
- Call-back functions methodology overcomes problem of user unpredictability
- Each function in the Balboa library has a discrete purpose
- Functions can be called and executed any time
- In turn, that function can dispatch other call-back functions or set up an internal environment that causes other call-back functions to occur (eg. setting up a hotspot tree or initiating a play sequence)

TECHNICAL SPECIFICATIONS

Language:	C
Environment:	Requires the OS-9 compiler / linker
Functions:	Balboa consists of the following managers (modules):
Dispatcher	Controls execution of functions
Signal	Handles OS-9 signals
Timer	Handles the installation of functions based on timer intervals
Video	Controls video display
Video Env	Handles sharing of images and pointer resources
Effects	Handles animation, video effects (Wipes, X-fades, etc)
Play	Handles delivery of assets to buffers and management of realtime files
Cursor	Controls cursor display and cursor management
Hotspot	Handles regions where pointer actions take place
Controls	Handles user interaction
Audio	Handles the play of audio
Status	Records debug, error, and status information
NVRI	Supports International standards for non-volatile RAM files
IFF	Supports International standards for Interchange File Format data
Animation	Handles animation display
Memory	Application-dependent. Balboa system kernel requires 3K. A typical fully functional configuration (Video Manager, Cursor/Hotspot Manager, Dispatcher Manager and Signal Manager) requires approximately 76K
Documentation	Balboa Programmer's Guide and Balboa Reference Guide (4 volumes)
Training	One week training courses available through Philips IMS
Release	MediumSUN® 3 and 4: Quarter inch tape cartridge Apple® Macintosh and PC compatibles: 3.5 in high density diskettes

Developed in cooperation with

Opt Image

For more information contact Philips IMS:

In the US: 1 (800) CDI - 5484
In Europe: +31 (40) 73 6228
In Japan: (03) 3665 - 9740
Applelink® D6431

PHILIPS

PHILIPS