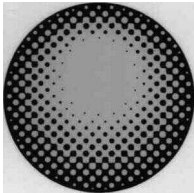


3DO PORTFOLIO VERSION 2.5
3DO TOOLKIT • VERSION



Guide to 3DO Documentation

3DO PORTFOLIO VERSION 2.5



Getting Started With 300 Portfolio

- * 300 Jumpstart
- * Peripherals
- * Example Programs

Programming the 300 Portfolio System

- * 3DO System Programmer's Guide
- * 3DO System Programmer's Reference

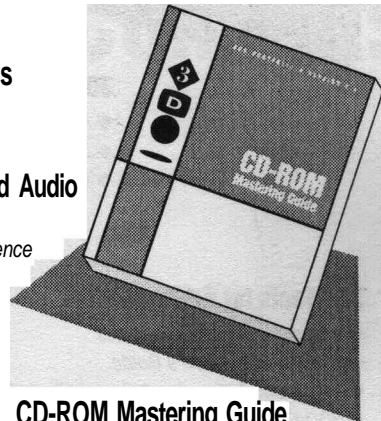
Programming 300 Portfolio Graphics

- * 300 Graphics Programmer's Guide
- * 300 Graphics Programmer's Reference

Programming 300 Portfolio Music and Audio

- * 300 Music and Audio Programmer's Guide
- ◆ 300 Music and Audio Programmer's Reference

LibSDO Programmer's Reference



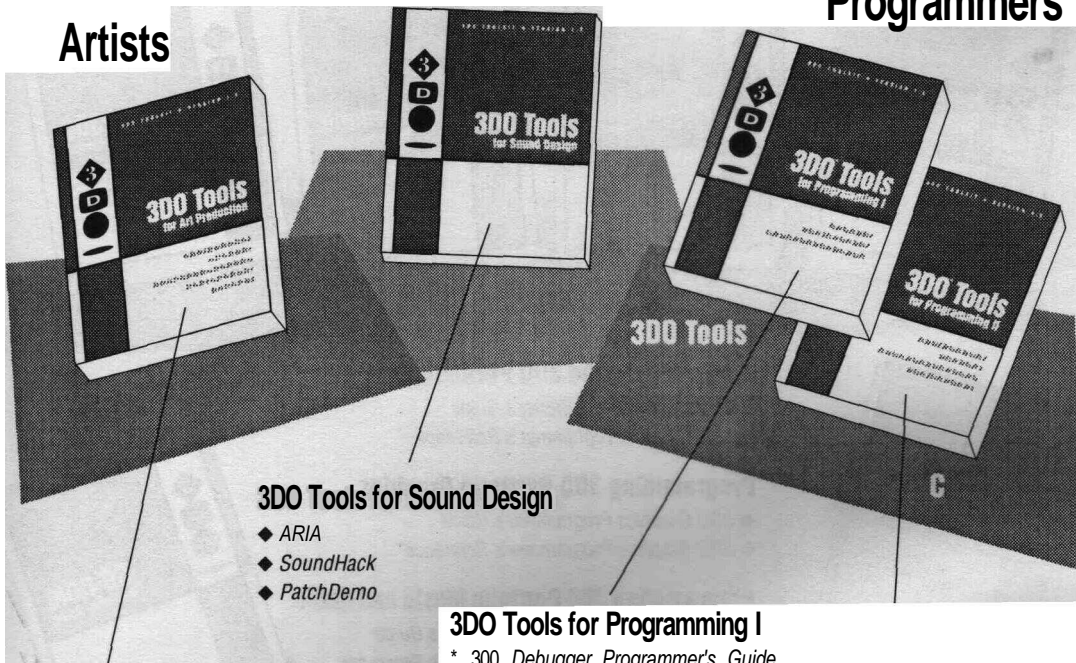
CD-ROM Mastering Guide

VERSION 1

Sound Designers

Programmers

Artists



3DO Tools for Sound Design

- ◆ ARIA
- ◆ SoundHack
- ◆ PatchDemo

3DO Tools for Programming I

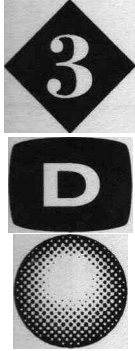
- * 300 Debugger Programmer's Guide
- * 300 DataStreamer Programmer's Guide
- * 300 DataStreamer Programmer's Reference

3DO Tools for Art Production

- * Introduction to 3DO Graphics
- * 300 Animator User's Guide
- * 300 Custom Plug-Ins & Custom Code Modules
- * 300 FontWriter User's Guide
- * Video Processing Guide

300 Tools for Programming II

+ ARUCOUMX*



3DO Documentation —Global Contents

3DO Portfolio 2.5—Contents

Getting Started With 300 Portfolio

Preface

About This Document.....	xv
Audience.....	xv
How This Document is Organized.....	xv
Jumpstart Example Programs	xv
Etiquette and MPW Information	xvi
Peripherals	xvi
Example Programs	xvii

1

Building and Running a Program

Chapter Overview	PGS-1
Programming in the 3DO Environment.....	PGS-2
Libraries	PGS-2
The 3DO Portfolio Operating System	PGS-2
LibSDO and the 3DO DataStreamer.....	PGS-3
Documentation	PGS-3
Preparing a Program.....	PGS-4
Creating a Makefile Using CreateSDOMake.....	PGS-4
Building the Program	PGS-6
Tips, Tricks, and Troubleshooting.....	PGS-7
MPW Extensions	PGS-7
Creating Your Own Makefile	PGS-8

Displaying Cels

Chapter Overview.....	PGS-9
Fundamentals of 3DO Graphics.....	PGS-10
3DO Cels and Images	PGS-10
Uncoded and Coded Cels	PGS-11
Commonly Used Cel Formats	PGS-12
Initialization and Shutdown in a 3DO Program.....	PGS-13
Initialization	PGS-13
Cleanup.....	PGS-16
Displaying Images and Cels.....	PGS-17
Tips, Tricks, and Troubleshooting	PGS-20
Languages	PGS-20
3DO Datatypes	PGS-20
Efficient Display Using Cel Lists	PGS-21

3

Playing an Animation

Chapter Overview.....	PGS-23
3DO Animation Basics.....	PGS-24
Animation File Formats	PGS-24
Tools for Creating 3DO Animations	PGS-24
Double Buffering	PGS-24
Working With Animations.....	PGS-26
Loading an Animation	PGS-26
Playing the Animation	PGS-27
Tips, Tricks, and Troubleshooting	PGS-30
Using an Animation Unified for Efficiency	PGS-30
Creating an Anti-Aliased Animation	PGS-30

4

Getting User Input

Chapter Overview.....	PGS-31
Getting Control Pad Input.....	PGS-32
Utility Functions for Using the Control Pad	PGS-32
The jsanimation.c Example	PGS-32
The basicslideshow.c Example	PGS-34
Tips, Tricks, and Troubleshooting	PGS-36
Control Pad Input and the 3DO DataStreamer	PGS-36
Other Input Devices	PGS-36
Saving Games and Scores	PGS-36

Moving and Distorting a Cel

Chapter Overview.....	PGS-37
Fundamentals of Cel Manipulation.....	PGS-38
The CCB Structure	PGS-38
Frequently Accessed CCB Fields	PGS-39
Moving and Distorting a Cel.....	PGS-40
Manipulating a Cel by Changing CCB Values	PGS-40
Using MapCel to Manipulate a Cel	PGS-43
Tips, Tricks, and Troubleshooting.....	PGS-45
CCB Formulas	PGS-45
Working with Cels Efficiently	PGS-45
Additional Code Examples	PGS-45

6

Portfolio Programming Etiquette

Chapter Overview.....	PGS-47
Accessing Low Memory.....	PGS-48
Example and Discussion	PGS-48
Making Assumptions About Memory Types.....	PGS-50
Example and Discussion	PGS-50
Relying on Side Effects of Multitasking	PGS-51
Using Busy Waiting.....	PGS-52
Discussion and Example.....	PGS-52
Ignoring Variable Frame Rates.....	PGS-54
Discussion and Example'.....	PGS-54
Using Fixed Pathnames.....	PGS-60
Using Self-Modifying Code.....	PGS-60
Leaving Unused Fields in DMA Structures Uninitialized.....	PGS-61
Bypassing the API.....	PGS-61
Hoarding System Resources.....	PGS-62

7

Toolkit Extensions to MPW

Chapter Overview.....	PGS-63
Overview of MPW Tools and Scripts.....	PGS-64
ppmtoSDO	PGS-66
makecel	PGS-68

8

Peripherals Overview

Chapter Overview.....	PGS-69
Classes of Peripherals.....	PGS-70
Digital	PGS-70
Analog	PGS-70
Noise Level	PGS-71
Daisy-Chaining Peripherals.....	PGS-72
Event Broker.....	PGS-72
Sample Programs	PGS-73
Related Documentation.....	PGS-75

9

The Light Gun

Chapter Overview.....	PGS-77
Overview.....	PGS-78
Requirements	PGS-78
How a Light Gun Works	PGS-78
Installing the Light Gun	PGS-80
The Light Gun Driver	PGS-81
Helpful Hints.....	PGS-81
Testing the Light Gun.....	PGS-81

10

The Extended Joystick

Chapter Overview	PGS-83
Overview.....	PGS-84
Requirements	PGS-84
Features of the Extended Joystick	PGS-84
Bit Order.....	PGS-87
Installing the Extended Joystick.....	PGS-88
Extended Joystick Driver	PGS-89
Testing the Extended Joystick.....	PGS- 89
Programming the Extended Joystick.....	PGS- 90
Extended Joystick Resistors	PGS-90
Calibrating the Extended Joystick	PGS-90
Recommended Sequence for Setup	PGS-91

11**The Mouse**

Chapter Overview.....	PGS-93
Requirements.....	PGS-94
Sample Programs.....	PGS-94
Installation.....	PGS-96
Testing the Mouse.....	PGS-97

12**Example Programs**

Sdoorbit.....	PGS-104
aaplayer	PGS-105
access.....	PGS-106
allocate	PGS-107
allocmem	PGS-108
animationvdl	PGS-109
animsample	PGS-110
beep.....	PGS-111
bounce.....	PGS-112
bs_example	PGS-113
capture_audio	PGS-115
CoalRiver.....	PGS-116
colorecho.....	PGS-117
compression	PGS-118
cpdump	PGS-119
drumbox	PGS-120
effectshandler.....	PGS-121
focus.....	PGS-122
fontlibexample	PGS-123
fontviewer.....	PGS-124
joystick_example	PGS-125
jsanimation	PGS-126
jsbasicssideshow.....	PGS-127
jsinteractivemusic.....	PGS-128
jsinteractivesound.....	PGS-129
jsintmusicthread	PGS-130
jsmovecel	PGS-131
jsplaybgndmusic.....	PGS-132
jsshowcel.....	PGS-133

jsslideshowvdl	PGS-134
kanjifontviewer	P6S-135
lightgundemo	P6S-136
lookie	PGS-137
lrex	PGS-138
ls.....	PGS-139
luckie	PGS-140
maus.....	PGS-141
memdebug	PGS-142
menu	PGS-143
minmax_audio	PGS-144
mouseufo	PGS-145
msgpassing	PGS-146
nvrnm	PGS-147
perftest	PGS-148
playmf.....	PGS-149
playsample	PGS-150
playsoundfile	PGS-151
signals	PGS-152
simple_envelope	PGS-153
slideshow	PGS-154
slideshow24	PGS-155
spoolsoundfile	PGS-156
storagetuner	PGS-157
symanim	PGS-158
ta_attach	PGS-159
ta_customdelay	PGS-160
ta_envelope	PGS-161
ta_pitchnotes	PGS-162
ta_spool.....	PGS-163
ta_sweeps.....	PGS-164
tajimer	PGS-165
tajuning	PGS-166
tajweakknobs.....	PGS-167
timerread	PGS-168
timersleep	PGS-169
tj_canon.....	PGS-170
tj_multi	PGS-171

tj_simple.....	PGS-172
tsc_soundfx	PGS-173
tsp_algorithmic.....	PGS-175
tsp_rooms	PGS-177
tsp_spoolsoundfile	PGS-179
tsp_switcher.....	PGS-180
type.....	PGS-181
ufo	PGS-182
userfolio	PGS-183
walker.....	PGS-184

13

Functions Used in Example Programs

DoControlPad	PGS-186
InitControlPad	PGS-188
KillControlPad	PGS-189
ReturnPreviousControlPad	PGS-190

Programming 3DO Portfolio Graphics

List of Figures	xv
-----------------------	-----------

List of Tables	xvii
----------------------	-------------

Preface.....	xi
--------------	-----------

3DO Graphics Programmer's Guide*

Preface

About This Book.....	ix
About the Audience.....	ix
How This Book Is Organized.....	ix
Related Documentation.....	x
Typographical Conventions.....	x

1

Understanding 3DO Graphics

NTSC Display Basics.....	GPG-2
Image Scanning	GPG-2
Frame Rates.....	GPG-2
The Scan Pattern	GPG-3
3DO NTSC Graphics	GPG-4
Introduction to the 3DO Graphics Systems.....	GPG-5
The Display Generator	GPG-6
The Cel Engine	GPG-10
The SPORT Bus	GPG-14

2

Understanding the Display Generator

Display Generator Components.....	GPG-18
The Frame Buffer.....	GPG-20
Left/Right Pixel Pairs	GPG-20
Display Modes	GPG-21
The Background Pixel Detector.....	GPG-22
The GLUT Sets.....	GPG-23
The Custom GLUT Set	GPG-23
The Fixed GLUT Set	GPG-23
Choosing a GLUT Set.....	GPG-24
Handling Background Pixels	GPG-24
Passing on Color Values	GPG-25
Slipstream.....	GPG-25
The Interpolator.....	GPG-25
Specifying Cornerweight Location	GPG-25
Color Averaging	GPG-26
Changing Cornerweight Positions in a Moving Cel	GPG-27
The D to A Converter.....	GPG-28
The Video Display List Processor.....	GPG-28

3

Understanding the Cel Engine and SPORT

The Gel Engine.....	GPG-30
Cel Source Data	GPG-32
The Preamble.....	GPG-36
The Cel Control Block (CCB)	GPG-36

The Data Unpacker (DUP).....	GPG-38
The Pixel Decoder (PDC).....	GPG-38
Reading Color Values in Coded Pixels	GPG-38
Reading Color Values in Uncoded Pixels	GPG-39
Passing Values to Other Cel Engine Components	GPG-39
Uncoding Rules	GPG-39
The Pixel Processor.....	GPG-44
Primary Source.....	GPG-46
The Primary Scalar	GPG-47
The Secondary Source Value	GPG-48
The Secondary Divider.....	GPG-48
The Final Math Stage	GPG-49
Passing On Color Values	GPG-49
The Projector.....	GPG-49
Transparent Pixels	GPG-50
Adding VH Values	GPG-50
Specifying Frame Buffer Coordinates.....	GPG-51
Setting the Gel's Origin Corner	GPG-52
Projecting Cel Pixels	GPG-52
Changing Cel Pixel Size	GPG-56
The Corner Engines	GPG-57
Cel Clipping	GPG-57
Backface Removal	GPG-62
SPORT Transfers to the Frame Buffer.....	GPG-62
Restrictions	GPG-63
Clearing the Frame Buffer	GPG-63
The Path of VH Values.....	GPG-64
An Overview of the Path	GPG-64
Processing Cel Source Data in the Pixel Decoder.....	GPG-65
Reading a P-Mode Value for the Pixel Processor.....	GPG-66
Processing VH Values in the Projector	GPG-66
Reading a Palette Value	GPG-69
Processing VH Values in the Interpolator	GPG-69

4

Using the Display Generator

An Overview of Frame Programming.....	GPG-73
Screens.....	GPG-73
Bitmaps	GP6-74
Screen Groups	GPG-74
Multiple Screen Groups in a Frame	GPG-75
Examples	GPG-76
Opening the Graphics Folio.....	GPG-79
Creating a Screen Group.....	GPG-79
The CreateScreenGroup() Call	GPG-79
Setting the Screen Count and Dimensions	GP6-80
Setting Bitmap Counts, Dimensions, and Buffers	GPG-81
Setting Screen VDL Types and Attaching Custom VDLs	GPG-82
Creating a Custom VDL.....	GPG-83
VDL Processor Operation	GPG-84
VDL Structures	GPG-85
Writing a VDL	GPG-95
Submitting a Screen VDL.....	GPG-97
Modifying a VDL	GPG-98
Setting a New VDL for an Existing Screen	GPG-99
Deleting a VDL	GPG-99
Setting a Screen Color Palette.....	GPG-99
A CLUT Set Review	GPG-99
Specifying a New Clor.....	GPG-100
Setting a New Color Register Value in the CLUT Set.....	GPG-101
Setting Multiple New Color Register Values in the CLUT Set	GPG-101
Reading Current CLUT Set Registers	GPG-101
Resetting the Fixed Palette for a Screen	GPG-102
Displaying a Screen Group.....	GPG-102
Adding a Screen Group to the Display	GPG-102
Displaying Screens	GPG-102
Removing a Screen Group from Display	GPG-103
Deleting a Screen Group	GPG-103
Rendering Into a Screen.....	GPG-104
Creating a Graphics Context	GPG-104
Drawing Graphics Primitives	GPG-106
Finding a Pixel's Color and Address	GPG-107

Rendering Text	GPG-108
Placing Characters.....	GPG-108
Setting a Clipping Rectangle	GPG-108
Refreshing Backgrounds with SPORT Transfers.....	GPG-109
Creating an IOReq for the SPORT Device	GPG-110
Copying VRAM Pages	GPG-110
Cloning a Single VRAM Page	GPG-111
Setting VRAM Pages to a Single Color or Pattern	GPG-112
Deferred SPORT Calls.....	GPG-112
Display Timing Calls.....	GPG-113
Getting a VBL IOReq	GPG-113
Waiting for a VBL Frame	GPG-113
Controlling Pixel Interpolation.....	GPG-113
Primary Data Structures.....	GPG-114
The Graphics Context (GrafCon) Data Structure	GPG-114
The Reel Data Structure.....	GPG-115
Function Calls.....	GPG-115

5

Using the Cel Engine

An Overview of Cel Programming.....	GPG-120
The CCB	GPG-120
The Cel Engine Control Word	GPG-121
Creating and Importing Cel Components.....	GPG-121
Working With a CCB.....	GPG-122
The Flag Values	GPG-123
The NEXTPTR Word	GPG-129
The SOURCEPTR Word	GPG-130
The PLUTPTR Word	GPG-130
The XPOS and YPOS Words	GPG-130
The HDX, HDY, VOX, and VDY Words	GPG-130
The HDDX and HDDY Words	GPG-131
The PIXC Word	GPG-131
The PRE0 and PRE1 Words	GPG-134
Creating Cel Source Data.....	GPG-134
Unpacked Source Data	GPG-135
Packed Source Data	GPG-135
Creating a Preamble.....	GPG-138
The First Preamble Word	GPG-139

The Second Preamble Word	GPG-141
Extracting Cel Source Data from an Existing Image.....	GPG-144
Setting a CCB for Data Extraction	GPG-144
Creating a Preamble for Extracted Data	GPG-145
Using a Cel to Reveal Bitmap Contents.....	GPG-147
Setting Read and Write Buffers	GPG-148
Creating and Loading a PLUT.....	GPG-149
Projecting a Linked Group of Cels.....	GPG-149
Setting Pointer Values in the CCBs	GPG-150
Projecting Cels.....	GPG-150
Changing a Cel through Its CCB.....	GPG-151
Changing Cel Position	GPG-151
Changing a Cel's Projection Quadrilateral	GPG-151
Skipping a Cel Within a Linked Group	GPG-152
Sequentially Animating a Cel	GPG-152
Working with the Cel Engine Control Word.....	GPG-153
Changing a Bitmap's Cel Engine Control Word	GPG-153
The Cel Engine Control Word Values	GPG-154
Controlling VH Values	GPG-156
Loading a New Cel Engine Control Word	GPG-158
Primary Data Structures.....	GPG-158
The Cel Control Block (CCB)	GPG-158
The FLAGS Word of the CCB	GPG-159
The PIXC Word	GPG-161
The AV Value of the CIXC Word	GPG-162
Packed Source Data	GPG-162
The First Preamble Word	GPG-163
The Second Preamble Word	GPG-163
The Cel Engine Control Word	GPG-163
Function Calls.....	GPG-163

6

Tips and Techniques

Bitmaps and Screens.....	GPG-166
Creating Off-Screen Bitmaps	GPG-166
Using Multiple Screens	GPG-166
Bitmap Item Value	GPG-166
Color Cycling	GPG-167
Cels.....	GPG-167

Cel Rendering Speed	6PG-167
CPU and DMA	GPG-168
When Cel-Display Calls Return	GPG-168
Displaying Text as a Cel	GPG-168
Using Relative Addressing with Cels—Tip	GPG-169
Images.....	GPG-169
Partial Images.....	GPG-169
Image Compression Format	GPG-169
Troubleshooting.....	GPG-170
Problem with Gaps Between Cels	GPG-170
Huge Cels Draw Slowly	GPG-171

3DO Graphics Programmer's Reference

Preface

About This Book.....	vii
About the Audience.....	vii
How This Book Is Organized.....	vii
Related Documentation.....	vii
Typographical Conventions.....	viii

1

Graphics Folio Calls

AddScreenGroup	GPR-6
CloneVRAMPages.....	GPR-7
CloneVRAMPagesDefer *	GPR-9
CloseGraphicsFolio	GPR-11
CopyVRAMPages.....	GPR-12
CopyVRAMPagesDefer	6PR-14
CreateBitmap	GPR-16
CreateScreenGroup.....	GPR-18
CreateVBLIOReq	GPR-23
CreateVRAMIOReq	GPR-24
DeleteBitmap	GPR-25
DeleteScreenGroup	GPR-26
DeleteVBLIOReq	GPR-27
DeleteVDL	GPR-28
DeleteVRAMIOReq	GPR-29
DisableHVG	GPR-30

DisableVAVG	GPR-31
DisplayOverlay	GPR-32
DisplayScreen	GPR-34
DrawCels	GPR-35
DrawChar	GPR-36
DrawScreenCels	GPR-37
DrawText16	GPR-38
DrawText8.....	GPR-39
DrawTo	GPR-40
EnableHAVG	GPR-41
EnableVAVG	GPR-42
FastMapCel	GPR-43
FastMapCelf16	GPR-44
FastMapCellnit	GPR-45
FillRect	GPR-46
GetCurrentFont	GPR-47
GetFirstDisplayInfo	GPR-48
GetPixelAddress	GPR-50
GetVBLIOReq	GPR-51
GetVRAMIOReq	GPR-52
MakeCCBRelative	GPR-53
MakeCLUTBackgroundEntry	GPR-54
MakeCLUTBlueEntry	GPR-55
MakeCLUTColorEntry	GPR-56
 MakeCLUTRedEntry	 GPR-58
MakeRGB15	GPR-59
MakeRGB15Pair	GPR-60
MapCel	GPR-61
ModifyVDL	GPR-62
MoveTo	GPR-63
OpenGraphicsFolio	GPR-64
QueryGraphics	GPR-65
QueryGraphicsList	GPR-67
ReadPixel	GPR-69
RemoveScreenGroup	GPR-70
ResetCurrentFont	GPR-71
ResetReadAddress	GPR-72

ResetScreenColors	6PR-73
SetBGPen	6PR-74
SetCEControl	GPR-75
SetCEWatchDog	GPR-76
SetClipHeight.....	6PR-77
SetClipOrigin.....	GPR-78
SetClipWidth	6PR-79
SetCurrentFontCCB	GPR-80
SetFGPen	GPR-81
SetReadAddress	GPR-82
SetScreenColor.....	GPR-83
SetScreenColors	GPR-84
SetVDL	6PR-85
SetVRAMPages.....	GPR-86
SetVRAMPagesDefer	GPR-87
SubmitVDL	GPR-89
WaitVBL.....	GPR-90
WaitVBLDefer	GPR-91
WritePixel	GPR-92

Glossary

Index

Colophon

Programming 300 Portfolio Music and Audio

List of Figures	xxv
List of Tables	xxvii
Preface	xxix

3DO Music and Audio Programmer's Guide

Preface

About this Book.....	xiii
About the Audience.....	xiii
How this Book Is Organized.....	xiii
Related Documentation.....	xiv
About the Code Examples.....	xv
Typographical Conventions.....	xv

1

How To Write Audio Software

How Do I Make a Simple Sound?.....	MPG-2
How Do I Play an Audio Sample?.....	MPG-2
How Do I Determine When a Sample or Envelope Finishes Playing on a Specific Instrument?.....	MPG-3
How Do I Play a Long Sound File?.....	MPG-4
How Do I Play a Musical Score?.....	MPG-4
How Do I Manage Multiple Sound Effects?.....	MPG-5
How Do I Create Complex Sound Effects?.....	MPG-6
How Do I Make a Sound Fade Out Smoothly without Popping?.....	MPG-6

2

Understanding 3DO Audio

What is 300 Audio?	MPG-10
Audio Hardware.....	MPG-10
The CD Player	MPG-11
The Digital Signal Processor (DSP)	MPG-12
Audio Software.....	MPG-15
Designing Sound with the Audio Folio	MPG-15
Playing Notes.....	MPG-18
Using the Music Library	MPG-20

Preparing Instruments

Introduction.....	MPG-24
The Process of Using Instruments.....	MPG-24
Opening the Audio Folio.....	MPG-26
Preparing Instruments.....	MPG-26
Loading an Instrument Template	MPG-34
Tuning Instruments.....	MPG-35
Creating a Tuning	MPG-36
Applying a Tuning	MPG-38
Deleting a Tuning	MPG-38
Bending Pitch	MPG-39
Loading and Attaching Samples.....	MPG-40
Loading Samples from Disk.....	MPG-40
Defining a Data-Streamed Sample	MPG-46
Sample Loops	MPG-46
Sample Trigger Points	MPG-47
Attaching a Sample to an Instrument.....	MPG-47
The Attachment Item	MPG-48
Attaching Multisamples to an Instrument.....	MPG-48
Detaching a Sample from an Instrument	MPG-49
Deleting a Sample	MPG-49
Debugging a Sample	MPG-50
Reading and Changing Audio Item Attributes.....	MPG-50
Reading Audio Item Characteristics.....	MPG-50
Setting Audio Item Characteristics.....	MPG-51
Creating and Attaching Envelopes.....	MPG-51
Envelope Players	MPG-51
Envelope Properties.....	MPG-52
Creating an Envelope	MPG-56
Attaching an Envelope to an Instrument	MPG-57
Detaching an Envelope from an Instrument.....	MPG-58
Deleting an Envelope	MPG-58
Modifying Attachments.....	MPG-58
Setting an Attachment's Attributes	MPG-58
Specifying a Start Point	MPG-59
Setting a Start-Independent Attachment.....	MPG-60
Creating an Instrument-Stopping Attachment	MPG-60

Linking Attachments	MPG-61
Connecting Instruments.....	MPG-62
Connecting One Instrument to Another	MPG-62
Disconnecting One Instrument From Another.....	MPG-63
Adding Reverberation.....	MPG-63
A Delay Line Overview	MPG-63
Creating a Delay Line	MPG-65
Deleting a Delay Line	MPG-65
Connecting a Delay Line to Create Reverberation	MPG-66
Using a Delay Line for Oscilloscope Data	MPG-68
Handling Knobs.....	MPG-68
Finding Knobs	MPG-68
Grabbing a Knob	MPG-69
Finding Knob Parameters	MPG-69
Tweaking a Knob	MPG-70
Tweaking a Knob to a Raw Value	MPG-70
Releasing a Knob	MPG-72
Using Probes.....	MPG-72
Creating Probes	MPG-72
Reading Probes	MPG-72
Deleting Probes	MPG-73
Allocating Amplitude.....	MPG-73
Adjusting Amplitude in a Mixer.....	MPG-74
Allocating System Amplitude	MPG-75
Freeing System Amplitude	MPG-75
Example Programs.....	MPG-76
Primary Data Structures.....	MPG-76
Function Calls.....	MPG-76
Audio Folio Control	MPG-76
Instrument Loading and Creation	MPG-77
Tuning	MPG-77
Loading, Creating, and Attaching Samples	MPG-77
Reading and Changing Audio-Item Attributes	MPG-78
Envelopes	MPG-78
Linking Attachments	MPG-78
Connecting Instruments	MPG-78
Adding Reverberation	MPG-79
Handling Knobs	MPG-79
Allocating Amplitude	MPG-79

Playing Instruments

Playing Instruments.....	MPG-81
Starting an Instrument.....	MPG-82
Releasing an Instrument.....	MPG-85
Stopping an Instrument.....	MPG-85
Cleaning Up When Finished	MPG-86
Independently Controlling Attachments.....	MPG-87
Starting an Attachment.....	MPG-87
Releasing an Attachment.....	MPG-88
Stopping an Attachment	MPG-88
Timing Audio Events.....	MPG-88
The Audio Clock	MPG-89
Timed Notification	MPG-89
Attachment Timing	MPG-91
Controlling the Audio Clock.....	MPG-93
Owning the Audio Clock	MPG-93
Setting the Audio Clock Rate	MPG-94
Setting Audio Tick Duration	MPG-94
Checking the Current Audio Clock Setting	MPG-95
Example Programs.....	MPG-95
Function Calls.....	MPG-103
Using Instruments	MPG-103
Playing Attachments	MPG-103
Timing Audio Events^.....	MPG-103
Audio Clock Control.....	MPG-104

5

Playing Sound Files

An Overview of the Sound File Player.....	MPG-106
Sound File Player Operation	MPG-107
Operating the Sound File Player.....	MPG-107
Creating a Sound File Player.....	MPG-108
Preloading a Sound File	MPG-108
Creating a Sound File Player and Preloading a File	MPG-109
Specifying an Output Mixer	MPG-109
Starting Sound File Playback	MPG-110
Servicing a Sound File	MPG-110
Stopping Sound File Playback	MPG-113

Rewinding a Sound File Player	MPG-U3
Cleaning Up After Playback	MPG-113
Typical Calling Sequences	MPG-115
An Example Program.....	MPG-116
Primary Data Structures.....	MPG-120
Controlling Sound Playback	MPG-120
Function Calls.....	MPG-122
Convenience Calls	MPG-122
Basic Calls	MPG-122

6

Advanced Sound Player

Overview of the Advanced Sound Player.....	MPG-123
Simple Advanced Sound Player Example	MPG-124
Players, Sounds, and Markers.....	MPG-125
Players.....	MPG-125
Sounds	MPG-125
Markers	MPG-126
Examples of Looping and Branching	MPG-126
Decision Functions.....	MPG-128
Decision Functions and Actions	MPG-129
Example Decision Function	MPG-130
Rules for Decision Functions	MPG-131
Caveats.....	MPG-131
For More Information	MPG-132
Function Calls.....	MPG-132
Player Management Function Calls	MPG-132
Sound Management Function Calls	MPG-133
Marker Management Function Calls	MPG-133
Decision Function Function Calls	MPG-134
Debug Function Calls	MPG-134
Constants	MPG-134

7

Using the Sound Spooler

How the Sound Spooler Works.....	MPG-138
How to Use the Sound Spooler.....	MPG-138
An Example.....	MPG-139
Convenience Functions.....	MPG-143

ssplSpoolData	MPG-143
ssplPlayData	MPG-143
Sound Spooler Function Calls.....	MPG-144
Convenience calls	MPG-144
SoundSpooler management calls	MPG-144
SoundBufferNode management calls	MPG-145
Low level calls	MPG-145
Callback routine	MPG-146

8

Creating and Playing Juggler Objects

Object-Oriented Programming Concepts.....	MPG-148
Objects	MPG-148
Methods and Messages	MPG-148
Object Variables.....	MPG-149
Classes and Instances	MPG-149
Creating New Classes	MPG-150
Managing Juggler Objects.....	MPG-150
Initializing the Juggler.....	MPG-150
Defining a New Class	MPG-151
Creating an Object	MPG-151
Destroying an Object	MPG-151
Checking an Object's Validity	MPG-152
Default Juggler Classes.....	MPG-152
The Jugglee Class.....	MPG-152
The Sequence Class	MPG-153
The Collection Class	MPG-154
Creating Sequences and Collections.....	MPG-154
Creating a Sequence	MPG-155
Creating a Collection	MPG-159
Sending Messages to Sequences and Collections.....	MPG-162
Calling Methods Directly	MPG-162
Message Macros	MPG-162
- Messages for Sequences	MPG-162
Messages for Collections.....	MPG-165
Playing Sequences and Collections.....	MPG-167
A Juggler Operational Overview	MPG-167
The Juggler Process	MPG-169
Bumping the Juggler	MPG-169

Terminating the Juggler.....	MPG-170
An Example Program.....	MPG-170
Function Calls and Method Macros.....	MPG-174
Juggler Control Calls	MPG-174
Object Management Calls	MPG-175
Method Macros	MPG-175
Object-Defining Tag Arguments.....	MPG-175

9

Playing MIDI Scores

A Brief MIDI Review.....	MPG-179
MIDI Channels	MPG-179
Channel Messages	MPG-180
MIDI Score Playback	MPG-180
Creating an Internal MIDI Environment.....	MPG-181
Creating a Virtual MIDI Device	MPG-181
Importing a MIDI Score	MPG-183
Providing MIDI Playback Functions	MPG-184
Setting the Audio Clock Speed	MPG-185
An Overview of the MIDI Score-Playing Process.....	MPG-185
Setting Up.....	MPG-186
Creating a MIDI Environment.....	MPG-186
Setting Voice and Program Limits	MPG-186
Creating a Score Context	MPG-187
Setting PIMap Entries	MPG-188
Setting Up a Mixer	MPG-192
Importing a MIDI Score.....	MPG-194
Declaring a MIDIFileParser Data Structure	MPG-194
Creating a Juggler Object.....	MPG-194
Loading the Score	MPG-195
Specifying the User Context.....	MPG-196
The Interpreter Procedure	MPG-197
Setting the Tempo.....	MPG-197
Setting the Audio Clock Rate	MPG-197
Scaling MIDI Time Stamps	MPG-198
Playing the MIDI Score.....	MPG-199
Dynamic Voice Allocation	MPG-200
Freeing Instruments Created by Voice Allocation	MPG-201
Using MIDI Functions.....	MPG-202

Interpreting and Executing a MIDI Message	MPG-202
Starting and Releasing a Note.....	MPG-203
Directly Starting and Releasing an Instrument.....	MPG-204
Changing a Program	MPG-204
Setting Channel Panning and Volume	MPG-205
Bending Pitch	MPG-205
Changing Characteristics During Playback.....	MPG-208
Cleaning Up.....	MPG-208
Creating a MIDI Score for Playback.....	MPG-210
An Example Program.....	MPG-210
Primary Data Structures.....	MPG-223
Function Calls.....	MPG-223
MIDI Environment Calls	MPG-223
MIDI Score Calls	MPG-224
MIDI Playback Calls	MPG-224

10

Tips and Techniques

DSP Resources.....	MPG-226
Maximum Number of Audio Voices	MPG-226
Allocating DSP Resources	MPG-226
DSP DMA	MPG-226
DSP Usage Monitor	MPG-227
Dynamic Voice Allocation	MPG-227
Managing Sound Effects.....	MPG-228
Loading an Instrument :.*.....	MPG-228
Attaching Multiple Samples to an Instrument.....	MPG-228
Audio Samples.....	MPG-229
Available RAM for Sampling	MPG-229
Looping Stereo and Mono Sound	MPG-229
Looping AIFFs	MPG-229
Translating PC VOX files.....	MPG-230
Loading Multiple Sound Files.....	MPG-230
Audio Samples and Pre-specified Memory	MPG-230
Playback Rate	MPG-231
Playing Scores.....	MPG-231
Streaming Audio versus Score Playback	MPG-231
Score Files	MPG-232
MIDI	MPG-233

Streaming Sound Files.....	MPG-233
Streaming And Spooling	MPG-233
Streaming Audio Sample Code	MPG-234
Playing Red Book Audio	MPG-234
Timing.....	MPG-234
DSP Time versus System Time	MPG-234
SMPTE.....	MPG-235
Multi-thread Timers	MPG-235
Troubleshooting.....	MPG-235
High Frequency Noise	MPG-235
ADPCM Error	MPG-235
22 kHz Sound Files Play at 44 kHz	MPG-236
SleepAudioTicksQ and Application Slowdown—Tip	MPG-236
Timing Problems When Spooling	MPG-237
Filters.....	MPG-237
Setting Cutoff Frequency	MPG-237
Modulating Selected Frequency Ranges	MPG-237
Miscellaneous.....	MPG-238
Allocating Signals	MPG-238
3-D sound	MPG-238

3DO Music and Audio Programmer's Reference

Preface

About this Book...;?!.....	xiii
About the Audience.....	xiii
How this Book is Organized.....	xiii
Related Documentation.....	xiii
Typographical Conventions.....	xiv

1

Audio Folio Calls

AbandonInstrument	MPR-6
AbortTimerCue	MPR-7
AdoptInstrument.....	MPR-8
AllocAmplitude	MPR-9
AllocInstrument	MPR-10
AttachEnvelope	MPR-11
AttachSample	MPR-13

AudioTimeLaterThan.....	MPR-14
AudioTimeLaterThanOrEqual	MPR-15
BendInstrumentPitch	MPR-16
CloseAudioFolio	MPR-17
CompareAudioTimes	MPR-18
ConnectInstruments	MPR-19
Convert12TET_F16	MPR-21
CreateCue	MPR-22
CreateDelayLine.....	MPR-23
CreateEnvelope	MPR-24
CreateInsTemplate.....	MPR-26
CreateInstrument.....	MPR-28
CreateProbe.....	MPR-30
CreateSample	MPR-31
CreateTuning	MPR-37
DebugSample	MPR-38
DefineInsTemplate	MPR-39
DefineSampleHere	MPR-41
DeleteCue	MPR-42
DeleteDelayLine	MPR-43
DeleteEnvelope	MPR-44
DeleteInstrument	MPR-45
DeleteProbe	MPR-46
DeleteTuning.....	MPR-47
DetachEnvelope	MPR-48
DetachSample.....	MPR-49
DisconnectInstruments.....	MPR-50
DisownAudioClock.....	MPR-51
EnableAudioInput	MPR-52
FreeAmplitude.....	MPR-53
FreeInstrument	MPR-54
GetAudioCyclesUsed.....	MPR-55
GetAudioDuration	MPR-56
GetAudioFolioInfo	MPR-57
GetAudioFrameCount.....	MPR-59
GetAudioItemInfo	MPR-60
GetAudioRate.....	MPR-62
GetAudioTime	MPR-63

QetCueSignal	MPR-64
GetKnobName	MPRJ5
QetNumKnobs	
GrabKnob	
LinkAttachments	MPR-6
LoadInsTemplate	MPR-70
LoadInstrument	MPR-71
LoadSample	MPR-7
LoadSampleHere	
MakeSample	
MonitorAttachment	MPR-7
OpenAudioFolio	MPR-78
OwnAudioClock	MPR-79
PauseInstrument	MPR-80
ReadProbe	MPR-81
ReleaseAttachment	MPR-82
ReleaseInstrument	MPR-83
ReleaseKnob	MPR-85
ResumeInstrument	MPR-86
ScanSample	MPR-87
SetAudioDuration	MPR-88
SetAudioItemInfo	MPR-89
SetAudioRate	MPR-90
SignalAtTime	MPR-92
SleepAudioTicks	MPR-93
SleepUntilTime	MPR-94
StartAttachment	MPR-95
StartInstrument	MPR-96
StopAttachment	MPR-98
StopInstrument	MPR-99
TraceAudio	MPR-100
TuneInsTemplate	MPR-102
TuneInstrument	MPR-103
TweakKnob	MPR-104
TweakRawKnob	MPR-105
UnloadInsTemplate	MPR-106
UnloadInstrument	MPR-107
UnloadSample	MPR-108

GetCueSignal	MPR-64
GetKnobName.....	MPR-65
GetNumKnobs	MPR-66
GrabKnob	MPR-67
LinkAttachments	MPR-68
LoadInsTemplate	MPR-70
LoadInstrument	MPR-71
LoadSample.....	MPR-72
LoadSampleHere	MPR-73
MakeSample	MPR-75
MonitorAttachment.....	MPR-77
OpenAudioFolio	MPR-78
OwnAudioClock	MPR-79
PauseInstrument	MPR-80
ReadProbe	MPR-81
ReleaseAttachment.....	MPR-82
ReleaseInstrument	MPR-83
ReleaseKnob	MPR-85
ResumeInstrument	MPR-86
ScanSample	MPR-87
SetAudioDuration	MPR-88
SetAudioItemInfo	MPR-89
SetAudioRate	MPR-90
SignalAtTime	MPR-92
SleepAudioTicksT.....	MPR-93
SleepUntilTime	MPR-94
StartAttachment	MPR-95
StartInstrument	MPR-96
StopAttachment.....	MPR-98
StopInstrument.....	MPR-99
TraceAudio	MPR-100
TuneInsTemplate	MPR-102
TuneInstrument	MPR-103
TweakKnob	MPR-104
TweakRawKnob	MPR-105
UnloadInsTemplate	MPR-106
UnloadInstrument.....	MPR-107
UnloadSample	MPR-108

WhereAttachment	MPR-109
Attachment	MPR-110
Cue	MPR-113
Envelope	MPR-115
Instrument.....	MPR-118
Knob	MPR-122
Probe	MPR-124
Sample	MPR-125
Template	MPR-130
Tuning	MPR-132

2

Music Library Calls

Juggler	MPR-136
Score Playing	MPR-137
Sound Files	MPR-138
Sound Player	MPR-139
Sound Spooler.....	MPR-141
CollectionClass	MPR-143
JuggleClass.....	MPR-144
SequenceClass	MPR-145
AbortObject.....	MPR-146
AllocObject	MPR-147
BumpJuggler	MPR-148
CreateObject	MPR-150
DefineClass	MPR-151
DestroyObject	MPR-152
FreeObject.....	MPR-153
GetNthFromObject	MPR-154
GetObjectInfo.....	MPR-155
InitJuggler	MPR-156
PrintObject.....	MPR-157
RemoveMthFromObject.....	MPR-158
SetObjectInfo	MPR-159
StartObject.....	MPR-162
StopObject.....	MPR-163
TermJuggler	MPR-164
ValidateObject.....	MPR-165
ChangeScoreControl.....	MPR-166

ChangeScorePitchBend	MPR-167
ChangeScoreProgram.....	MPR-169
ConvertPitchBend	MPR-170
CreateScoreContext	MPR-172
DeleteScoreContext	MPR-173
DisableScoreMessages.....	MPR-174
FreeChannellInstruments	MPR-175
GetScoreBendRange	MPR-176
InitScoreDynamics	MPR-177
InitScoreMixer	MPR-178
InterpretMIDIEvent	MPR-180
InterpretMIDIMessage	MPR-181
LoadPIMap	MPR-183
MFDefineCollection	MPR-184
MFLoadCollection	MPR-185
MFLoadSequence	MPR-186
MFUnloadCollection	MPR-187
NoteOffIns	MPR-188
NoteOnIns.....	MPR-189
PurgeScoreInstrument	MPR-190
ReleaseScoreNote.....	MPR-192
SelectSamplePlayer	MPR-193
SetPIMapEntry	MPR-194
SetScoreBendRange	MPR-196
StartScoreNote	MPR-197
StopScoreNote	MPR-198
TermScoreMixer	MPR-199
UnloadPIMap	MPR-200
CloseSoundFile	MPR-201
CreateSoundFilePlayer.....	MPR-202
DeleteSoundFilePlayer	MPR-203
LoadSoundFile	MPR-204
OpenSoundFile	MPR-206
ReadSoundFile.....	MPR-207
RewindSoundFile	MPR-208
ServiceSoundFile	MPR-209
StartSoundFile	MPR-210
StopSoundFile	MPR-211

UnloadSoundFile.....	MPR-212
spAddMarker	MPR-213
spAddSample	MPR-215
spAddSoundFile.....	MPR-218
spBranchAtMarker.....	MPR-221
spClearDefaultDecisionFunction	MPR-223
spClearMarkerDecisionFunction	MPR-224
spContinueAtMarker.....	MPR-225
spCreatePlayer.....	MPR-226
SPDecisionFunction.....	MPR-229
spDeletePlayer.....	MPR-232
spDumpPlayer.....	MPR-233
spFindMarkerName.....	MPR-234
spGetMarkerName	MPR-235
spGetMarkerPosition	MPR-236
spGetPlayerFromSound	MPR-237
spGetPlayerSignalMask	MPR-238
spGetPlayerStatus	MPR-239
spGetSoundFromMarker.....	MPR-241
spIsSoundInUse	MPR-242
spLinkSounds	MPR-243
spLoopSound	MPR-244
spPause	MPR-245
spRemoveMarker	MPR-246
spRemoveSound	MPR-247
spResume	MPR-249
spService.....	MPR-250
spSetBranchAction	MPR-252
spSetDefaultDecisionFunction	MPR-253
spSetMarkerDecisionFunction	MPR-255
spSetStopAction	MPR-256
spStartPlaying	MPR-257
spStartReading	MPR-259
spStop	MPR-261
spStopAtMarker.....	MPR-262
SoundBufferFunc	MPR-263
ssplAbort	MPR-266
ssplAttachInstrument	MPR-267

ssplCreateSoundSpooler	MPR-268
ssplDeleteSoundSpooler.....	MPR-270
ssplDetachInstrument.....	MPR-271
ssplDumpSoundBufferNode	MPR-272
ssplDumpSoundSpooler.....	MPR-273
ssplGetSBMsgClass.....	MPR-274
ssplGetSequenceNum.....	MPR-275
ssplGetSpoolerStatus	MPR-276
ssplGetUserData	MPR-278
ssplIsSpoolerActive	MPR-279
ssplPause	MPR-280
ssplPlayData	MPR-281
ssplProcessSignals.....	MPR-282
ssplRequestBuffer	MPR-284
ssplReset	MPR-287
ssplResume	MPR-289
ssplSendBuffer	MPR-290
ssplSetBufferAddressLength	MPR-292
ssplSetSoundBufferFunc	MPR-293
ssplSetUserData	MPR-294
ssplSpoolData.....	MPR-295
ssplStartSpooler	MPR-296
ssplStartSpoolerTags	MPR-297
ssplStopSpooler	MPR-298
ssplUnrequestBufat.....	MPR-299
UserBufferprocessor	MPR-300

3

Instrument Templates

add.dsp	MPR-307
adpcmhalfmono.dsp	MPR-308
adpcmmono.dsp	MPR-309
adpcmvarmono.dsp	MPR-310
benchmark.dsp	MPR-311
dcsqxdhalfmono.dsp	MPR-312
dcsqxdhalfstereo.dsp.....	MPR-313
dcsqxdmono.dsp	MPR-314
dcsqxdstereo.dsp	MPR-315
dcsqxdvarmono.dsp	MPR-316

decodeadpcm.dsp	MPR-317
deemphcd.dsp	MPR-318
delayltap.dsp	MPR-319
delaymono.dsp	MPR-320
delaystereo.dsp	MPR-321
directin.dsp.....	MPR-322
directout.dsp	MPR-323
envelope.dsp	MPR-324
envfollower.dsp	MPR-325
filterednoise.dsp	MPR-326
fixedmonoS.dsp	MPR-327
fixedmonosample.dsp	MPR-328
fixedstereo16swap.dsp	MPR-329
fixedstereoS.dsp	MPR-330
fixedstereosample.dsp	MPR-331
halfmonoS.dsp.....	MPR-332
halfmonosample.dsp	MPR-333
halfstereoS.dsp	MPR-334
halfstereosample.dsp.....	MPR-335
head.dsp	MPR-336
impulse.dsp	MPR-337
maximum.dsp.....	MPR-338
minimum.dsp	MPR-339
mixer12x2.dsp	MPR-340
mixer2x2.dsp	MPR-342
mixer4x2.dsp	MPR-343
mixer8x2.dsp	MPR-344
mixer8x2amp.dsp	MPR-346
multiply.dsp	MPR-348
noise.dsp	MPR-349
oscupdownfp.dsp	MPR-350
pulse.dsp	MPR-351
pulsejfo.dsp	MPR-352
pulser.dsp	MPR-353
randomhold.dsp	MPR-354
rednoise.dsp	MPR-355
sampler.dsp	MPR-356
samplerenv.dsp	MPR-357

samplermod.dsp	MPR-358
sawenv.dsp.....	MPR-359
sawenvsvlenv.dsp	MPR-360
sawfilterednoise.dsp	MPR-361
sawfilteredsaw.dsp	MPR-362
sawtooth.dsp	MPR-363
square.dsp	MPR-364
squarejfo.dsp	MPR-365
submixer2x2.dsp	MPR-366
submixer4x2.dsp	MPR-367
submixer8x2.dsp	MPR-369
subtract.dsp.....	MPR-371
svfilter.dsp	MPR-372
tail.dsp	MPR-374
tapoutput.dsp	MPR-375
timesplus.dsp	MPR-377
triangle.dsp	MPR-378
trianglejfo.dsp.....	MPR-379
varmono16.dsp	MPR-380
varmonoS.dsp	MPR-381

Glossary

Index

Colophon

Programming the 300 Portfolio System

List of Figures	v
List of Tables	vii
Preface.....	ix

300 System Programmer's Guide

Preface

About This Book.....	xv
About The Audience.....	xv
How This Book is Organized.....	xv
Related Documentation.....	xvi
About the Code Examples.....	xvii
Typographical Conventions.....	xvii

1

Understanding the Kernel

Description of the Kernel.....	SPG-2
Multitasking.....	SPG-2
Privileged and Non-Privileged Tasks	SPG-3
Multitasking	SPG-3
Task Termination	SPG-5
Parent Tasks, Child fcsks, and Threads	SPG-5
Managing Memory.....	SPG-5
Types of Memory	SPG-5
Memory Size.....	SPG-6
Allocating Memory.....	SPG-7
Memory Fences	SPG-9
Returning Memory.....	SPG-10
Sharing Memory	SPG-10
Working With Folios.....	SPG-10
Managing Items.....	SPG-11
Items	SPG-11
Creating an Item	SPG-12
Opening Items	SPG-12
Using Items	SPG-12

Deleting Items	SPG-13
Semaphores.....	SPG-13
Intertask Communication.....	SPG-14
Signals.....	SPG-14
Messages	SPG-15
Portfolio Error Codes.....	SPG-17

2

Tasks and Threads

Overview.....	SPG-19
What Is aTask?.....	SPG-20
Starting and Ending Tasks.....	SPG-20
Creating a Task	SPG-20
Ending aTask	SPG-22
Controlling the State of a Task.....	SPG-22
Yielding the CPU to Another Task	SPG-23
Going to and From the Waiting Queue	SPG-23
Changing a Task's Parent	SPG-24
Changing Task Priorities	SPG-24
Starting and Ending Threads.....	SPG-25
Creating a Thread	SPG-25
Ending aThread	SPG-26
Advanced Thread Usage	SPG-26
Loading and Executing Separate Code Modules.....	SPG-27
Example.....^.....	SPG-29
Function Calls.....	SPG-32
Starting Tasks	SPG-32
Ending Tasks	SPG-32
Loading and Unloading Code	SPG-32
Starting Threads	SPG-32
Ending Threads.....	SPG-33
Controlling Tasks	SPG-33

Using Tags and TagArgs

About Tags and TagArgs.....	SPG-36
Tags and TagArgs.....	SPG-36
Special Tag Commands.....	SPG-36
Tags and VarArgs.....	SPG-37
Parsing Tags.....	SPG-38
TagArg Function Calls.....	SPG-39

4

Managing Linked Lists

About Linked Lists.....	SPG-42
Characteristics of Portfolio Lists.....	SPG-42
Characteristics of Nodes	SPG-43
Creating and Initializing a List.....	SPG-43
Adding a Node to a List.....	SPG-44
Adding a Node to the Head of a List	SPG-44
Adding a Node to the Tail of a List.....	SPG-44
Adding a Node After Another Node in a List	SPG-44
Adding a Node Before Another Node in a List.....	SPG-45
Adding a Node According to Its Priority	SPG-45
Adding a Node According to Other Node Values	SPG-46
Changing the Priority of a Node.....	SPG-46
Removing a Node From a List.....	SPG-46
Removing the First Node	SPG-46
Removing the Last Node	SPG-47
Removing a Specific Node	SPG-47
Finding out If a List Is Empty.....	SPG-47
Traversing a List.....	SPG-47
Traversing a List From Front to Back	SPG-47
Traversing a List From Back to Front.....	SPG-48
Finding a Node by Name.....	SPG-49
Finding a Node by Its Ordinal Position.....	SPG-49
Determining the Ordinal Position of a Node.....	SPG-50
Counting the Number of Nodes in a List.....	SPG-50
Example.....	SPG-51
Primary Data Structures.....	SPG-51
The Node Structure.....	SPG-51
MinNode and NamelessNode Structures	SPG-52

The List Data Structure	SPG-53
TheListAnchorUnion	SPG-53
The Link Data Structure	SPG-54
Function Calls.....	SPG-54
Initializing a List	SPG-54
Adding Nodes to a List	SPG-54
Changing the Priority of a List Node	SPG-55
Removing Nodes From a List	SPG-55
Checking to See If a List Is Empty	SPG-55
Testing Nodes and Lists	SPG-55
Traversing a Linked List	SPG-55
Finding a Node by Name	SPG-56
Ordinal Node Position Functions	SPG-56
Counting the Nodes in a List	SPG-56

5

Managing Memory

About Memory.....	SPG-58
Types of Memory	SPG-58
Organization of Memory	SPG-59
How Memory Allocation Works	SPG-61
Guidelines for Using Memory	SPG-70
Allocating Memory.....	SPG-70
Allocating a Memory Block	SPG-71
Freeing a Memory Block	SPG-72
Allocating Memory in Programs Ported From Other Platforms	SPG-73
Freeing Memory in Programs Ported From Other Platforms	SPG-73
Getting Information About Memory.....	SPG-73
Finding Out How Much Memory Is Available	SPG-73
Getting the Memory Type of a Memory Location	SPG-74
Getting the Size of a Memory Page	SPG-75
Finding Out Which VRAM Bank Contains a VRAM Location	SPG-75
Validating Memory Pointers	SPG-76
Reclaiming Memory for Other Tasks.....	SPG-76
Allowing Other Tasks to Write to Your Memory.....	SPG-76
Transferring Memory to Other Tasks.....	SPG-77
Using Private Memory Lists.....	SPG-77
Creating a Private Memory List.....	SPG-78
Adding Memory to a Private Memory List	SPG-78

Allocating Memory From a Private Memory List.....	SPG-78
Freeing Memory to a Private Memory List.....	SPG-79
Deallocating a Private Memory List	SPG-79
Sharing Private Memory Lists Among Threads	SPG-79
Using Private Memory Pools.....	SPG-79
Creating a Memory Pool	SPG-80
Allocating Memory From a Specific Memory Pool.....	SPG-80
Freeing Memory to a Specific Memory Pool	SPG-81
Getting Memory From the System-Wide Free Memory Pool.....	SPG-81
Debugging Memory Usage.....	SPG-81
Example.....	SPG-83
Function Calls.....	SPG-85
Allocating Memory.....	SPG-85
Freeing Memory	SPG-85
Getting Information About Memory	SPG-85
Reclaiming Unused Memory.....	SPG-85
Changing the Control Status of Memory.....	SPG-86
Using Private Memory Lists	SPG-86
Using Memory Pools	SPG-86
Debugging Memory Problems	SPG-86

6

Managing Items

About Items.....	SPG-88
Creating or Opening an Item	SPG-89
Creating an Item *	SPG-89
Specifying the Item Type	SPG-90
Tag Arguments	SPG-90
VarArgs Tag Arguments	SPG-92
The Item Number.....	SPG-93
The Easy Way Out.....	SPG-93
Using an Item.....	SPG-94
Finding an Item	SPG-94
Getting a Pointer to an Item	SPG-94
Checking If an Item Exists	SPG-95
Changing Item Ownership	SPG-95
Changing an Item's Priority	SPG-95
Deleting an Item	SPG-96
Opening an Item.....	SPG-96

Closing an Item.....	SP6-96
Function Calls.....	SPG-97
Creating Items	SPG-97
Opening Items	SPG-97
Managing Items	SPG-98
Closing and Deleting Items	SPG-98

7

Sharing System Resources

About Semaphores.....	SPG-100
The Problem: Sharing Resources Safely	SPG-100
The Solution: Locking Shared Resources When Using Them	SPG-100
The Implementation: Semaphores	SPG-100
Using Semaphores	SPG-100
Creating a Semaphore.....	SPG-101
Locking a Semaphore.....	SPG-101
Unlocking a Semaphore.....	SPG-102
Finding a Semaphore.....	SPG-102
Deleting a Semaphore.....	SPG-102
Function Calls.....	SPG-103
Creating a Semaphore	SPG-103
• Locking a Resource	SPG-103
Unlocking a Resource	SPG-103
Finding a Semaphore	SPG-103
Deleting a Semaphore	SPG-103

8

Communicating Among Tasks

About Intertask Communication.....	SPG-106
Using Signals.....	SPG-106
Allocating a Signal Bit	SPG-106
Receiving a Signal	SPG-107
Sampling and Changing the Current Signal Bits	SPG-108
Sending a Signal.....	SPG-108
Freeing Signal Bits	SP6-109
Sample Code for Signals	SPG-109
Passing Messages.....	SPG-112
Creating a Message Port.....	SPG-113
Creating a Message	SPG-113

Sending a Message.....	SP6-116
Receiving a Message	SPG-117
Working With a Message	SPG-118
Pulling Back a Message	SP6-119
Replying to a Message.....	SPG-120
Finding a Message Port	SPG-121
Example Code for Messages.....	SPG-121
Function Calls.....	SPG-124
Allocating Signals	SPG-124
Creating Messages and Message Ports	SPG-125
Deleting Messages and Message Ports	SPG-125
Finding a Message Port	SPG-125
Sending Messages.....	SPG-126
Receiving Messages	SPG-126
Replying to Messages	SPG-126

9

The Portfolio I/O Model

Introduction.....	SPG-128
Hardware Devices and Connections.....	SPG-128
Internal Buses.....	SPG-128
The Expansion Port	SPG-129
The Control Port	SPG-129
I/O Architecture.....	SPG-130
Devices	SPG-130
Drivers	SPG-131
I/O Requests (IOReqs).....	SPG-132
Performing I/O.....	SPG-132
Preparing for I/O	SPG-133
Creating an IOReq	SPG-133
Initializing an IOInfo Structure	SPG-134
Passing IOInfo Values to the Device	SPG-136
Asynchronous vs. Synchronous I/O	SPG-136
Completion Notification	SPG-138
Reading an IOReq	SPG-139
Continuing I/O	SPG-140
Aborting I/O	SPG-140
Finishing I/O	SPG-140

Examples.....	SPG-141
Reading the System Time	SPG-141
Using the Timer Device to Wait	SPG-143
Function Calls.....	SPG-144
Opening and Closing Devices	SPG-144
Creating, Deleting, and Creating Pointers for IOReqs	SPG-145
Controlling the I/O Process	SPG-145

10 Portfolio Devices

Introduction.....	SPG-148
Communicating With Devices.....	SPG-148
The Timer Device.....	SPG-149
Working With the Timer Device	SPG-149
The Microsecond Unit.....	SPG-149
The Vertical Blank Unit.....	SPG-152
High-Performance Timing	SPG-155
Time Arithmetic	SPG-156
Simplified Timer Device Interface	SPG-156
The SPORT Device.....	SPG-157
Working With the SPORT Device	SPG-157
Copying VRAM Pages	SPG-157
Replicating VRAM Pages	SPG-158
Setting VRAM Pages to a Fixed Value	SPG-159
The File Pseudo-Device.....	SPG-160
Getting Filesystem Status	SPG-160
Getting File Status.....	SPG-161
Creating and Deleting Files	SPG-162
Allocating Blocks for a File	SPG-162
Writing Data to a File	SPG-162
Marking the End of a File	SPG-163
Reading Data From a File	SPG-163
Getting Directory Information	SPG-164
Getting the Path of a File	SPG-164

The Filesystem and the File Folio

3DO File System Interface.....	SPG-166
Design Philosophy	SPG-166
Files	SPG-166
Directories	SPG-167
Avatars	SPG-167
Pathnames and Filenames.....	SPG-168
Aliases.....	SPG-169
Defining Aliases	SPG-169
Storing Aliases	SPG-170
Predefined Aliases	SPG-170
Byte-Stream File Access.....	SPG-171
Opening a File	SPG-171
Reading the Data	SPG-172
Setting Up the Read Position	SPG-172
Cleaning Up	SPG-172
Accessing Directories.....	SPG-173
Opening the Directory	SPG-173
Reading the Directory	SPG-173
Cleaning Up	SPG-174
Finding Mounted File Systems	SPG-174
The Current Directory.....	SPG-174
Finding the Current Directory.....	SPG-174
Changing the Current Directory	SPG-174
Loading and Executing an Application.....	SPG-175
Loading and Launching a Task	SPG-175
Loading a Code Module	SPG-176
Executing a Loaded Code Module.....	SPG-176
Unloading a Code Module	SPG-177
Working With NVRAM.....	SPG-177
How Not to Manage NVRAM	SPG-177
How to Manage NVRAM	SPG-178
How a Title Should Save Its Status	SPG-178
Working With NVRAM Files	SPG-178
NVRAM Chunk Definition	SPG-180
File System Folio Function Calls.....	SPG-181
File Access Calls	SPG-181

Program Calls	SPG-181
Directory Access Calls	SPG-182
Byte-Stream Calls	SPG-182
Examples.....	SPG-183
Using NVRAM	SPG-183
Displaying Contents of a File	SPG-186
Scanning the File System	SPG-189
Listing the Contents of a Directory	SPG-192

12 The Event Broker

About the Event Broker.....	SPG-196
Specifying and Monitoring Events	SPG-196
Working With Input Focus	SPG-197
Reconfiguring or Disconnecting an Event Broker Connection	SPG-198
Other Event Broker Activities	SPG-198
Sending Messages Between Tasks and the Event Broker.....	SPG-198
Message Flavors	SPG-198
Message Types	SPG-201
Flavor-Specific Message Requirements	SPG-202
The Process of Event Monitoring.....	SPG-202
Connecting a Task to the Event Broker.....	SPG-203
Creating a Message Port.....	SPG-203
Creating a Configuration Message	SPG-203
Creating a Configuration Block	SPG-203
Sending the Configuration Message	SPG-208
Receiving the Configuration Reply Message.....	SPG-209
Monitoring Events Through the Event Broker.....	SPG-209
Waiting for Event Messages on the Reply Port	SPG-209
Reading an Event Message Data Block	SPG-211
Reading Event Data	SPG-214
High-Performance Event Broker Use.....	SPG-219
The Pod Table	SPG-219
Gaining Access to the Pod Table	SPG-219
Relinquishing Access to the Pod Table	SPG-219
Structure of the Pod Table	SPG-220
Empty Generic Class Substructures	SPG-221
Non-Empty Generic Class Substructures	SPG-221
Use and Abuse of the Pod Table	SPG-223

Synchronization Between the Pod Table and Event Messages	SPG-225
Reconfiguring or Disconnecting a Task.....	SPG-225
Reconfiguring the Event Broker Connection	SPG-225
Disconnecting From the Event Broker.....	SPG-226
Other Event Broker Activities.....	SPG-226
Getting Lists of Listeners and Connected Pods	SPG-226
Working With Input Focus	SPG-229
Commanding a Pod	SPG-230
Event Broker Convenience Calls.....	SPG-233
Connecting to the Event Broker	SPG-233
Monitoring a Control Pad or a Mouse	SPG-234
Disconnecting From the Event Broker.....	SPG-235
Function Calls.....	SPG-235
Connecting To and Disconnecting From the Event Broker.....	SPG-235
Monitoring Events	SPG-235

13

The International Folio

What Is Internationalization?.....	SPG-238
The Locale Data Structure	SPG-239
NumericSpec Structure	SPG-240
DateSpec Arrays	SPG-243
Using the Locale Structure.....	SPG-243
Determining the Current Language and Country.....	SPG-243
Working With International Character Strings	SPG-244
Formatting Numbers or Currency	SPG-245
Formatting Dates	SPG-246
Converting Character Sets With the JString Folio.....	SPG-246
Function Calls and Macros.....	SPG-247
Obtaining a Locale Structure	SPG-247
Manipulating Characters and Dates	SPG-247
Converting Character Sets	SPG-248

14 The Compression Folio

Introduction.....	SPG-250
How the Compression Folio Works.....	SPG-250
Compressing Data	SPG-250
Decompressing Data	SPG-250
How to Use the Compression Folio.....	SPG-251
The Callback Function.....	SPG-252
Controlling Memory Allocations.....	SPG-253
Convenience Calls.....	SPG-253
Example.....	SPG-253
Function Calls.....	SPG-256
Convenience calls	SPG-256
Compression Calls	SPG-257
Decompression Calls	SPG-257
Buffer Calls	SPG-257

15 The Math Folio

Overview.....	SPG-259
Dealing With Overflow.....	SPG-260
List of Math Calls.....	SPG-260
Addition Calls.....	SPG-260
Subtraction Calls	SPG-260
Division Calls*.....	SPG-261
Same-Product Multiplication Calls	SPG-261
Matrix by Matrix Multiplication	SPG-262
Vector by Matrix Multiplication	SPG-263
Different-Product Multiplication Calls	SPG-263
Matrix Transposition Calls	SPG-264
Geometry Calls	SPG-264
Conversion Calls	SPG-265
Comparative Calls	SPG-266
One's and Two's Complement Calls	SPG-266
Reciprocal Calls	SPG-267
Squaring and Square Root Calls	SPG-267

16**Access: A Simple User Interface**

How Access Works.....	SPG-270
How to Use Access.....	SPG-270
Setting Up Message Communication With Access	SPG-271
Creating a Tag Argument List	SPG-271
Sending a Message to Access and Waiting for a Reply	SPG-274
Reading a Reply Message.....	SPG-275
Example.....	SPG-275
Access Message Tag Arguments.....	SPG-280

3DO System Programmer's Reference**Preface**

About this Book.....	xiii	
About the Audience.....		xiii
How this Book is Organized.....	xiii	
Related Documentation.....		xiv
Typographical Conventions.....		xiv

1**Kernel Folio Calls**

AbortIO	SPR-8
AddHead	SPR-9
AddTail	SPR-10
AllocMem	SPR-11
AllocMemBlocks	SPR-14
AllocMemFromMemList	SPR-16
AllocMemFromMemLists	SPR-18
AllocMemList.....	SPR-20
AllocSignal.....	SPR-21
AvailMem	SPR-24
CallFolio	SPR-26
CallFolioRet.....	SPR-28
CheckIO	SPR-30
CheckItem.....	SPR-31
ClearCurrentSignals	SPR-32
CloseItem	SPR-33
CloseNamedDevice	SPR-34

ControlMem	SPR-35
CountBits	SPR-37
CreateBufferedMsg	SPR-38
CreateIOReq	SPR-40
CreateItem	SPR-42
CreateMemDebug	SPR-44
CreateMsg	SPR-48
CreateMsgPort.....	SPR-50
CreateSemaphore	SPR-52
CreateSmallMsg	SPR-53
CreateThread	SPR-55
CreateUniqueMsgPort.....	SPR-57
CreateUniqueSemaphore	SPR-59
DeleteIOReq	SPR-60
DeleteItem	SPR-61
DeleteMemDebug	SPR-62
DeleteMsg	SPR-63
DeleteMsgPort.....	SPR-64
DeleteSemaphore	SPR-65
DeleteThread.....	SPR-66
DoIO	SPR-67
DumpMemDebug	SPR-69
DumpNode	SPR-70
DumpTagList	SPR-71
exit.....^.....	SPR-72
ffs	SPR
FindAndOpenDevice	SPR-74
FindAndOpenFolio.....	SPR-75
FindAndOpenItem	SPR-76
FindAndOpenNamedItem	SPR-78
FindDevice	SPR-79
FindFolio	SPR-80
FindItem	SPR-81
FindLSB	SPR-82
FindMSB	SPR-83
FindMsgPort	SPR-84
FindNamedItem	SPR-85
FindNamedNode	SPR-86

FindNodeFromHead	SPR-87
FindNodeFromTail.....	SPR-88
FindSemaphore.....	SPR-89
FindTagArg	SPR-90
FindTask	SPR-91
FindVersionedItem	SPR-92
FirstNode	SPR-93
free	SPR-94
FreeMem.....	SPR-95
FreeMemList.....	SPR-96
FreeMemToMemList.....	SPR-97
FreeMemToMemLists	SPR-98
FreeSignal.....	SPR-99
GetBankBits	SPR-100
GetCurrentSignals.....	SPR-101
GetFolioFunc	SPR-102
GetMemAllocAlignment	SPR-103
GetMemTrackSize	SPR-104
GetMemType	SPR-105
GetMsg	SPR-106
GetNodeCount	SPR-107
GetNodePosFromHead	SPR-108
GetNodePosFromTail	SPR-109
GetPageSize.....	SPR-110
GetSysErr	SPR-112
GetTagArg.....Z.....	SPR-113
GetTaskSignals	SPR-115
GetThisMsg	SPR-116
InitList	SPR-118
InsertNodeAfter	SPR-119
InsertNodeBefore.....	SPR-120
InsertNodeFromHead	SPR-121
InsertNodeFromTail	SPR-122
IsEmptyList.....	SPR-123
IsItemOpened	SPR-124
IsListEmpty.....	SPR-125
IsMemReadable	SPR-126
IsMemWritable	SPR-127

IsNode	SPR-128
IsNodeB	SPR-129
LastNode	SPR-130
LockItem.....	SPR-131
LockSemaphore	SPR-132
LookupItem	SPR-133
malloc	SPR-134
MkNodeID.....	SPR-135
NextNode	SPR-136
NextTagArg	SPR-137
OpenItem	SPR-139
OpenNamedDevice	SPR-141
PrevNode	SPR-142
PrintError.....	SPR-143
PrintfSysErr	SPR-145
ReadHardwareRandomNumber	SPR-146
RemHead	SPR-147
RemNode.....	SPR-148
RemTail	SPR-149
ReplyMsg	SPR-150
ReplySmallMsg.....	SPR-152
SampleSystemTime	SPR-154
SampleSystemTimeTV.....	SPR-155
SanityCheckMemDebug	SPR-156
ScanList.....^.....	SPR-157
ScanListB	SPR-158
ScavengeMem	SPR-159
SendIO	SPR-160
SendMsg	SPR-162
SendSignal	SPR-164
SendSmallMsg	SPR-165
SetItemOwner.....	SPR-166
SetItemPri.....	SPR-167
SetNodePri	SPR-168
UniversalInsertNode	SPR-169
UnlockItem	SPR-171
UnlockSemaphore	SPR-172
WaitIO	SPR-173

WaitPort	SPR-174
WaitSignal	SPR-175
Yield	SPR-177

2

File Folio Calls

ChangeDirectory	SPR-181
CloseDirectory	SPR-182
CloseDiskFile	SPR-183
CloseDiskStream	SPR-184
CreateAlias.....	SPR-185
CreateFile	SPR-186
DeleteFile	SPR-187
ExecuteAsSubroutine.....	SPR-188
ExecuteAsThread	SPR-190
GetDirectory	SPR-192
LoadCode	SPR-193
LoadProgram	SPR-195
LoadProgramPrio	SPR-196
OpenDirectoryItem	SPR-198
OpenDirectoryPath	SPR-199
OpenDiskFile	SPR-200
OpenDiskFileInDir.....	SPR-201
OpenDiskStream	SPR-202
ReadDirectory.....	SPR-203
ReadDiskStream	SPR-205
SeekDiskStream	SPR-206
UnloadCode	SPR-208

3

Math Folio Calls

AbsVec3_F16	SPR-216
AbsVec4_F16.....	SPR-217
Add32	SPR-218
Add64	SPR-219
AddF14	SPR-220
AddF16	SPR-221
AddFSO	SPR-222
AddF32	SPR-223

AddFGO	SPR-224
Atan2F16	SPR-225
CloseMathFolio	SPR-226
CompareS64	SPR-227
CompareSF32	SPR-228
CompareSFGO	SPR-229
CompareU64.....	SPR-230
CompareUF32	SPR-231
CompareUFBO	SPR-232
Convert32_F16	SPR-233
Convert32_F32	SPR-234
ConvertF14_F16	SPR-235
ConvertF16_32	SPR-236
ConvertF16_F14	SPR-237
ConvertF16_F30	SPR-238
ConvertF30_F16	SPR-239
ConvertF32_F16	SPR-240
ConvertS32_64	SPR-241
ConvertSF16_F32	SPR-242
ConvertU32_64.....	SPR-243
ConvertUF16_F32	SPR-244
CosF16	SPR-245
CosFSO	SPR-246
CosF32	SPR-247
Cross3_F16-*	SPR-248
DivRemS32.....	SPR-249
DivRemSF16	SPR-250
DivRemU32	SPR-251
DivRemUF16.....	SPR-252
DivS64	SPR-253
DK/SF16	SPR-254
DivU64	SPR-255
DivUF16	SPR-256
Dot3_F16	SPR-257
Dot4_F16	SPR-258
Mul32	SPR-259
Mul64	SPR-260
MulF14	SPR-261

MulManyF16	SPR-262
MulManyVec3Mat33_F16	SPR-263
MulManyVec3Mat33DivZ_F16	SPR-264
MulManyVec4Mat44_F16	SPR-265
MulMat33Mat33_F16	SPR-266
MulMat44Mat44_F16	SPR-267
MulObjectMat33_F16	SPR-268
MulObjectMat44_F16	SPR-269
MulObjectVec3Mat33_F16	SPR-270
MulObjectVec4Mat44_F16.....	SPR-272
MulS32_64	SPR-274
MulScalarF16	SPR-275
MulSF16	SPR-276
MulSF16_F32	SPR-277
MulSFSO	SPR-278
MulSF30_F60	SPR-279
MulU32_64	SPR-280
MulUF16	SPR-281
MulUF16_F32	SPR-282
MulUF30_F60	SPR-283
MulVec3Mat33_F16	SPR-284
MulVec3Mat33DivZ_F16.....	SPR-285
MulVec4Mat44_F16.....	SPR-286
Neg32	SPR-287
Neg64	SPR-288
NegF14T.....	SPR-289
NegF16	SPR-290
NegFSO	SPR-291
NegF32	SPR-292
NegFGO	SPR-293
Not32	SPR-294
Not64	SPR-295
NotF14	SPR-296
NotF16	SPR-297
NotFSO	SPR-298
NotF32	SPR-299
NotFGO	SPR-300
OpenMathFolio	SPR-301

RecipSF16	SPR-302
RecipUF16	SPR-303
SinF16	SPR-304
SinFSO	SPR-305
SinF32	SPR-306
Sqrt32	SPR-307
Sqrt64_32.....	SPR-308
SqrtF16	SPR-309
SqrtF32_F16	SPR-310
SqrtF60_F30	SPR-311
Square64	SPR-312
SquareSF16	SPR-313
SquareUF16	SPR-314
Sub32	SPR-315
Sub64	SPR-316
SubF14	SPR-317
SubF16	SPR-318
SubFSO	SPR-319
SubF32	SPR-320
SubFGO	SPR-321
Transpose33_F16	SPR-322
Transpose44_F16	SPR-323

4

Event Broker Calls

GetControlPad	SPR-326
GetMouse	SPR-328
InitEventUtility	SPR-330
KillEventUtility.....	SPR-331

5

Timer Calls

AddTimes	SPR-334
CompareTimes	SPR-335
CreateTimerLOReq	SPR-336
DeleteTimerLOReq	SPR-337
Sublimes	SPR-338
TimeLaterThan	SPR-339
TimeLaterThanOrEqual	SPR-340
WaitTime	SPR-341
WaitUntil.....	SPR-342

International Folio Calls

intlCloseLocale	SPR-344
intlCompareStrings	SPR-345
intlConvertString.....	SPR-346
intlFormatDate	SPR-348
IntlFormatNumber	SPR-350
intlGetCharAttrs	SPR-353
intlLookupLocale	SPR-355
intlOpenLocale.....	SPR-357
intlTransliterateString	SPR-358

7

JString Folio Calls

ConvertASCII2ShiftJIS	SPR-362
ConvertShiftJIS2ASCII	SPR-363
ConvertShiftJIS2Unicode	SPR-364
ConvertUnicode2ShiftJIS	SPR-365

8

Compression Folio Calls

Compress	SPR-368
CreateCompressor.....	SPR-369
CreateDecompressor	SPR-371
Decompress	SPR-373
DeleteCompressor	SPR-374
DeleteDecompressor	SPR-375
FeedCompressor.....	SPR-376
FeedDecompressor.....	SPR-377
GetCompressorWorkBufferSize	SPR-378
GetDecompressorWorkBufferSize	SPR-379

9

Portfolio Items

Attachment	SPR-383
Bitmap	SPR-385
Cue	SPR-387
Envelope	SPR-389
ErrorText.....	SPR-392

File	SPR-393
File alias	SPR-394
Folio	SPR-395
Instrument	SPR-396
IOReq	SPR-400
Knob	SPR-401
Locale	SPR-403
Message	SPR-404
Message port.....	SPR-406
Probe	SPR-407
Sample	SPR-408
ScreenGroup.....	SPR-413
Semaphore	SPR-415
Task	SPR-416
Template	SPR-419
Tuning	SPR-421
VOL.....	SPR-423

Glossary

Index

Colophon

CD-ROM Mastering Guide

Preface

Audience.....	xi
How This Document is Organized.....	xi
How to Work With This Document.....	xii
Typographical Conventions.....	xii

1

Hardware and Software Requirements

Requirements for a 3DO Development System.....	CDM-2
3DO Development System Hardware Requirements	CDM-2
3DO Development System Software Requirements	CDM-3
Requirements for a CD-ROM Recording Station.....	CDM-3
Recording Station Layout	CDM-3
Hardware Requirements	CDM-4
Software and Media	CDM-4
Requirements for 3DO Testing Stations	CDM-4

2

Preparing for the Mastering Process

Application Size Limitations.....	CDM-6
Factors Influencing Available Space.....	CDM-6
Determining Available Space for Your Application	CDM-6
Programming Notes.....	CDM-7
Using Relative Pathnames t.....	CDM-7
Hints on Error Checking and Other Advice	CDM-7
Checking for Low Memory Pointers.....	CDM-8
Why Avoiding Low Memory Access is Important	CDM-8
How to Detect and Remove Low Memory Pointers	CDM-8
Testing Your Title in the Minimum Memory Configuration.....	CDM-10
What is minmem?	CDM-10
When and Why to Test Your Application with minmem	CDM-10
How to Use minmem	CDM-11

Creating and Testing the Image File

Background and Overview.....	CDM-14
Layout Tool Background Information	CDM-14
Optimized Layout Process Overview	CDM-16
Required Files and Folders.....	CDM-16
Preparing Your Title's Files for the Layout Process	CDM-16
Checking the AppStartup File.....	CDM-17
Modifying the BannerScreen File	CDM-17
Preparing Simple cdrom.image Files.....	CDM-19
Editing the cdrom.tcl File for the Simple Image	CDM-19
Creating the Simple Image	CDM-20
Testing the Simple Image	CDM-21
Preparing an Optimized CD-ROM Image.....	CDM-22
Creating the Base Image for Information Collection	CDM-22
Preparing the Debugger for Information Collection	CDM-22
Running Your Title to Collect Access Information	CDM-23
Running the Layout Tool to Create an Optimized Image	CDM-24
Testing the CD-ROM Image File on a Macintosh.....	CDM-26
Layout Tool Troubleshooting.....	CDM-27

4

Mastering and Testing the CD-ROM

Preparation and Setup for the Mastering Process.....	CDM-30
Maximizing Recording Speed	CDM-30
Setting Up the QuickTOPIX Software	CDM-30
Creating a CD-ROM Disc for Testing	CDM-33
Testing the CD-ROM Disc Using the Debugger.....	CDM-34
Testing the CD-ROM in Minimum Memory Configuration	CDM-34
Debugging Hints	CDM-35
Testing the CD-ROM Disc on a Testing Station	CDM-36
Creating a CD-ROM Disc for Encryption.....	CDM-37

5

Delivery for Encryption

About the Encryption Requirements.....	CDM-40
Overview of Encryption Process.....	CDM-41

CD-ROM Mastering Checklist

Reading List.....	CDM-43
Preparing for the Mastering Process.....	CDM-43
Creating and Testing a Simple CD-ROM Image File.....	CDM-44
Creating an Optimized Image.....	CDM-45
Testing the Optimized CD-ROM Image on the Macintosh	CDM-46
Mastering the CD-ROM Disc.....	CDM-46
Testing the CD-ROM.....	CDM-47
Preparing Materials for Encryption.....	CDM-48

B

Title Authorization and Royalty Collection

Getting a Title Authorized.....	CDM-52
Licensee Designs the Packaging	CDM-52
Licensee Submits Preliminary Materials.....	CDM-52
Non-Standard Titles	CDM-53
Product Submission to 3DO	CDM-53
Right to Manufacture Authorization and Royalty Payment.....	CDM-54

C

CD-ROM Mastering Etiquette

CD-ROM Basics.....	CDM-56
Constant Linear Velocity	CDM-56
3DO Drive Specifications	CDM-56
Data Rate Issues:».....	CDM-56
Error Detection and Correction	CDM-57
Programming Do's and Don'ts.....	CDM-58
Making Assumptions about Data Delivery Rates	CDM-58
Handling Data Delays.....	CDM-58
Alternating File Reads	CDM-58
Testing Performance	CDM-59
Other Things You Should Do	CDM-59
Head Seeks and CD-ROM Mechanism Lifetime.....	CDM-60

File Formats

Appendix Overview	CDM-61
3DO File Format.....	CDM-62
File Format Description	CDM-62
Using Multiple Chunks	CDM-64
3DO File Format BNF	CDM-66
Chunk Definitions	CDM-67
MDFL FORM.....	CDM-74
MThd Chunk - Header.....	CDM-74
MIDI Track Chunks (MTrk)	CDM-74
DSP Instrument Templates.....	CDM-75
DCOD Chunk (Code PlusTagArgs)	CDM-75
DKNB Chunk (Knob)	CDM-76
DRLC Chunk (Relocation)	CDM-77
DRSC Chunk (Resource Allocation Request)	CDM-77
DNMS Chunk (Packed Resource Names)	CDM-78
AIFF (Audio Interchange File Format).....	CDM-79
AIFF-C	CDM-79
PPM (Portable Pixel Map).....	CDM-79

300 Programmer's Reference

AbortAsyncLoadFile	LDO-2
AsyncLoadFile.....	LDO-3
BlitFontChar.....	LDO-5
CEL_DATAPTR	LDO-7
CEL_NEXTPTR	LDO-8
CEL_PLUTPTR	LDO-9
CEL_PREOWORD	LDO-10
CEL_PRE1WORD	LDO-11
CenterCRectInCRect	LDO-12
CenterCRectInDisplay	LDO-13
CenterCRectOverlPoint.....	LDO-14
CenterFPointInDisplay	LDO-15
CenterIPointInCRect	LDO-16
CenterIPointInDisplay	LDO-17
CenterIPointInSRect	LDO-18
CenterRectAACellInCRect	LDO-19
CenterRectAACellInDisplay	LDO-20
CenterRectAACellInSRect	LDO-21
CenterRectAACelOverFPoint	LDO-22
CenterRectAACelOverlPoint.....	LDO-23
CenterRectCellInCRect.....	LDO-24
CenterRectCellInDisplay	LDO-25
CenterRectCellInSRect.....	LDO-26
CenterRectCellListInCRect.....	LDO-27
CenterRectCellListInDisplay.....	LDO-28
CenterRectCellListInSRect	LDO-29
CenterRectCellListOverFPoint.....	LDO-30
CenterRectCellListOverlPoint.....	LDO-31
CenterRectCelOverFPoint.....	LDO-32
CenterRectCelOverlPoint	LDO-33
CenterSRectInDisplay	LDO-34
CenterSRectInSRect	LDO-35
CenterSRectOverlPoint	LDO-36
ChainCelsAtHead	LDO-37
ChainCelsAtTail.....	LDO-38
CheckAsyncLoadFile	LDO-40
CleanupMSEvents	LDO-41

ClearBitmap	LDO-42
CloneCel	LDO-43
CloneTextCel.....	LDO-45
CreateBackdropCel	LDO-46
CreateBasicDisplay	LDO-48
CreateCel	LDO-50
CreateLRFormCel.....	LDO-52
CreateSubrectCel	LDO-54
CreateTextCel.....	LDO-56
CRectBounds	LDO-57
CRectFromCel	LDO-58
CRectFromIVal.....	LDO-59
CRectFromSRect	LDO-60
CRectIntersection	LDO-61
CrossFadeCels	LDO-62
CrossFadeCelsS	LDO-64
DeleteBasicDisplay	LDO-66
DeleteCel.....	LDO-67
DeleteCelList.....	LDO-68
DeleteTextCel	LDO-69
DetachTextCelCCB	LDO-70
DisableMSEvent.....	LDO-71
DispatchMSEvents.....	LDO-72
DrawAnimCel	LDO-73
DrawImage	LDO-75
DrawTextChar.....	LDO-76
DrawTextString	LDO-77
EnableMSEvent	LDO-78
EraseTextInCel	LDO-79
FadeFromBlack	LDO-80
FadeToBlack	LDO-81
FinishAsyncLoadFile	LDO-82
FPointFromFVal	LDO-83
FPointFromIPoint.....	LDO-84
FPointFromIVal	LDO-85
GetAnimCel	LDO-86
GetCelBitsPerPixel	LDO-87
GetCelBytesPerRow	LDO-88

GetCelDataBufferSize.....	LDO-89
GetChunk	LDO-90
GetFontCharInfo	LDO-91
GetFontCharWidest.....	LDO-92
GetFontCharWidth	LDO-93
GetFontStringWidth	LDO-94
GetHSecTime	LDO-95
GetMSecTime	LDO-96
GetTextCelColor.....	LDO-97
GetTextCelColors	LDO-98
GetTextCelCoords	LDO-99
GetTextCelFormatBuffer.....	LDO-100
GetTextCelFormatFlags	LDO-101
GetTextCelLeadingAdjust.....	LDO-102
GetTextCelMargins	LDO-103
GetTextCelPenNumber.....	LDO-104
GetTextCelSize.....	LDO-105
GetTextCelSpacingAdjust.....	LDO-106
GetTextCelTabStops	LDO-107
GetTextExtent	LDO-108
GetTime	LDO-109
GetTimerIOReq	LDO-110
GetTSecTime	LDO-111
GetUSecTime	LDO-112
GetVBLTime.....	LDO-113
HZ_TO_USEC	LDO-114
ICornerFromSRect.....	LDO-115
InsetCRect	LDO-116
InsetSRect.....	LDO-117
IPointFromFPoint.....	LDO-118
IPointFromFVal	LDO-119
IPointFromIVal.....	LDO-120
IPointInCRect.....	LDO-121
IPointInSRect.....	LDO-122
ISJ.ASTCEL	LDO-123
ISizeFromCRect	LDO-124
LastCellInList.....	LDO-125
LinkCel.....	LDO-126

LoadAnim	LDO-127
LoadCel.....	LDO-129
MapCelListToCRect	LDO-141
MapCelListToFPoint.....	LDO-142
MapCelListToIPoint	LDO-143
MapCelListToSRect	LDO-144
MapCelToCQuad	LDO-145
MapCelToCRect	LDO-146
MapCelToFPoint	LDO-147
MapCelToIPoint	LDO-148
MapCelToSRect	LDO-149
MSEC_TO_USEC	LDO-150
OffsetAACelByFDelta	LDO-151
OffsetAACelByIDelta	LDO-152
OffsetCel	LDO-153
OffsetCelByFDelta	LDO-154
OffsetCelByIDelta	LDO-155
OffsetCelListByFDelta	LDO-156
OffsetCelListByIDelta	LDO-157
OffsetCRect.....	LDO-158
OffsetSRect.....	LDO-159
ParseCel	LDO-160
ParseFont	LDO-161
ReadFile	LDO-162
RenderCelFillRect	LDO-163
RenderCelHLine	LDO-164
RenderCelOutlineRect	LDO-165
RenderCelPixel	LDO-166
RenderCelVLine	LDO-167
ReturnCelPixel	LDO-168
SaveFile	LDO-169
SECJTOJSE	LDO-170
SetTextCelColor	LDO-171
SetTextCelColors	LDO-172
SetTextCelCoords	LDO-173
SetTextCelFormatBuffer.....	LDO-174
SetTextCelFormatFlags	LDO-176
SetTextCelLeadingAdjust	LDO-177

SetTextCelMargins.....	LDO-178
SetTextCelPenNumber	LDO-179
SetTextCelSize	LDO-180
SetTextCelSpacingAdjust.....	LDO-181
SetTextCelTabStops.....	LDO-182
SetupMSEvents	LDO-183
Sleep.....	LDO-185
SleepHSec	LDO-186
SleepMSec.....	LDO-187
SleepTSec	LDO-188
SleepUSec	LDO-189
SRectBounds	LDO-190
SRectFromCel.....	LDO-191
SRectFromCRect	LDO-192
SRectFromIVal.....	LDO-193
SRectIntersection	LDO-194
taCreateTextCel	LDO-195
taModifyTextCel	LDO-196
TimerCancel.....	LDO-197
TimerChangeUserdata	LDO-198
TimerMsgAfterDelay	LDO-199
TimerMsgAfterDelayVBL.....	LDO-200
TimerMsgAtTime	LDO-201
TimerMsgAtTimeVBL.....	LDO-202
TimerMsgHeartbeat	LDO-203
TimerMsgHeartbeatVBL.....	LDO-205
TimerReset	LDO-206
TimerRestart.....	LDO-207
TimerServicesClose	LDO-208
TimerServicesOpen.....	LDO-209
TimerServicesShutdown	LDO-210
TimerServicesStartup	LDO-211
TimerServicesVerify.....	LDO-212
TimerSignalAfterDelay	LDO-213
TimerSignalAfterDelayVBL.....	LDO-214
TimerSignalAtTime	LDO-215
TimerSignalAtTimeVBL	LDO-216
TimerSignalHeartbeat	LDO-217

TimerSignalHeartbeatVBL	LDO-218
TimerSuspend	LDO-219
UnloadAnim	LDO-220
UnloadCel	LDO-221
UnloadFile	LDO-222
UnloadFont	LDO-223
UnloadImage	LDO-224
UpdateTextInCel	LDO-225
VGetTextExtent.....	LDO-227
vUpdateTextInCel.....	LDO-228
WaitAsyncLoadFile	LDO-229
WriteMacFile.....	LDO-230
XCORNERFROMSRECT	LDO-231
XSIZEFROMCRECT	LDO-232
YCORNERFROMSRECT	LDO-233
YSIZEFROMCRECT	LDO-234



3DO Toolkit 1.5 **—Contents**

3DO Toolkit 1.5—Contents

Tools For Art Production

Preface

About this document.....	GIN-vii
Audience.....	GIN-vii
How this document is organized.....	GIN-vii
Typographical conventions.....	GIN-viii

1

Preparing Art for 3DO Titles

Introduction.....	GIN-1
Chapter overview	GIN-1
3DO graphics overview.....	GIN-2
How the 3DO hardware processes images and cels.....	GIN-3
What are images and cels?	GIN-3
<i>What is a 3DO cel?</i>	GIN-3
Where does the color come from?	GIN-4
Where does the cel's shape come from?.....	GIN-5
Where do the special effects come from?	GIN-6
Working with images, cels, and animations.....	GIN-8
Working with images	GIN-8
Working with cels	GIN-9
Working with 3DO animations	GIN-10

Tools for 3DO Art Production

Introduction.....	GIN-11
Chapter overview	GIN-11
Finding the right tool.....	GIN-12
Finding out what a tool can do.....	GIN-13
Custom code modules and custom plug-ins.....	GIN-14

3 Tips, Tricks, and Troubleshooting

Introduction.....	GIN-15
Chapter overview	GIN-15
Cels.....	GIN-16
Video and animations.....	GIN-16
From PICT to Cinepak	GIN-16
Resolution for video	GIN-16
Converting from DVI or Video for Windows to Cinepak	GIN-17
Video data rate and compression	GIN-17
MPEG	GIN-17
Saving animation to PC format.....	GIN-17
Video production tips.....	GIN-17
Transparency and resizing.....	GIN-19
Transparency and the color black	GIN-19
Resizing art from 640 x 480 to 320 x 240	GIN-19
Working with fonts.....	GIN-19
The 3DO PPM Translator tool.....	GIN-20
MakeCel out of memory	GIN-20
Documentation for PPM tools	GIN-20
Photoshop and DeBabelizer.....	GIN-20
Photoshop version	GIN-20
Photoshop problems	GIN-21
Photoshop and coded cels	GIN-21
Corrupt CelWriter files from DeBabelizer	GIN-21
3DO Animator.....	GIN-21
Loading animations based on variable-size cels	GIN-21
Loading QuickTime movies into 3DO Animator	GIN-21
Dithering in 3DO Animator.....	GIN-22
Creating a simple animation	GIN-22
Not enough memory for 3DO Animator.....	GIN-22
3DO Animator doesn't launch	GIN-22

Using the 3DO Frame Grabber

Introduction.....	GIN-23
Using the 3DO Frame Grabber.....	GIN-23
3DO Frame Grabber and 3DO Animator.....	GIN-24

30 Animator User's Guide

Preface

About this document.....	ANI-xv
Audience.....	ANI-xv
How this document is organized.....	ANI-xvi
Typographical conventions.....	ANI-xvi
System prerequisites.....	ANI-xvii
3DO Animator Preferences files	ANI-xvii
Memory restrictions	ANI-xvii

1

Creating 3DO Images and Cels

Introduction.....	ANI-1
Chapter overview	ANI-1
For more information	ANI-1
Resizing source art.....	ANI-2
How to resize source art while opening a file	ANI-2
Creating a simple uncoded cel.....	ANI-4
Images in 3DO Animator	ANI-4
Converting 32-bit source art to a 16-bit uncoded cel	ANI-4
Creating a simple coded cel.....	ANI-6
Introduction	ANI-6
How to create a coded cel	ANI-6
Customizing a palette.....	ANI-8
Working with the 3DO palette	ANI-8
Working with a custom palette	ANI-9
Saving and loading palettes	ANI-10
Creating an irregularly shaped cel.....	ANI-12
Creating a packed irregularly shaped cel	ANI-12
Creating an unpacked, irregularly shaped cel	ANI-14
Working with special effects.....	ANI-18
How PIXC and P-Mode works	ANI-18

How to make an entire cel translucent.....	ANI-18
How to enhance luminance	ANI-20
How to define a translucent area in an opaque cel.....	AMI-22
Setting nondefault cel characteristics.....	ANI-25
Loading previously prepared cel characteristics	ANI-25
Creating custom settings for a cel	ANI-25

2

Animation Basics

Introduction.....	ANI-27
Chapter overview	ANI-27
For more information	ANI-27
Creating animations with 3DO Animator.....	ANI-28
Flip-book type animations	ANI-28
Disk space considerations	ANI-28
Overview of creating animations with 3DO Animator	ANI-29
Special features for processing animations with 3DO Animator	ANI-29
Creating a simple animation.....	ANI-30
Creating frames	ANI-30
Painting one frame at a time	AMI-31
Playing an animation	ANI-33
Saving an animation	ANI-33
Using special features to create animations quickly.....	ANI-35
Painting onto flipping frames	ANI-35
Using Anim Move,.....	AMI-36

3

Working with Custom Brushes

Introduction.....	ANI-39
Chapter overview	ANI-39
For more information	ANI-39
Required files for working through examples	AMI-39
Painting with a custom brush.....	AMI-40
For more information	AMMO
Creating a custom brush	ANI-40
Working with an existing animated brush.....	ANI-42
A first look at an animated brush	ANI-42
Creating an animation with an animated brush	ANI-43
Creating an animated brush using Anim Selection.....	ANI-44

Loading an animation	ANI-44
Creating an animated selection	ANI-45
Using the selection as an animated brush	ANI-46
Saving animated brushes and selections.....	ANI-48
Saving an animated brush	ANI-48
Saving an animated selection	ANI-48

4

Advanced 3DO Animation

Introduction.....	ANI-49
Chapter overview	ANI-49
For more information	ANI-49
Transition effects.....	ANI-50
About this section	ANI-50
Creating moving vertical blinds	ANI-50
Animated distortion.....	ANI-52
Creating a distorted triangle shape	ANI-52
Using the light table.....	ANI-55
About this section	ANI-55
How to create a new animation with the light table	ANI-55
Adding sound to an animation.....	ANI-58
How to add sounds to an animation	ANI-58

5

Menu Commands and Dialogs

Introduction..... *	ANI-61
Chapter overview	ANI-61
For more information	ANI-62
The File menu.....	ANI-63
File menu commands discussed in this section	ANI-63
New command	ANI-63
Open command	ANI-64
Open As command	ANI-66
Save As command	ANI-68
Build an Animation command	ANI-70
Build an Animation As command	ANI-70
The Edit menu.....	ANI-71
3DO Preferences command	ANI-71
The Selection menu.....	ANI-74

Selection menu commands discussed in this section	ANI-74
The Apply Selection to All Frames command	ANI-74
The How to Apply a Move to All Frames command	ANI-75
The How to Apply a Resize to All Frames command	ANI-75
The Brush menu.....	ANI-77
The Goodies menu.....	ANI-77
The Color menu.....	ANI-78
Color menu commands discussed in this section	ANI-78
Replace Colors command	ANI-79
Lock Colors command	ANI-79
Show NTSC Hot Colors command	ANI-79
Remap NTSC Hot Colors command	ANI-79
The 3DO Options menu	ANI-81
Access 3DO Palette command	ANI-81
Show/Hide 3DO Palette command	ANI-81
Show/Hide 3DO Board Controls command	ANI-82
Show/Hide 3DO PMode Control Panel command	ANI-83
Set CCB command	ANI-84
Translucency Options command	ANI-84
Quick Build 3DO Anim command	ANI-85
Show NTSC Hot Colors & Remap NTSC Hot Colors commands	ANI-85
The Anim menu.....	ANI-86
Anim menu commands discussed in this section	ANI-86
Set # of Frames command	ANI-86
Show/Hide ControlPanel command	ANI-87
Go to Frame command	ANI-90
Key Frames command	ANI-90
Append command	ANI-91
Save Range command	ANI-91
Delete command	ANI-91
Insert command	ANI-92
Clear command	ANI-92
Light Table command	ANI-93
Grid Distortion command	ANI-95
Anim Selection command	ANI-95
Process command	ANI-96

The Set 3DO Object Type Dialog

Introduction.....	... ANI-97
Chapter overviewANI-97
Bringing up the dialog.....	... ANI-98
Color sliders and color type.....	... ANI-98
3DOpalette.....	... ANI-99
Function buttons.....	, ANI-100
Function buttons with No Mode selectedANI-100
Function buttons for coded celsAN 1-102
CCB Settings.....	..ANI-103
Load SettingsAMI-103
Cel Type.....	..ANI-103
P-Mode.....	..ANI-103
SFANI-103

Custom Plug-Ins and Custom Code Modules

Preface

About this document.....	. PLU-xi
How to use this document.....	..PLU-xi
Audience.....	. PLU-xi
How this document is organized.....	. PLU-xi
Typographical conventions.....	PLU-xii
Installing the Custom Plug-Ijjs. and CCMs.....	PLU-xii

1

Working with 3DO Images

Introduction.....	PLU-1
Chapter overviewPLU-1
How to use this chapter.....	..PLU-1
Creating a basic image.....	PLU-2
Resizing source art.....	..PLU-2
Exporting source art with the 3DO Image WriterPLU-3
Creating images with custom VDLs.....	PLU-5
Creating an image with a custom VDLPLU-5
Standard and custom VDLsPLU-6
Previewing 3DO images.....	PLU-7
About the 3DO Artist CardPLU-7

How to preview an image with the 3DO Image Writer	PLU-7
How to preview an image with SDOImagePreview	PLU-7
Loading 3DO Images.....	PLU-8

2

Working With 3DO Gels

Introduction.....	PLU-9
Chapter overview	PLU-9
How to use this chapter	PLU-9
For more information	PLU-9
Overview of cel creation process.....	PLU-10
Creating uncoded cels.....	PLU-11
How to convert 16-bit source art to a 16-bit uncoded cel	PLU-11
How to convert 24-bit source art to an 8-bit uncoded cel.....	PLU-12
Creating coded cels with the 3DO CelWriter.....	PLU-14
How to create a coded cel	PLU-14
Creating cels with the 3DO Coded-8 CelWriter.....	PLU-15
Creating a cel with an auto-generated palette	PLU-15
Using custom palettes with the Coded-8 CelWriter.....	PLU-16
3DO Coded-8 CelWriter—Background information	PLU-17
Creating cels with the Anti-Alias CelWriter.....	PLU-19
How to create an anti-aliased cel	PLU-19
How to display an anti-aliased cel	PLU-19
Displaying an image and a cel on the 3DO Station.....	PLU-20
Using the 3DO Debugger to display source art	PLU-20
Loading 3DO cels into Photoshop.....	PLU-20

3

Using Transparency and Special Effects

Introduction.....	PLU-21
Chapter overview	PLU-21
Using channels for special effects—Overview.....	PLU-22
Which channels can I use?	PLU-22
Creating an irregularly shaped cel.....	PLU-23
How to create an irregularly shaped cel (no true black in cel)	PLU-23
How to use a special color for an irregularly shaped cel (Coded-8 CelWriter) ...	PLU-25
Creating special effects using P-Mode.....	PLU-27
How to use an extra channel to brighten parts of a cel	PLU-27
Creating and editing a P-Mode preset.....	PLU-29

How to create a P-Mode preset	PLU-29
Which cels allow P-Mode	PLU-30
For more information	PLU-30

4

Plug-In Dialogs

Introduction.....	PLU-31
Chapter overview	PLU-31
The 3DO Image Writer dialog	PLU-32



VDL Options dialog.....	PLU-33
The GLUT bypass option	PLU-33
The 3DO CelWriter dialog	PLU-35
The Cel Control Settings dialog.....	PLU-37

5

3DO Custom Code Module for DeBabelizer

Introduction.....	PLU-41
Chapter overview	PLU-41
Currently available CCMs.....	PLU-42
Creating a 3DO animation.....	PLU-43
Introduction	PLU-43
How to create an ANIM file with the CCM	PLU-43
Creating a series of 3DO cels.....	PLU-47
Creating 8-bit coded cels from DeBabelizer.....	PLU-49

3DO FontWriter User's Guide

Preface

About this document.....	FOW-ix
Audience.....	FOW-ix
How this document is organized.....	FOW-ix
Related documentation.....	FOW-x
Typographical conventions.....	FOW-x

1

*

3DO FontWriter Quick Start

Introduction.....	FOW-1
Chapter overview	FOW-1
For more information	FOW-1
Creating a new anti-aliased font.....	FOW-1
How to create a new font	FOW-2
3DO FontWriter windows.....	FOW-3
Character Edit window overview	FOW-4

3DO FontWriter Reference

Introduction	POW-5
The 3DO FontWriter window.....	FOW-6
3DO FontWriter window options	FOW-7
The Character Edit window.....	FOW-8
Choosing a character	FOW-8
Using Character Edit window drawing tools	FOW-8
Changing character width and baseline	FOW-9
Choosing between a character and its shadow or outline	FOW-10
The Preview window.....	FOW-11
Experimenting with different colors	FOW-11
The Character Map window.....	FOW-12
The Sample Text window.....	FOW-12
Window update and text changes	FOW-13
The File menu.....	FOW-13
The Edit menu.....	FOW-14
The View menu.....	FOW-15
The Transform menu.....	FOW-17
The Special menu.....	FOW-17

Video Processing Guide

Preface

About this document.....	VID-xi
Audience.....	VID-xi
How this document is organized.....	VID-xi
Typographical conventions.....	VID-xii

1

Video Preparation Overview

Introduction.....	VID-1
-------------------	-------

2

Selecting Hardware

Introduction.....	VID-3
Digital hardware requirements.....	VID-4
Hardware options.....	VID-4

3

Capturing and Processing Materials

Introduction.....	VID-7
Preparing source material.....	VID-8
Securing a master copy	VID-8
Shooting your own video	VID-8
Preparing film source material	VID-8
Will it compress well?	VID-9
Capturing video	VID-10
Capture process	VID-11
Processing digitized materials.....	VID-12
3-2 Pulldown	VID-12
Deinterlacing	VID-12
Cropping and resizing.....	VID-13
Cropping	VID-13
Resizing	VID-13

4

Compressing and Playing Your Movie

Getting QuickTime and Cinepak tools.....	VID-16
Quality issues	VID-16
Preparing for Cinepak compression.....	VID-16
Compressing animations	VID-16
Eliminating noise	VID-16
Setting the data rate	VID-17
Compression options.....	VID-17

5

Video Tools

Introduction.....	VID-19
MovieEdit.....	VID-20
Saving movies with MovieEdit	VID-20
MovieEdit user interface	VID-21
Limitations.....	VID-22
The 3-2 Pulldown tool.....	VID-22
Understanding Telecine	VID-22
Choosing a movie	VID-23
Field dominance.....	VID-23
Removing composite frames	VID-24

Saving a processed movie	VID-26
Troubleshooting	VID-26
MovieCompress.....	VI D-27
Movie Compression Settings dialog options	VID-27
Example for using the MovieCompress Tool	VID-30
Compressing with ConvertToMovie.....	VID-32

A

EZ Squeeze

Introduction.....	VI D-35
About this appendix	VID-35
Overview.....	VID-36
Caveats	VID-36
Audience	VID-36
Updates	VID-36
Installing EZ Squeeze.....	VID-37
Quick install	VID-37
Full installation.....	VID-37
Sample data.....	VID-38
Making a movie.....	VID-38
Equipment	VID-38
Video Source	VID-38
Audio	VID-38
Compressing with EZ Squeeze	VID-38
Adjusting parameters.....	VID-39
Weaver and scripts	VID-39
Playing a Movie with EZQPlayer.....	VID-39
Summary of EZQPlayer Controls	VID-39
Command line arguments.....	VID-40

3DO Tools for Programming I

3DO Debugger Programmer's Guide

Preface

About this document.....	DBG-xi
About the 3DO Debugger	DBG-xi
Audience.....	DBG-xi
How this document is organized.....	DBG-xii
Typographical conventions.....	DBG-xii

1

3DO Debugger Quick Start

Introduction.....	DBG-1
Chapter overview	DBG-1
3DO Debugger overview.....	DBG-2
Starting a debugging session.....	DBG-3
Preparing for launch	DBG-3
Starting the Debugger.....	DBG-5
Using display windows to examine the source code.....	DBG-6
Working with a Source window	DBG-6
Setting breakpoints.....	DBG-7
Examining a structure and changing variable values	DBG-8
Moving through the program efficiently.....	DBG-11

2

Menus, Windows, and Dialog Boxes

Introduction.....	DBG-13
Chapter overview	DBG-13
3DO Debugger menus.....	DBG-14
The File menu	DBG-14
The Edit menu	DBG-17
The View menu	DBG-18
The Execution menu	DBG-20
The Target menu	DBG-22
Display windows.....	DBG-24
The Source window	DBG-24
Using conditional breakpoints from the Source window.....	DBG-27
The Data window	DBG-28

The Disassembly window	DBG-32
Interface shortcuts.....	DBG-34
The Variables window.....	DBG-35
Using multiple Variables windows	DBG-35
Working with Variables windows	DBG-35
Variables window Options menu	DBG-37
The Breakpoints window.....	DBG-39
Working with the Breakpoints window	DBG-39
The Stack window.....	DBG-40
Working with the Stack window	DBG-40
Stack View options	DBG-41
The Terminal window.....	DBG-42
Detail display windows.....	DBG-42
The Registers window	DBG-42
The Symbol window	DBG-44
Preferences and Configuration dialog box.....	DBG-46
Preferences dialog pop-ups	DBG-46
Preferences dialog Configuration checkboxes	DBG-48

A

Examine Field Expression Syntax

B

Debugger Script Commands

Introduction.....	DBG-51
modbin	DBG-52
stripaif	DBG-53
sendstr	DBG-54

C

Terminal Window Commands

Overview.....	DBG-55
iostress	DBG-57
items.....	DBG-58
killprintf.....	DBG-60
killtask	DBG-61
lmadm	DBG-62
lmdf	DBG-65
setalias	DBG-66
setbg.....	DBG-67

setcd	DBG-68
setfg	DBG-69
setmaxmem	DBG-70
setminmem	086-71
setpri	DB6-72
setverbose	DB6-73
showavailmem	DBG-74
showed	DB6-75
showerror.....	DBG-76
showkernelbase	DBG-77
showmemlist	DBG-78
showmemmap	DBG-79
showshellvars	DBG-80
showtask	DBG-81
sleep	DBG-82
sysload	DBG-83

3DO DataStreamer Programmer's Guide

Preface

About this document.....	DSG-vii
Audience.....	DSG-vii
How this document is organized.....	DSG-vii
Typographical conventions.....	DSG-viii

1

3DO DataStreamer Overview

Introduction.....	DSG-1
Chapter overview	DSG-1
For more information	DSG-1
Why use the DataStreamer?.....	DSG-2
Examples for using the DataStreamer.....	DSG-2
Future development.....	DSG-2
Preparing a streaming application.....	DSG-3

Preparing and Playing a Simple Stream

Introduction.....	DSG-5
Chapter overview	DSG-5
For more information	DSG-5
What are stream files?.....	DSG-6
Stream file preparation basics	DSG-6
Stream file terminology	DSG-7
Analyzing data and making trade-offs.....	DSG-8
System resource limitations	DSG-8
Trade-offs	DSG-9
Generating and converting data.....	DSG-11
Generating data	DSG-11
Converting data to 3DO format.....	DSG-12
Compressing data.....	DSG-13
Creating chunk files	DSG-13
Creating a stream file with the Weaver tool.....	DSG-15
Creating a Weaver script.....	DSG-15
Using the Weaver tool	DSG-17
Examining the output stream	DSG-17
Playing back a simple stream.....	DSG-18
Playback example applications	DSG-18
Special considerations for different types of data	DSG-19

3

Inside Streaming: Basics *""

Introduction.....	DSG-21
Chapter overview	DSG-21
DataStreamer threads and data flow.....	DSG-22
DataStreamer threads	DSG-22
Data flow overview	DSG-23
Playing a Stream—Overview.....	DSG-25
Creating a message port and a message.....	DSG-25
Example for setting up a message port.....:	DSG-25
The message-sending mechanism	DSG-26
Allocating data buffers.....	DSG-28
Example for creating a buffer list.....	DSG-28
Deciding on buffer size	DSG-29
Launching the required threads.....	DSG-30

Initializing and connecting data acquisition and streamer threads	DSG-30
Initializing and connecting subscriber threads	DSG-32
Audio Playback.....	DSG-34
Weaver script commands for audio	DSG-34
Initialization for audio	DSG-34
Displaying persistent data with the Join subscriber.....	DSG-36
Displaying a background image	DSG-36

4

Inside Streaming: Control

Introduction.....	DSG-37
Chapter overview	DSG-37
Time and the DataStreamer.....	DSG-38
The stream clock	DSG-38
Changing time within a stream.....	DSG-39
Starting and stopping a stream	DSG-39
Jumping to a new location	DSG-41
Performing fast-forward and fast-rewind operations.....	DSG-42

5

Debugging and Optimization

Introduction.....	DSG-43
Chapter overview	DSG-43
Minimizing filler.....	DSG-44
Introduction	DSG-44
Guidelines for minimizing filler	DSG-44
Using appropriate streamblock size and number of buffers.....	DSG-45
Streamblock size restrictions	DSG-45
Dealing with data rate spikes	DSG-45
Using more buffers to prevent starving buffers	DSG-46
Selecting priorities.....	DSG-47
Recommended order of process priorities	DSG-47
How bad priorities can stop your stream	DSG-48
Using tracing to understand data flow.....	DSG-49
How tracing is implemented	DSG-49
Using tracing in your application	DSG-50

Creating New Subscribers

Introduction.....	DSG-55
Chapter overview	DSG-55
Conceptual background.....	DSG-56
Buffering	DSG-56
Logical channels	DSG-56
Control calls	DSG-56
Creating and initializing subscribers.....	DSG-57
Initializing a subscriber.....	DSG-57
Creating the subscriber thread	DSG-58
Implementing subscriber functionality.....	DSG-60
The SubscriberMsg data structure.....	DSG-60
Subscriber message opcodes	DSG-62
Queued data and positioning operations.....	DSG-63
Video data.....	DSG-63
Audio data	DSG-63

300 DataStreamer Programmer's Reference

Preface

About this document.....	DSR-iii
For more information	DSR-iii
Audience.....	DSR-iii
How this document is organized.....	DSR-iii
Typographical conventions.....	DSR-iv

1

DataStreamer Functions

AllocPoolItem	DSR-5
AllocPoolMem	DSR-6
AsynchReadBlockFile.....	DSR-7
CloseBlockFile.....	DSR-9
CloseDataAcq	DSR-10
CloseTextSubscriber.....	DSR-11
CreateBlockFileIOReq	DSR-12
CreateItemPool	DSR-13
CreateMemPool	DSR-14
CreateMsgItem	DSR-15

DeleteItemPool	DSR-16
DeleteMemPool	DSR-17
DisposeDataAcq	DSR-18
DisposeDataStream	DSR-19
DisposeSCelSubscriber	DSR-20
DisposeSemaphore.....	DSR-21
DisposeTextSubscriber.....	DSR-22
DisposeThread	DSR-23
DSClockSync	DSR-24
DSConnect.....	DSR-26
DSControl	DSR-28
DSGetChannel.....	DSR-30
DSGetClock.....	DSR-32
DSGoMarker.....	DSR-33
DSIsMarker.....	DSR-35
DSPreRollStream	DSR-36
DSSetChannel	DSR-38
DSSetClock	DSR-40
DSStartStream	DSR-42
DSStopStream	DSR-44
DSSubscribe	DSR-46
DSWaitEndOfStream.....	DSR-48
FillPoolWithMsgltems.....	DSR-50
ForEachFreePoolMember.....	DSR-51
GetBlockFileBlockSize	DSR-52
GetFramesRemaining	DSR-53
GetMsgPortSignal.....	DSR-54
GetSCelList	DSR-55
GetSCelListAndType	DSR-56
GetSegmentCount.....	DSR-57
GetStreamText.....	DSR-58
GetStreamTextCel	DSR-59
GetTextFontName	DSR-60
InitDataAcq	DSR-61
InitDataStreaming	DSR-62
InitTextSubscriber	DSR-63
MakeName	DSR-64
NewDataAcq	DSR-65

NewDataStream	DSR-66
NewMsgPort.....	DSR-68
NewSCelSubscriber	DSR-69
NewTextSubscriber.....	DSR-70
NewThread	DSR-71
OpenBlockFile	DSR-73
PollForMsg	DSR-74
PrintfDSError.....	DSR-75
ReadDoneBlockFile	DSR-76
RemoveMsgItem	DSR-77
RemoveMsgPort.....	DSR-78
ReturnPoolItem	DSR-79
ReturnPoolMem	DSR-80
SetSCelChannelFlags	DSR-81
SetSCelMemoryFcns.....	DSR-82
SetTextMemoryFcns	DSR-83
WaitForMsg	DSR-84
WaitReadDoneBlockFile	DSR-86

2

Data Structures and Data Types

AcqContext	DSR-89
BlockFile	DSR-92
DataAcqMsg	DSR-93
DataAcqOpcode	DSR-95
DSDDataBuf	DSR-96
DSErrorCode.....	DSR-97
DSHeaderChunk	DSR-99
DS_MSG_HEADER	DSR-101
DSRequestMsg	DSR-102
DSRequestOpcode.....	DSR-105
DStreamCB	DSR-107
DSubscriber	DSR-109
GenericMsg	DSR-110
ItemDesc	DSR-111
ItemPool	DSR-112
MemPool	DSR-113
MemPoolEntry.....	DSR-114
StreamChunk	DSR-115

StreamOpcode	DSR-116
SubscriberMsg	DSR-117
SUBS_CHUNK_COMMON	DSR-119

3

Data Streaming Tools

Introduction.....	DSR-121
SanmMaker	DSR-123
AnimToSanm	DSR-124
Chunkify	DSR-125
ChunkMatcher	DSR-126
CtlMaker	DSR-127
DumpAIFF	DSR-129
DumpMF	DSR-130
Dumpstream	DSR-131
DumpStreamCCBs	DSR-132
GetDSErrorText.....	DSR-133
MovieToStream	DSR-134
MovieToStream_Shuttle	DSR-135
SAudioTool	DSR-136
SFTToStream	DSR-137
SquashSnd	DSR-139
Weaver	DSR-140

4

Weaver Script Commands -»

Overview.....	DSR-143
audioclockchan	DSR-145
dataacqdeltapri	DSR-146
enableaudiomask	DSR-147
file	DSR-148
markertime	DSR-149
mediablocksize	DSR-150
numsubsmessages	DSR-151
preloadinstrument	DSR-152
streamblocksize	DSR-153
streambuffers	DSR-154
streamdeltapri.....	DSR-155
streamstarttime	DSR-156

subscriber.....	.DSR-157
writetochunk.....	.DSR-158
writetochunkDSR-159
writemarkertable....	.DSR-160
writestopchunkDSR-161
writestreamheader	.DSR-162

300 Tools For Programming II

ARM User's Guide- 3DO Edition

Preface

About this manual.....	AUG-vii
ARM tool release components.....	AUG-vii
Programming and modelling tools	AUG-viii
Retargetable libraries	AUG-viii
Documentation	AUG-ix
How this book is organized.....	AUG-ix
How this book differs from the ARM documentation.....	AUG-ix

1

The ARM C Compiler

Introduction.....	AUG-1
Chapter overview	AUG-1
About armcc.....	AUG-2
Invoking armcc.....	AUG-2
Keyword options.....	AUG-2
Flag options.....	AUG-4
Controlling the linker	AUG-4
Preprocessor flags	AUG-4
Controlling code generation	AUG-5
Controlling warning messages	AUG-7
Controlling additional compiler features	AUG-9

2

The ARM Assembler (armasm)

Introduction.....	AUG-13
Chapter overview	AUG-13
For more information	AUG-13
Command line options.....	AUG-13

3

The ARM Linker (armlink)

Introduction.....	AUG-17
Chapter overview	AUG-17
About the ARM linker.....	AUG-17

Linker command line.....	AUG-19
General options.....	AUG-19
Output format options.....	AUG-20
Special options.....	AUG-23
Input file list.....	AUG-25

4

The ARM Librarian (armlib)

Introduction.....	AUG-27
Chapter overview	AUG-27
For more information	AUG-27
Command line options.....	AUG-27

5

The ARM Object Format Decoder (decaof)

Introduction.....	AUG-29
Chapter overview	AUG-29
For more information	AUG-29
Command line options.....	AUG-29

6

An Introduction to the Run-Time Libraries

7

ARM Assembly Language

Overview.....	AUG-33
Chapter overview/*	AUG-33
General	AUG-34
Input lines.....	AUG-34
AREAS	AUG-34
ORG and ABS	AUG-35
Symbols	AUG-36
Labels	AUG-36
Local labels.....	AUG-36
Comments	AUG-37
Constants	AUG-37
The END directive	AUG-38

The ARM instruction set.....	AUG-38
Conditional execution and the 'S' bit.....	AUG-38
Register Names and '!'.....	AUG-39
Branch instructions - B and BL	AUG-40
Data processing instructions - MOV and MVN	AUG-40
Data processing instructions - arithmetic and logical	AUG-42
Data processing instructions - comparison operations	AUG-43
PSR transfer - MSR and MRS	AUG-44
Single data transfer - LDR and STR	AUG-45
Block data transfer - LDM and STM	AUG-46
Multiplies - MUL, MLA	AUG-47
Single data swap - SWP	AUG-48
Software interrupt/supervisor call - SWI	AUG-48
Pseudo-instructions - ADR and NOP	AUG-48
Generic coprocessor instructions.....	AUG-49
Coprocesor data transfers - LDC and STC	AUG-49
Coprocesor data operations - CDP	AUG-50
Coprocesor register transfers - MCR and MRC	AUG-50
Floating point instructions.....	AUG-50
Floating point data transfer - LDF and STF	AUG-51
Floating point register transfer - FIX and FIT	AUG-51
Floating point register transfer - status and control	AUG-52
Floating point multiple data transfer - LFM and SFM	AUG-52
Floating point comparisons - CMF and CNF	AUG-53
Floating point binary operations	AUG-54
Floating point unary operations	AUG-55
Directives.....	AUG-56
Storage reservation and initialisation - DCB, DCW and DCD	AUG-56
Floating point store initialisation - DCFS and DCFD	AUG-56
Describing the layout of store - ^A and #	AUG-57
Organisational directives - END, ORG, LTORG and KEEP	AUG-57
Links to other object files - IMPORT and EXPORT	AUG-58
Links to other source files - GET/INCLUDE	AUG-58
Diagnostic generation - ASSERT and !	AUG-59
Dynamic listing options - OPT	AUG-59
Titles - TTL and SUBT	AUG-60
Miscellaneous directives-ALIGN, NOFP, RUST and ENTRY	AUG-60

Symbolic capabilities.....	AUG-61
Setting constants.....	AUG-61
Local and global variables - GBL, LCL and SET.....	AUG-61
Variable substitution - \$	AUG-62
Built-in variables	AUG-62
Expressions and operators.....	AUG-63
Unary operators	AUG-63
Binary operators	AUG-64
Conditional assembly - [, I and].....	AUG-67
Repetitive assembly - WHILE and WEND.....	AUG-67
Macros.....	AUG-67
Usage	AUG-67
Defining a macro	AUG-68
Setting default parameter values	AUG-68
Macro invocation	AUG-68

ARM Reference Manual- 3DO Edition

Preface

About this document.....	ARR-vii
Audience.....	ARR-vii
How this document is organized.....	ARR-vii
Information not included in this document.....	ARR-viii

1

The ARM C Compiler "»

Introduction.....	ARR-1
Overview	ARR-1
About the ARM C compiler.....	ARR-2
Recommended texts.....	ARR-3
Using the ARM C compiler.....	ARR-4
File naming conventions	ARR-4
Filename validity	ARR-5
Object files.....	ARR-5
Included files	ARR-5
Implementation details.....	ARR-7
Character sets and identifiers	ARR-7
Data Elements.....	ARR-8
Arithmetic limits (limits.h and float.h)	ARR-9

Structured data types	ARR-12
Pointers	ARR-12
Pointer subtraction	ARR-13
Arithmetic operations	ARR-13
Expression evaluation	ARR-14
Implementation limits	ARR-14
Standard implementation definition.....	ARR-15
Translation	ARR-15
Environment	ARR-16
Identifiers	ARR-17
Characters	ARR-17
Integers	ARR-19
Floatingpoint.....	ARR-19
Arrays and pointers	ARR-19
Registers	ARR-19
Structures, unions, enumerations and bitfields	ARR-20
Qualifiers	ARR-20
Declarators	ARR-20
Statements	ARR-20
Preprocessing directives	ARR-21
Library functions	ARR-21
Portability.....	ARR-24
Fundamental data types	ARR-25
Byte ordering	ARR-26
Store alignment	ARR-26
Pointers and pointer arithmetic	ARR-27
Function-argument evaluation	ARR-28
System-specific code	ARR-28
ANSI C vs K&R C.....	ARR-28
Lexical elements	ARR-29
Arithmetic	ARR-30
Expressions	ARR-31
Declarations.....	ARR-31
Statements	ARR-31
Preprocessor	ARR-32
PCC compatibility mode.....	ARR-32
Language and preprocessor compatibility	ARR-32
Standard headers and libraries	ARR-34

Machine-specific features.....	ARR-36
Pragma directives	ARR-36
Specifying pragmas from the command line	ARR-37
Pragmas controlling the preprocessor.....	ARR-37
Pragmas controlling printf/scanf argument checking	ARR-38
Pragmas controlling optimisation	ARR-38
Pragmas Controlling Code Generation	ARR-38
Special function declaration keywords	ARR-40
Special variable declaration keywords	ARR-41
Floating point support.....	ARR-42

2

ARM C Library Porting Guide

Introduction.....	ARR-43
Chapter overview	ARR-43
Example code	ARR-43
Source organisation.....	ARR-44
Building a target-specific library.....	ARR-46
Retargetting the library.....	ARR-46
Variant selection	ARR-46
Basic choices.....	ARR-48
Address space model	ARR-48
I/O model.....	ARR-50
Details of target-dependent code.....	ARR-50
ANSI library functions.....	ARR-50
I/O support *	ARR-50
Floating-point support	ARR-52
Kernel	ARR-53
Miscellaneous	ARR-54

3

The ARM Linker

Introduction.....	ARR-55
Chapter overview	ARR-55
About the ARM linker.....	ARR-56
Library module inclusion.....	ARR-57
Area placement and sorting rules.....	ARR-58
Linker pre-defined symbols.....	ARR-59

The handling of relocation directives.....	ARR-60
The subject field	ARR-60
The relocation value	ARR-61
PC-relative relocation	ARR-61
Forcing use of an inter-link-unit entry point.....	ARR-61
Additive relocation	ARR-61
Based area relocation	ARR-62
The relocation of instruction sequences	ARR-62
ARM object format.....	ARR-63
Plain binary format.....	ARR-64
ARM image format.....	ARR-64
Extended intellec hex Format (IMF).....	ARR-65
ARM shared library format.....	ARR-66
Stub properties	ARR-66
The shared library addressing architecture	ARR-67
Including data areas in a shared library	ARR-69
Entry veneers	ARR-69
Versions, compatibility and foreverness	ARR-73
Describing a shared library to the linker	ARR-74
Linker pre-defined symbols	ARR-77
Overlays.....	ARR-77
Static overlays	ARR-77
Dynamic overlays	ARR-78
Assignment of AREAs to overlay segments	ARR-79
Describing an overlay structure to the linker	ARR-80
The overlay manager.....	ARR-81
The structure of a PCIT section	ARR-82
The load_seg_and_go code	ARR-83
The load_segment code	ARR-84
Intercepting returns to overwritten segments	ARR-86
Return handler code	ARR-87
The load_seg_and_ret code	ARR-88
The check for_invalidated_returns Code	ARR-89

ARM Cookbook

Preface

About this document.....	ACB-ix
Audience.....	ACB-ix
How this document is organized.....	ACB-ix
Examples used in this document.....	ACB-x

1

ARM Instruction Set and Processor Features

Introduction.....	ACB-1
Chapter overview	ACB-1
Examples	ACB-1
Overview of key features.....	ACB-2
Making the most of conditional execution.....	ACB-2
About this recipe	ACB-2
The ARM'S ALU status flags	ACB-3
Execution conditions	ACB-3
Setting the ALU flags in the PSR	ACB-4
Illustration—Euclid's GCD algorithm	ACB-4
Running the C example.....	ACB-5
Running the assembler example	ACB-5
Related topics	ACB-6
Using the Barrel Shifter.....	ACB-6
About this recipe.....	ACB-6
Addressing an entry in a table of words	ACB-6
The ARM'S Barrel Shifter.....	ACB-7
LDR/STR	ACB-8
Data processing operations	ACB-8
Program status register transfer instructions	ACB-8
Related topics	ACB-9
Flexibility of load and store multiple.....	ACB-9
About this recipe.....	ACB-9
Multiple vs single transfers	ACB-9
The register list	ACB-9
Increment/Decrement, Before/After.....	ACB-10
Base register writeback	ACB-10
Stack notation.....	ACB-10
Related topics	ACB-11

Loading constants into registers.....	ACB-11
About this recipe	ACB-11
Why is loading constants an issue?	ACB-11
MOV/MVN	ACB-12
Assembling the example	ACB-12
LDR Rd, =numeric constant	ACB-13
Literal pools	ACB-13
Assembling the example	ACB-14
LDR Rd, =PC relative expression	ACB-14
Assembling the example	ACB-15
ADRandADRL	ACB-15
Assembling the example	ACB-16
Related topics	ACB-17

2

Exploring ARM Assembly Language

Introduction.....	ACB-19
Chapter overview	ACB-19
Integer to string conversion.....	ACB-20
About this recipe.....	ACB-20
The algorithm	ACB-20
The implementation	ACB-21
Creating a runnable example	ACB-22
Running the example	ACB-23
Stacks in assembly language	ACB-23
Related topics*	ACB-25
Multiply by a constant.....	ACB-26
About this recipe	ACB-26
Introduction	ACB-26
The problem of finding the optimum sequence	ACB-27
Experimenting with armcc assembly output	ACB-27
Discussion of speed improvement.....	ACB-28
Related topics	ACB-29
Division by a constant.....	ACB-30
About this recipe.....	ACB-30
Introduction	ACB-30
Special case for divide-by-2 ⁿ	ACB-30
Explanation of divide-by-constant ARM code	ACB-30
How to generate divide-by-constant sequences	ACB-34

Related topics	ACB-35
Choosing a division implementation.....	ACB-36
Divide implementations in the C-library	ACB-36
Guaranteed-performance divide routines for real-time applications	ACB-37
Summary.....	ACB-39
Related topics	ACB-39
Digital signal processing on the ARM.....	ACB-40
About this recipe	ACB-40
Introduction	ACB-40
Examples of some DSP code on the ARM	ACB-41
ARM6 multiplier performance issues	ACB-44
Related topics	ACB-47
Using 16-bit data on the ARM.....	ACB-48
About this recipe	ACB-48
Introduction	ACB-48
How "16-bit" is my data ?	ACB-48
Little endian loading recipes	ACB-49
Little endian storing recipes.....	ACB-51
Big endian loading recipes	ACB-52
Big endian storing recipes	ACB-53
Detecting overflow into the top 16 bits	ACB-54
Using ARM registers as 16-bit registers	ACB-55
Related topics	ACB-56
ARM600 page table generation.....	ACB-57
About this recipe	ACB-57
Repetitive assembly	ACB-57
Macro usage and conditional assembly	ACB-58
Assembling the page tables in plain binary format	ACB-58
Related topics	ACB-59
Pseudo random number generation.....	ACB-60
About this recipe	ACB-60
The details	ACB-60
Using this example	ACB-60
Related topics	ACB-61
Loading a word from an unknown alignment.....	ACB-62
About this recipe.....	ACB-62
The details	ACB-62
Related topics	ACB-62

Byte order reversal.....	ACB-63
About this recipe	ACB-63
The details	ACB-63
Related topics	ACB-63
ARM assembly programming performance issues.....	ACB-64
About this recipe	ACB-64
Performance issues	ACB-64
Related topics	ACB-72
Examples	ACB-72

3

Interfacing Assembly Language and C

Introduction.....	ACB-73
Register usage under the ARM procedure call standard.....	ACB-73
About this recipe	ACB-73
Introduction to the ARCS	ACB-74
Register names and usage under the ARCS	ACB-74
64 Bit integer addition	ACB-76
More detailed ARCS register usage information	ACB-78
Related topics	ACB-79
Passing and returning structs.....	ACB-80
About this recipe	ACB-80
The default way to pass and return a struct	ACB-80
The optimisation of integer-like structures	ACB-81
Returning non integer-like structs in registers	ACB-83
Related topics *	ACB-85
In-LineSWIs.....	ACB-85
About this recipe	ACB-85
Introduction	ACB-85
Using a SWI which returns no result	ACB-86
Using a SWI which returns one result	ACB-87
Using a SWI which returns 2-4 results	ACB-88
The SWI number is not known until run time	ACB-89
Related topics	ACB-90

4

Programming in C

Overview.....	ACB-91
Examples	ACB-91
A simple C program.....	ACB-91
About this recipe	ACB-91
Prerequisites.....	ACB-92
Making a simple runnable program	ACB-92
Running the program	ACB-93
Separate compiling	ACB-93
Separate linking	ACB-93
Related topics	ACB-94
Writing efficient C for the ARM.....	ACB-94
About this recipe.....	ACB-94
Function design considerations	ACB-94
Register allocation and how to help it.....	ACB-99
Static and extern variables - minimising access costs	ACB-101
The switch() statement.....	ACB-102
Related topics.....	ACB-104
C Programming for deeply embedded applications.....	ACB-104
About this recipe.....	ACB-104
Introduction	ACB-105
Using the standalone runtime system	ACB-106
A simple program	ACB-106
Error handling	ACB-107
longjmp and setjmp	ACB-108
Floating point support.....	ACB-109
Running out of heap	ACB-111
Stack overflow checking	ACB-111
Extending the standalone runtime system	ACB-112
The size of the standalone runtime library	ACB-113
Related topics	ACB-113
ARM shared libraries.....	ACB-114
About this recipe.....	ACB-114
About ARM shared libraries	ACB-114
How ARM shared libraries work	ACB-114
Locating a library which matches the stub	ACB-115
Instructing the linker to make a shared library.....	ACB-116

Making a toy string library	ACB-118
Suggested further exercises	ACB-120
Related topics	ACB-121

ARM Technical Specifications- 3DO Edition

Preface

Overview.....	ATS-vii
Audience.....	ATS-vii
Chapters not included in this version.....	ATS-vii

1

ARM Image Format

Introduction.....	ATS-1
Chapter overview	ATS-1
Properties of ARM image format.....	ATS-1
Executable AIF	ATS-2
The layout of AIF.....	ATS-3
AIF Header Layout	ATS-5
Zero-initialisation code.....	ATS-7
Self-move and self-relocation code	ATS-7

2

ARM Object Format

Introduction.....	ATS-11
Chapter overview	ATS-11
About AOF.....	ATS-12
Terminology	ATS-12
Byte sex or endian-ness	ATS-12
Alignment	ATS-13
The overall structure of an AOF file.....	ATS-13
Chunk file format	ATS-13
ARM object format.....	ATS-15
Format of the AOF header chunk	ATS-16
Format of area headers	ATS-17
Attributes + Alignment.....	ATS-18
Area attributes summary	ATS-21

Format of the areas chunk.....	ATS-21
Relocation directives	ATS-22
Format of the symbol table chunk (OBJ_SYMT).....	ATS-24
Symbol attributes	ATS-25
Symbol attribute summary	ATS-27
The string table chunk (OBJ_STRT).....	ATS-27
The identification chunk (OBJJDFN).....	ATS-27

3

ARM Object Library Format

Introduction.....	ATS-29
Chapter overview	ATS-29
About the library format.....	ATS-29
Terminology	ATS-30
Byte sex or endian-ness	ATS-30
Alignment	ATS-30
Library File Format.....	ATS-31
LIB_DIRY.....	ATS-31
Time stamps	ATS-32
LIB_TIME	ATS-33
LIBJ/SRN	ATS-33
LIB_DATA	ATS-33
Object code libraries.....	ATS-33
OFL_SYMT	ATS-34
OFLJTIME	ATS-34

4

ARM Procedure Call Standard

Introduction.....	ATS-35
Chapter overview	ATS-35
Design criteria	ATS-36
The ARM procedure call standard.....	ATS-36
Register names.....	ATS-37
General registers	ATS-38
Floating point registers	ATS-39
The stack	ATS-40
The stack backtrace data structure	ATS-41
Function invocations and backtrace structures	ATS-41
Control arrival	ATS-42
Data representation and argument passing	ATS-43

Control return	ATS-44
ARCS variants.....	ATS-45
32-bit PC vs 26-bit PC	ATS-45
Implicit vs explicit stack-limit checking	ATS-46
Floating-point arguments in floating-point registers	ATS-46
Reentrant vs non-reentrant code	ATS-46
APCS-2 compatibility	ATS-47
C language calling conventions.....	ATS-47
Argument representation	ATS-47
Argument list marshalling	ATS-48
Non-simple value return	ATS-48
Function entry—Introduction	ATS-48
Function entry—Establishing the static base	ATS-49
Function entry—creating the stack backtrace structure.....	ATS-50
Function entry—saving and restoring floating point registers	ATS-51
Function entry—Checking for stack limit violations	ATS-52
Stack limit checking—small, fixed frames	ATS-53
Stack limit checking—large, fixed frames	ATS-53
Stack limit checking—Vari-sized frames	ATS-53
Function exit	ATS-54
Some examples.....	ATS-55
The APCS in non-user ARM modes.....	ATS-57
Aborts and pre-ARM6-based ARMs	ATS-57

3DO Tools For Sound Design

Preface

About this document.....	TSD-iii
Audience.....	TSD-iii
<i>How this document is organized.....</i>	<i>7SD-/i></i>
Typographical conventions.....	TSD-iv

1

Music and Sound Effects for 3DO Titles

Introduction.....	TSD-1
Chapter overview	TSD-1
Resource allocation and planning.....	TSD-2
RAM-resident vs. ROM-resident sound	TSD-2
Spooled vs. streamed sound	TSD-2
Sound file characteristics	TSD-3
Audio production.....	TSD-4
Sampling sound effects	TSD-4
Working with sampled music	TSD-5
Creating algorithmic sound effects	TSD-6
Incorporating sound effects into your title.....	TSD-6
Playing MIDI scores	TSD-6
Streaming audio data with the 3DO DataStreamer	TSD-6
Aesthetic considerations.....	TSD-7
Tweak, tweak, tweak!.....	TSD-7
Project management.....	TSD-8

2

Installing and Launching ARIA

Introduction.....	TSD-11
Chapter overview	TSD-11
Starting ARIA.....	TSD-12
Setting up the 3DO Debugger	TSD-12
Setting up ARIA	TSD-13

Setup for working with MIDI.....	TSD-14
Files and folder setup	TSD-14
Required files for real-time MIDI	TSD-14
Setting up for MIDI	TSD-15
Installing ARIA using drag and drop.....	TSD-16

3

Creating Patches with ARIA

Introduction.....	TSD-17
Chapter overview	TSD-17
ARIA interface overview.....	TSD-18
Block palette	TSD-18
Patch document window	TSD-18
Log window	TSD-18
Creating your first patch.....	TSD-19
Creating a simple wave form	TSD-19
Creating frequency and amplitude knobs	TSD-20
Connecting and disconnecting blocks	TSD-21
Editing knobs	TSD-22
Creating an output block	TSD-22
Compiling and playing a patch.....	TSD-23
How to compile a patch	TSD-23
How to play a patch	TSD-23
Interacting with a patch.....	TSD-24
Using the 3DO control pad for input	TSD-24
Playing patches from the keyboard	TSD-25
Creating a patch with a sample.....	TSD-26
Creating blocks for the sampler patch	TSD-26
Compiling and playing the sampler patch	TSD-27
Tips and tricks	TSD-28
Modifying sound with envelopes.....	TSD-29
Preparing a patch with an envelope	TSD-29
Tips and tricks for working with envelopes	TSD-32
Automating patch creation.....	TSD-34
How to automate patch creation	TSD-34
Working with PatchDemo scripts.....	TSD-35
How to use PatchDemo scripts with ARIA	TSD-35

Creating MIDI Projects with ARIA

Introduction.....	TSD-37
Chapter overview	TSD-37
Preparing your environment.....	TSD-38
MIDI files and PIMap files.....	TSD-39
Working with PIMaps	TSD-39
Playing a MIDI file on the 3DO Station.....	TSD-41
Working with real-time MIDI.....	TSD-43
Tips and tricks.....	TSD-44
MIDI project interface tips	TSD-44
MIDI tricks	TSD-44

5

Using SoundHack

Introduction.....	TSD-45
Chapter overview	TSD-45
For more information	TSD-45
Playing and converting a sound file.....	TSD-46
Loading a sound file	TSD-46
Playing the current sound file	TSD-46
Converting the current sound file to 3DO AIFC format	TSD-46
Spacializing a sound—Binaural filter.....	TSD-47
Processing a file with the binaural filter	TSD-47
Multiplying spectra—convolution.....	TSD-48
Multiplying the spectra of two sound files	TSD-48
Memory considerations.....	TSD-48

6

SoundHack Menus and Dialogs

Introduction.....	TSD-49
Chapter overview	TSD-49
The File menu.....	TSD-50
Open command	TSD-50
Open Any command	TSD-51
Close command	TSD-51
Save A Copy command	TSD-52
lisTen to AIFF file command	TSD-52
Import SND resource command	TSD-52

Export SND resource command	TSD-52
Quit command	TSD-52
The Edit menu.....	TSD-52
The Hack menu.....	TSD-52
Header Change command	TSD-53
Loops & Markers command	TSD-53
Binaural Filter	TSD-54
Convolution command	TSD-56
Gain Change command	TSD-58
Phase Vocoder command	TSD-58
Spectral Dynamics command	TSD-61
Varispeed command	TSD-63
The Control menu.....	TSD-65
Show Output command	TSD-65
Show Spectrum command	TSD-65
Pause Process command	TSD-65
Continue Process command	TSD-65
Stop Process command	TSD-65

7

Using PatchDemo

Introduction.....	TSD-67
Chapter overview	TSD-67
PatchDemo features and background.....	TSD-68
Background	TSD-68
Working with PatchDemo... *	TSD-69
Creating a patch file	TSD-69
Running a patch	TSD-71
Adjusting instrument knobs.....	TSD-71

8

PatchDemo Reference

Introduction.....	TSD-73
AttachSample	TSD-74
Connect	TSD-75
DelayLine	TSD-76
LoadInstrument	TSD-77
LoadSample.....	TSD-78
Tweak	TSD-79

Tips, Tricks, and Troubleshooting

Available audio tools.....	TSD-81
SoundHack	TSD-81
SquashSound	TSD-81
ARIA	TSD-81
PatchDemo	TSD-82
AIFF Files.....	TSD-82
AIFF files in Examples folder	TSD-82
AIFF and compression	TSD-82
Preparing Audio.....	TSD-82
Mastering audio	TSD-82
Editing Synthesized Audio	TSD-82

A

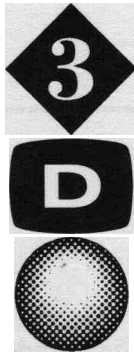
Instruments Available in ARIA

Introduction.....	TSD-85
-------------------	--------

B

AIFF Samples

Overview.....	TSD-87
File names.....	TSD-88
Auditioning samples from the Sound Designer II Edit window.....	TSD-88
Folder contents.....	TSD-89
Samples in GMPercussion22k and GMPercussion44k	TSD-89
Samples in the PitchedL folder	TSD-91
Samples in the PitchedLR folder	TSD-91
Samples in the Unpitched folder	TSD-91



3DO Toolkit 1.5 —Glossary

Glossary

This glossary contains terms used in the 3DO Portfolio and 3DO Toolkit documentation.

Symbols and Numbers

0x00 byte

The NULL byte that is part of any C string but is not counted in the string's length.

15-bit color

A color defined with 15 bits: 5 for red, 5 for green, and 5 for blue. 15-bit color is used within the frame buffer and as the output of the cel engine.

1555 mode

A display mode that uses bit 15 as a cornerweight bit, bits 14-10 as red value bits, bits 9-5 as green value bits, and bits 4-0 as blue value bits.

15541 mode

A display mode that uses bit 15 as a cornerweight bit, bits 14-10 as red value bits, bits 9-5 as green value bits, bits 4-1 as blue value bits, and bit 0 as a second cornerweight bit.

24-bit color

A color defined with 24 bits: 8 for red, 8 for green, and 8 for blue. 24-bit color is used only within the display generator.

3-2 Pulldown

Tool for removing composite frames (intermediate frames inserted during Telecine conversion from film to video) from a QuickTime movie. Removing composite frames increases the movie's perceived resolution and reduces bandwidth required to display it on the 3DO Interactive Multiplayer. See also Telecine process.

3DO Animator

A full-color painting program based on the Studio/32 program from Electronic Arts. It allows you to create animations and edit QuickTime movies. You can save files as 3DO cels, 3DO images, and animations in 3DO file format.

3DO CD Browser

A tool that lets you access the 3DO Content Library. The Content Library contains a large collection of still pictures, movies, and sounds you can use while developing 3DO titles. The 3DO CD Browser lets you search the library using keywords, which are matched against abstracts describing individual sounds, pictures, or movie fragments. You can save the results of your search for later use.

3DO cel

A bitmapped image stored in DRAM or VRAM that passes through the cel engine and is projected into the frame buffer. See also coded cel, uncoded cel.

3DO Content Library

The 3DO Content Library contains a large collection of copyright free still pictures, textures, movies, sound effects, music, and so on that you can use while developing 3DO titles. Each sound, picture, or other piece of art is described by an abstract, and all abstracts are collected in a database that you can search with the 3DO CD Browser.

3DO DataStreamer

Code set for playing and synchronizing a data stream that consists of different media types, each supported by its own subscriber library or code set.

3DO Debugger

A cross-debugger that runs on the Macintosh and lets you examine code running on the 3DO Station from a Macintosh Terminal window. You can look at and step through source code, assembly, and data. You can also dereference pointers, evaluate and change variables, and look at the different stacks.

300 driver

Part of the 3DO Debugger. The 3DO driver performs low-level control of the 3DO Station through the NuBus communication board. The Macintosh automatically launches the 3DO driver each time you boot the system using the 3DO driver system extension (INIT).

3DO file format

A chunk-type file format that provides a standard for use on the 3DO Station. A number of chunks have already been defined and are described in the "3DO File Format" appendix to the CD-ROM *Mastering Guide*. See also chunk.

3DO FontWriter

A tool that builds fonts you can use on the 3DO system from fonts on a Macintosh system disk. You can edit the font after it's been built and display it in the Font-Writer Character Edit window.

3DO image

The simplest kind of 3DO graphic. A 3DO image is a 320 x 240 (NTSC) or 384 x 288 (PAL) array of 16-bit pixel values, laid out in the format of the frame buffer. The hardware can draw images—which are useful for large, static backgrounds—very quickly.

300 Portfolio

The operating system of the 3DO Station. You must use the libraries, or folios, that are part of the operating system to communicate with the 3DO hardware.

300 PostPro

A post-production tool for 3DO title developers. PostPro can open and save various visual (but not sound) file types and do file conversion. It displays files on a color Macintosh monitor and on a television display monitor simultaneously, and its data-editing capabilities let you view modifications immediately on both monitors.

300 Station

The development platform for 3DO applications.

300 Toolkit

A collection of software tools for developing applications for the 3DO Station.

\$samples

A standard MPW shell file alias that points to the 3DO directory where sample files are stored.

abstract

A term used by the 3DO CD Browser. An abstract is a line of text describing a piece of 3DO Content Library data. This line points to the location of the associated Content Library data. For example, a sound could be described by the abstract "track and field indoor 50-yard dash false start." To search the Content Library, specify keywords that are then matched against the abstracts.

ADC

Analog-to-digital converter, a chip that converts an analog audio signal into a digital sample stream.

AIF file

ARM Image Format file, an image file produced by the ARM linker; currently the only kind of file the Portfolio operating system can load and execute.

AIFF

Audio Interchange File Format, the file format used for 3DO digital wave samples.

alias

A string (such as \$boot, \$samples) that must be used to avoid hard-coding full pathnames in code. Usually set in a shell startup script.

alternate multiply value (AMV)

A value carried within a cel pixel, used in the pixel processor to multiply the source color value for that pixel. If you mix foreground and background colors to achieve a translucent effect, you can make one of the two colors more dominant by giving it an alternate multiply value—for example, to simulate very dark or rather light glass.

amplitude

Generally, the loudness of a sound. Also, a reserved special name for any knob.

Apple MIDI Manager

Macintosh software, including a driver, that allows MIDI applications such as sequencers and ARIA to send real-time MIDI messages to each other. See also MIDI.

AMV bits

Bits derived from a coded cel pixel that contain an alternate multiply value for the pixel processor.

analog-to-digital converter

See ADC.

anchor

A C union that marks the beginning or end of a linked list.

Animator

See 3DO Animator.

anti-aliasing

A technique that reduces the stair-step effects of lines and curves in a graphic.

AOFile

ARM Object Format file, an object file that's produced by the ARM C compiler and the ARM assembler.

ARIA

A Macintosh application for sound designers for: 1) constructing new 3DO instruments graphically in a patch window; 2) setting up a MIDI project window for playing back and debugging MIDI files from disk on the 3DO; 3) setting up a MIDI project window and using it to get the 3DO system to respond to real-time MIDI events from a sequencer or keyboard, using sampled or synthetic.

An on-screen oscilloscope ("scope") lets you view a sound on the 3DO screen, and an audio monitor screen permits monitoring DSP source usage.

attachment item

An item that defines the attachment between a sample and an instrument, both of which remain as independent entities.

audio ticks

Timed events created 240 times per second using 3DO internal timers. Audio ticks are used to time internal note events such as the shape of a note's envelope.

audlock cycling

One type of the cyclic running mode in the digital signal processor (DSP). In this mode, the cycle time is set by the DAC chip's crystal to synchronize DSP cycles with DAC chip cycles.

B

background color

A 24-bit color assigned to a background pixel. The background color is determined by the contents of the background color registers within the GLUT set.

background image

An image (digitized or live analog) that appears behind the foreground cel contained in the frame buffer. If the foreground cels have any background pixels defined as transparent, the background image appears through them. See also 3DO image.

background pixel

A frame buffer pixel set to 000. A background pixel can be assigned a background color or can take a color value from a background image supplied by Slipstream. See also background color, transparent pixel.

background pixel detector

A component of the display generator that looks for and flags background pixels, notifying the GLUT set and Slipstream whenever one appears.

background register

A special GLUT register containing a red, green, or blue component used to color a background pixel.

banks

Unit of measure for VRAM (Video Random Access Memory). Each bank of VRAM is one megabyte large. <>>

block

In an ARIA window, an atomic instrument, knob, or outjack that can be patched to other blocks with "wires," so that the whole can be compiled into a new single instrument.

blockport

In an ARIA patch window, a triangular protuberance on a block that represents either an input or output, and can be connected to a blockport with a wire. See also block.

buffered message

A message with an auxiliary buffer for data storage. This message is used to send more than eight bytes of data within the message itself. It is also referred to as a pass-by value message.

byte

Eight bits, considered unsigned unless otherwise stated; usually used to store flag bits or ASCII characters.

capture mask

Type of event mask that lists all event types that the event broker should check the status of whenever it sends an event message.

CCB

See cel control block.

CD Browser

See 3DO CD Browser.

cel

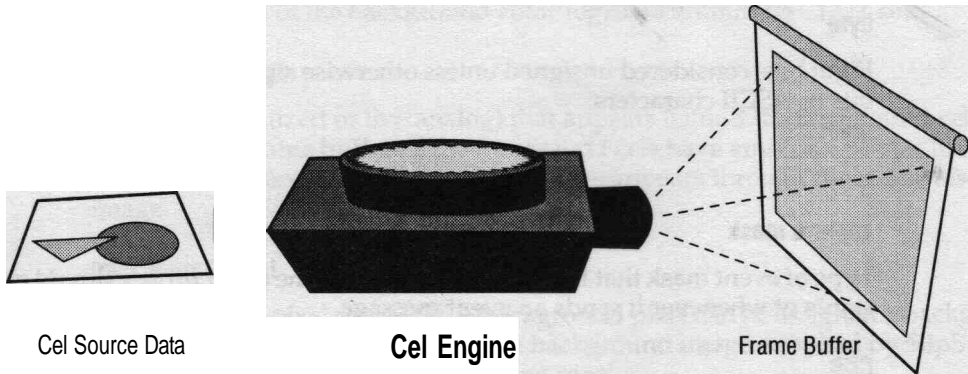
See 3DO cel.

cel control block (CCB)

A data structure that contains information controlling a cel's projection by the cel engine. The CCB contains the cel's location in the frame buffer, the shape of the cel projection quadrilateral, a pointer to the cel's source data, values for fixed luminance, values for fixed cornerweight position, links to other cels, and more.

cel engine

The part of the graphics hardware that projects a 3DO cel into the frame buffer.



cel engine control word

A 32-bit value whose bits primarily control how the cornerweight and least-significant bits pass through the projector.

cel list

You can create a list of 3DO cels by setting the `ccb_NextPtr` field in a cel to the next cel. The cel engine draws a list of cels more quickly than it draws the cels individually. To end the cel list, turn on the `CCB_LAST` bit in the `ccb_Flags` field.

cel origin corner

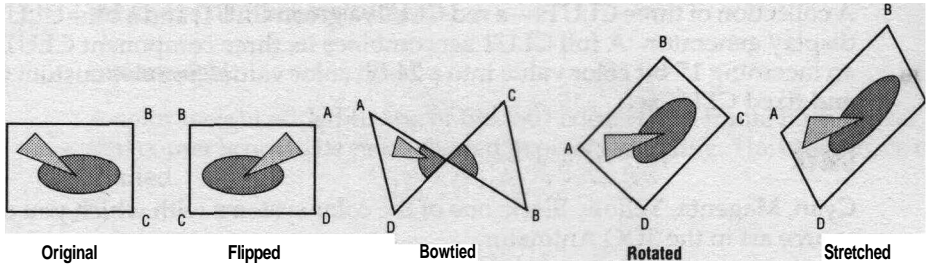
The frame buffer location ofThe first corner of the cel projection quadrilateral. The locations of the other three corners are computed based on this corner.

cel pixel

A pixel (1-16 bits deep) of a bitmapped image that is processed by the cel engine and then projected into the frame buffer. See also frame buffer pixel.

cel projection quadrilateral

A four-sided area in the frame buffer (defined by four points) into which the cel engine projects the contents of a cel. The shape of the cel quadrilateral determines how a cel is sized or distorted during projection. You can flip the corners, stretch the corners along one or both dimensions, and even "bowtie" the cel.



cel RAM

Any RAM accessible to the cel engine.

CFBV

See current frame buffer value.

chunk

A file in 3DO file format consists of chunks. A typical chunk file holds data for one media type. Each chunk consists of a chunk header with the chunk ID and size, and a chunk body that contains the data.

Currently available chunks include the cel control chunk, image control chunk, PDAT (pixel data) chunk, animation chunk, PLUT chunk, and VDL chunk. Some chunk types require other chunks in front of them; for example, a PDAT chunk has to be preceded by a cel control chunk or an image control chunk.

class

A body of code that defines an object. The class specifies an object's variables and methods and defines how those methods affect the object variables.

Content Library

See 3DO Content Library.

GLUT

Color lookup table, a collection of color registers in the display generator. A GLUT interprets an incoming 15-bit color value from the frame buffer by looking up the contents of the register that has the same number as the incoming red, green, and blue values.

GLUT set

A collection of three CLUTs—a red GLUT, a green GLUT, and a blue GLUT—in the display generator. A full GLUT set combines its three component CLUTs to turn an incoming 15-bit color value into a 24-bit color value. See also custom GLUT set and fixed GLUT set.

CMYK

Cyan, Magenta, Yellow, Black; one of the color systems with which you can create source art in the 3DO Animator.

codec

Compressor/decompressor. A codec is a methodology for compressing and decompressing video images.

coded cel

A 3DO cel containing pixels that do not have direct RGB colors but contain indexes into a color table that is part of the PLUT (pixel lookup table). A coded cel must pass through the pixel decoder in the cel engine. The PLUT, which is part of the cel control block and is used by the cel engine, returns a 15-bit color value for each incoming pixel, so that the pixel can then be projected into the frame buffer. See also uncoded cel.

collection class

A class that provides a useful tool for creating higher-level event structures: assemblies of sequences, assemblies of other collections, or assemblies of sequences and collections combined.

color averaging

See pixel weighting.

color depth

The number of bits per pixel used to define color. For the 3DO Station, you can work with 1, 2, 4, 5, 6, 8, and 16 bits per pixel.

color lookup table

See GLUT.

color register

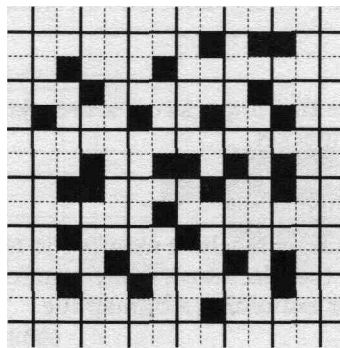
A register within a GLUT indexed with a number from 0 to 31. The color register contains an 8-bit red, green, and blue value that corresponds to an incoming 5-bit red, green, and blue value.

control signal instruments

A type of instrument that generates a low-frequency signal typically used to tweak the knobs of other instruments.

cornerweight

A color weight added to one of the four corners of a frame buffer pixel to specify that corner is not to be color-shaded in pixel weighting. The other three corners are shaded.



Pixel

Cornerweight

cornerweight bits

Any pixel bits (in either the frame buffer or a cel) used to specify a cornerweight position. A cornerweight position is specified with either one or two bits. Cornerweight bits are also called *VH bits*.

cornerweight position

The specified corner of a frame buffer pixel where a cornerweight appears.

cornerweight value

A value carried by two cornerweight bits used to determine cornerweight position. Also called *VH value*.

current frame buffer value (CFBV)

The 15-bit color value contained by the first frame buffer pixel in a cel pixel's projection location. This value is used in the pixel processor.

custom GLUT set

One of two GLUT sets within the display generator. The custom GLUT set contains writable color registers that you can set with the video display list processor (VDLP). See also GLUT set; compare fixed GLUT set.

cyclic mode

A running mode of the digital signal processor (DSP). Cyclic mode regularly resets the DSP's state and starts instruction execution from the beginning of instruction memory. The reset cycle can be set by 3DO timers or by the crystal of the DAG chip.

DAC

Digital-to-analog converter, a component of the display generator that converts the digital pixel stream from the interpolator into an analog NTSC or PAL signal.

data unpacker

A component of the cel engine that unpacks cel source data if the data is packed. Also known as DUP or the unpacker.

Data window

A window for looking at the target machine memory in hexadecimal and ASCII format that you can create on the Macintosh while running the 3DO Debugger. This is useful, for example, for examining a bitmap file.

debugger

See 3DO Debugger.

«*•

decoder

See pixel decoder.

delay line

A reverberation tool that temporarily stores an audio signal and delays it.

delay instrument

A reverberation tool that writes an audio signal into a delay line. *

digital signal processor (DSP)

A programmable processor designed to process digital audio.

direct memory access

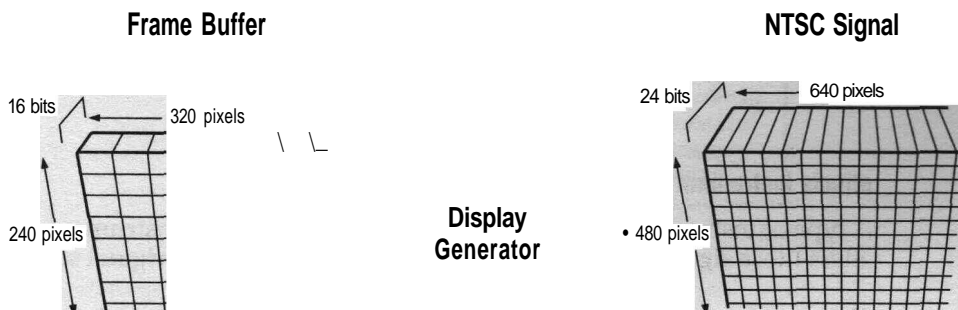
See DMA.

display control command

A command in the video display list that sets a display generator parameter. Each display control command is tied to one horizontal line of the NTSC or PAL display and is not executed until the horizontal blank before that line.

display generator

The part of the graphics hardware that turns the pixels stored in the frame buffer into a color NTSC or PAL signal.

**display mode**

Any one of several methods of interpreting the 16 bits of a frame buffer pixel as it passes through the display generator. The display mode sets the number of bits used for color value, cornefweight value, and color table value. See also 1555 mode, 15541 mode, P555 mode, P5541 mode.

dithering

A technique that creates the effect of additional colors by mixing a picture's available colors. You can use dithering when there are not enough bits in the frame buffer to represent true color.

DMA

Direct memory access. A way of moving memory without using the CPU.

double-buffering

An animation technique that rapidly alternates between two frame buffers: one for feeding data to the display generator, the other for receiving data from software or the cel engine. Each time the display generator finishes rendering one frame, the two frame buffers are swapped. In this way, a frame buffer is never displayed while rendering is in process.

driver

See 3DO driver.

DSP

See digital signal processor.

DUP

See data unpacker.

effects instrument

A type of instrument that accepts audio signals from other instruments and then alters those signals in ways (such as filtering) that alter the original sound.

EI memory

See external input memory.

envelope

The shape formed by the changes in amplitude of a sound wave, such as that produced by the decay in volume after a string is plucked. See also envelope player.

envelope player

Envelope players read the envelope data and apply it to an instrument attribute during note play.

EO memory

See external output memory.

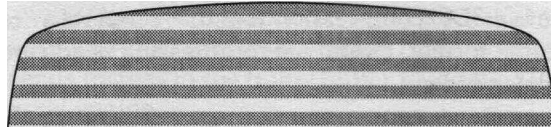
EPSF


Encapsulated PostScript file format. You can convert an EPSF file using the 3DO Animator tool.

41

even field

A field that contains all the even-numbered scan lines of a frame. The 3DO system uses this field to present left-eye images in a stereoscopic display. Compare odd field.



Odd Field  Even Field

event

Activities on interactive devices. An event includes activities such as control port button presses and releases, control pad presses and releases, key strokes, mouse interaction, and so on.

event broker

A task that constantly monitors activity on attached interactive devices.

event broker message

Message through which all communication between the event broker and connected tasks takes place.

external input memory

The part of the digital signal processor's data memory dedicated to accepting input from input DMAs and the ARM CPU. The DSP can read external input memory, but it can't write to it.

external output memory

The part of the digital signal processor's data memory dedicated to providing output to output DMAs and the ARM CPU. The DSP can write to external output memory, but it can't read it.

EZ Squeeze

EZ Squeeze is a 3DO compression tool that converts a "raw" QuickTime image into an internal format. Useful for compressing headshots, small-frame video, and blue-screen video.

field

One half of the scan lines contained in a video frame. An NTSC display presents a frame by alternately displaying even and odd fields 60 times per second; a PAL display does the same thing 50 times per second. See also even field, odd field.

file format

See 3DO file format, chunk.

fixed GLUT set

One of two GLUT sets within the display generator. The fixed GLUT set contains fixed linear color values that you can't change. See also GLUT set; compare custom GLUT set.

FlashWrite

A 3DO hardware feature that quickly copies the contents of a single 32-bit register to all the pixels of a frame buffer to clear the buffer to a single color.

flavor

Identifies the purpose of an event broker message. The flavor of the message determines the type or types of data structures contained in the data block, and specifies the way the message recipient should handle the message.

folio

A library of functions that provide a software interface to system hardware. Examples are the Kernel folio, Graphics folio, Math folio, and File folio.

folio type

A 32-bit integer argument, used with item creation, that specifies both the folio in which the item is used, and, within that folio, the type of item.

frame

One fully rendered still image sent to the NTSC or PAL display. To create a moving image, the display generator sends 30 frames per second to the display on NTSC and 25 frames per second on PAL.

frame buffer

The portion of VRAM (video RAM) used to store the bitmapped contents of the current frame. The 3DO hardware projects cels onto the frame buffer via the cel engine and sends images to the frame buffer using the SPORT mechanism. For illustration, see cel engine.

frame buffer pixel

A 16-bit pixel in the frame buffer. See also cel pixel.

frame buffer pointer

A pointer to the beginning address of the frame buffer in VRAM .

free-running mode

A running mode of the display signal processor (DSP) in which the DSP executes all the instructions in instruction memory and then stops until restarted.

frequency

Generally, the pitch, or "highness" of a sound. Measured in Hertz, MIDI pitch numbers, or 16-bit raw known values.

A reserved special name for any knob that is to respond to MIDI pitch numbers.

GrafCon

A graphics context data structure. This structure keeps track of the current status of the pen.

H**half-word**

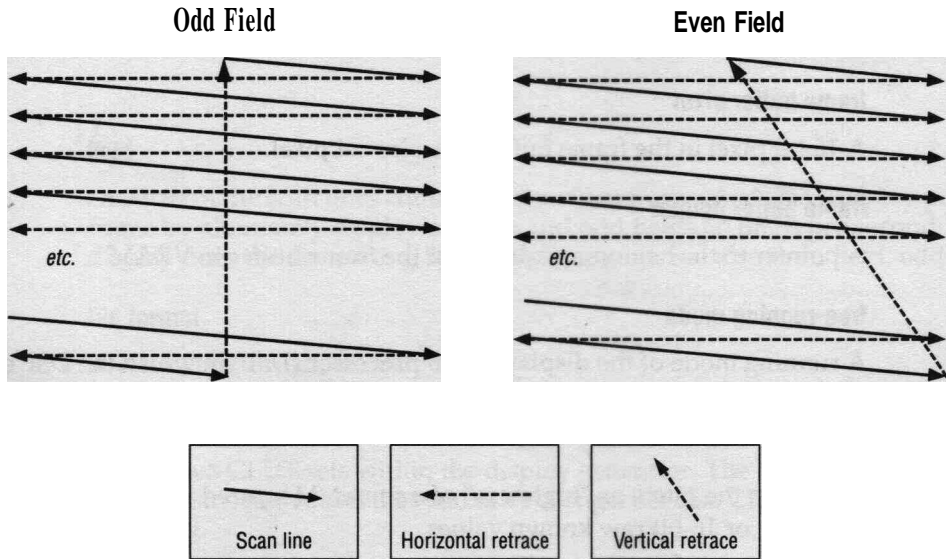
16 bits, or 2 bytes.

head-of-list anchor

A union that marks the beginning, or head, of a linked list.

horizontal blank

The rendering time between the end of one scan line and the beginning of the next scan line. Also known as *horizontal interrupt*. During this time, the NTSC or PAL display's electron gun is turned off. Because many 3DO events are tied to the horizontal blank, conditions can change between one scan line and the next. Compare vertical blank.



host

The Macintosh on which you prepare and compile a program. The compiled program runs on the target machine.

HSL

Hue, saturation, luminance; one color system with which to create source art in 3DO Animator.

HSV

Hue, saturation, value; one color systems with which to create source art in the 3DO Animator.

ILBM

The file format for animations from Deluxe Paint Amiga. You can convert an ILBM file using the 3DO Animator tool.

image

See 3DO image.

input focus

Determines which task should respond to the user's actions when several tasks occupy the screen at the same time.

instance

A single implementation of a class or an object defined by a class.

internal memory (I memory)

The part of the digital signal processor's (DSP) data memory dedicated to internal DSP work. The DSP alone can read from and write to internal memory.

input DMA channel

One of thirteen DMA channels in the digital signal processor (DSP) dedicated to writing information from DRAM, passing it through a FIFO buffer, and writing it to external input memory.

instruction memory (N memory)

The code memory of the digital signal processor (DSP) that is used to store DSP instruction code. The ARM CPU can write to instruction memory; the DSP can only read from instruction memory.

instrument

An externally created piece of digital signal processor code that generates or modifies a digital audio signal. An instrument can contain inputs, outputs, knobs, and a sampled sound pointer that feed it audio, allow it to modify the audio, and receive audio.

instrument resources

The digital signal processor's data memory, instruction memory, and ticks allocated to an instrument for playback.

instrument table

A reference table kept by the audio folio that lists the hardware resources available for each instrument name. When a note call asks for an instrument by name, the instrument table provides the highest-quality hardware available for that instrument.

instrument template

A DSP program that defines the qualities of an instrument—its inputs, outputs, knobs, signals it generates, among other qualities.

interpolator

The component of the display generator that breaks each frame buffer into four subpixels and then shades the subpixels to create smooth, anti-aliased images using cornerweights.

item

A system-maintained handle for a shared resource. An item contains a unique number and a pointer to the resource. Tasks can use an item by referring to its item number. The term is also used generically to refer to the item resources themselves. For example, data structures, allocated RAM, I/O devices, folios, and tasks are items.

item number

A number that uniquely identifies an item.

item priority

A value that determines how frequently system resources are allocated to the item. Priority values can range from 0 through 255.

juggler

A hierarchical event scheduler controlled by the music library, that times and plays events. The juggler is part of an object-oriented programming environment in which you can assemble and play back score objects.

juggle class

The root class for all other juggler classes. Its elements include pointers for a set of standard methods common to all juggler objects.

knob

A defined part of an instrument that accepts input from a task and modifies a parameter of the instrument according to that input.

In an ARIA patch window, a blue-colored block that will become available as a knob in the compiled instrument. See also block.

left register

A 16-bit register in the external output memory of the digital signal processor; this register is connected to the left-channel DAC. The DAC reads the contents of this register and—44,100 times a second—converts it to the left channel of analog output audio.

linear ascending color table

The color table of a GLUT set whose registers contain regularly ascending red, green, and blue values. This type of color table must be used for consistent luminance and pixel mixing effects in the cel engine. Compare nonlinear color table.

linked list

A linearly connected collection of data structures. The Portfolio operating system provides many procedure calls specifically designed to manage linked lists.

list

A data structure that ties the anchor and node elements together in a linked list.

listener

Any task connected to the event³² broker in order to receive event notification or to pass data to pods.

M**Macintosh Programmer's Workshop (MPW)**

The Apple programming environment for developing programs on the Macintosh. It is primarily a shell in which the compiler, assembler, editor, and linker are run. The 3DO Toolkit supplements MPW with the compiler, tools, and code libraries needed to develop program code for the 3DO Station.

MacPaint file

A file generated by the MacPaint application on the Macintosh. You can convert a MacPaint file to 3DO file format using the 3DO Animator tool.

marquee

The "crawling ants" outline used on the Macintosh to indicate a selection.

memory fence

A kernel-imposed division of memory pages that dedicates separate pages for separate tasks. A user task cannot write to pages of memory outside the memory fence surrounding its own allocated pages of memory.

memory sharing

This term refers to the common memory block shared between a task and its threads, where the task and the threads can read and write from the same portion of memory simultaneously. This is usually accomplished by declaring global variables accessible to both the main task and its thread.

message

An item used to send 8 bytes of data from one task to another. A message is typically used to send a pointer to more data for the receiving task to read.

message port

An item created by a task to receive messages. The task cannot receive messages without a message port.

method

The function that is invoked when a message is sent to an object.

mixer

A type of instrument that accepts output signals from other instruments, mixes the signals together, and then passes its output signals directly to the DAC for audio output.

MIDI

Musical Instrument Digital Interface, a data exchange format designed to carry timed musical events between musical instruments or equipment designed to handle musical events.

MIDI channel

Channel, which ranges from 1 to 16, that allows MIDI messages to be sent to specific devices within a MIDI network.

MIDI file

A file in a format specified by the MIDI standard, which stores the musical events that make up a musical composition.

MIDIFileParser

Data structure that tracks the overall score parameters as the MIDI data is translated into a juggler object.

MIDI project

In ARIA, a window or document containing a graphical layout of blocks and wires, which can be compiled into a new instrument. See also block.

MIDI timing clocks

A unit used to measure timing units associated with each MIDI message.

MIDItimestamp

The number of MIDI ticks that have passed between the beginning of the score and the beginning of the MIDI event associated with the timestamp.

MooV

The native file format for QuickTime movie files. You can convert a MooV file to 3DO file format using the 3DO Animator tool.

MovieCompress

Video compression tool that uses Apple QuickTime codecs. Permits specification of frame rate, key frame rate, data rate, depth, and codec.

MovieEdit

Tool for playing and editing QuickTime movies. Functionality includes cut, copy, and paste at all stages of processing, except after compression. The MovieEdit window shows dynamic information about time code, time unit, and frame number that allows more precise editing than some other tools.

MPW

See Macintosh Programmer's Workshop.

N**N memory**

See instruction memory.

node

A single data structure in a linked list that carries information for a single component of the list.

node priority

A value that determines a node's location in a linked list, relative to the other nodes in the list.

nonlinear color table

The color table of a GLUT set whose GLUT registers contain red, green, and blue values that don't regularly ascend or descend with the register numbers. Compare linear ascending color table.

note

The sound created by an instrument when the instrument is asked to play. A note can be musical or nonmusical. It can, for example, be a sound effect.

note stages

The defining parts of a note's duration: the start stage, the release stage, and the stop stage.

Note Off

A MIDI channel message that turns off a voice at a set pitch in the receiving instrument.

Note On

A MIDI channel message that turns on voice in a receiving instrument and sets the pitch and amplitude of the voice.

NoteTracker

Data structure tied to an audio instrument. This structure is used to track whether an audio instrument is playing a note or not.

NTSC display

The display of a standard TV screen in the United States, defined by the National Television System Committee. Using the 3DO system, you can work on this display with a 320 x 240 color resolution and 640 x 480 contrast resolution.

NuBus communications card

A card that communicates between the Macintosh and the 3DO Station and is used by the 3DO driver. The NuBus card is provided with the 3DO development system; you must install it in the Macintosh according to the instructions in the Macintosh documentation.

NVRAM (nonvolatile battery-backed RAM)

Nonvolatile battery-backed RAM that is not intended to be addressed in a memory-mapped fashion, but rather is addressed as a simple filesystem structure.

object

A data structure with associated functions.

object list

The list of place holders that defines the collection's assembly of other juggler objects.

odd field

A field that contains all the odd-numbered scan lines of a frame. The 3DO system uses this field to present right-eye images in a stereoscopic display. See even field for comparison and for illustration.

outjack

In an ARIA patch window, a blue-colored block that will become available for connection to another instrument or to Output in the compiled instrument. See also block.

output DMA channel

One of four DMA channels in the digital signal processor dedicated to writing information from external input memory (after passing it through a FIFO buffer) into DRAM.

P555 mode

(The P stands for *Palette*.) A display mode that uses bit 15 as a palette selector bit, bits 14-10 as red value bits, bits 9-5 as green value bits, and bits 4-0 as blue value bits.

P5541 mode

(The P stands for *Palette*.) A display mode that uses bit 15 as a palette selector bit, bits 14-10 as red value bits, bits 9-5 as green value bits, bits 4-1 as blue value bits, and bit 0 as a single cornerweight bit.

packed cel

A 3DO cel whose source data has been compacted to reduce storage space. A packed cel must pass through the cel engine's data unpacker prior to further processing. See also cel engine, data unpacker. Compare unpacked cel.

packet

A component of a packed cel that contains a group of compressed pixels.

palette

The set of colors available within one of the display generator's GLUT sets.

palette selection bit

A frame buffer pixel bit used to specify which of the two GLUT sets within the display generator to use.

parent pixel

The originating frame buffer pixel for any set of four subpixels created by the display generator's interpolator. The parent pixel contains a 24-bit color value and a cornerweight position that are used to determine the color of each subpixel.

PatchBay

A Macintosh application or desk accessory (included with Apple MIDI Manager) that is used to configure connections between MIDI applications, including ARIA.

PatchDemo

A tool run from the 3DO Debugger, that reads an ASCII file containing a PatchDemo script allowing you to audition the resulting audio output. On-screen faders are displayed for all available knobs.

PatchDemo script

Text in simple language, stored in an ASCII file, that is understood by PatchDemo and also by ARIA. The script contains instructions to load instruments, connect instruments, and tweak knobs. ARIA can also export PatchDemo scripts from a graphical patch window, and import scripts, laying them out in a new patch window.

*

Patch document

In ARIA, a document that contains a graphical layout of blocks and wires, which can be compiled into a new instrument.

Patch window

In ARIA, a window containing a graphical layout of blocks and wires, which can be compiled into a new instrument.

PBM tools

See PPM tools.

PDAT chunk

Pixel data chunk, one of the chunks available as part of the 3DO file format. A PDAT chunk always has to be preceded by either a cel control chunk or an image control chunk.

PDC

See pixel decoder.

PDV

See primary divide value.

peek

The action a task takes when it reads the current setting of an instrument knob.

pen

An invisible cursor that moves through a bitmap as function calls are drawing graphics primitives or rendering text.

performance analyzer

Part of the 3DO Debugger tool. You can use the performance analyzer to monitor the percentage of execution time spent in different routines. Currently, this tool is not fully implemented.

permanent folio

A folio that constantly resides in 3DO RAM, ready to handle procedure calls. A permanent folio cannot be purged from RAM.

PICT

Apple's preferred file format for exchanging files between different graphics programs. You can convert a PICT file to 3DO file format using the 3DO Animator or 3DO Construction Set tool.

PIMap (program-to-instrument map)

Map used to associate program numbers with instrument template names.

Pitch Bend Change

A MIDI channel message that specifies an amount of pitch bend to apply to all voices in a channel.

pitch bend wheel

Used to bend the pitch of voices up or down in the current channel.

pixel data chunk

See PDAT chunk.

pixel decoder

A component of the cel engine that decodes cel pixels if they are part of a coded cel. It extracts a 15-bit color value and—optionally—a 2-bit cornerweight value and a 9-bit alternate multiply value. The pixel decoder is also known as the PDC or simply as the *decoder*.

pixel lookup table

See PLUT.

pixel processor

A component of the cel engine that can modify pixel color by scaling it (for lighting and shading effects) and/or merging it with pixel colors from another source (for translucence and other effects).

Every pixel drawn by the cel engine goes through the pixel processor. The processor loads in a primary source, usually the pixel itself. Every channel of the source (red, green, and blue) is multiplied by the primary multiply value and then divided by the primary divide value. Then, the secondary source is loaded. This value is usually the background pixel. The secondary source is divided by 1, 2, or 4, and then added to, subtracted from, or XORed with the primary source.

pixel weighting

A process that gives each frame buffer pixel added color weight in one of its four corners. This way, the image there can be anti-aliased for a smooth appearance in the NTSC or PAL display. Pixel weighting is also known as *color averaging*. See also anti-aliasing, cornerweight.

placeholders

Contains a pointer to a juggler object and a variable telling how many times to repeat that object in playback.

PLUT

Pixel lookup table, a collection of registers in the pixel decoder of the cel engine. These registers interpret the incoming value of each coded cel pixel and return a 15-bit color value. They may also return a cornerweight value, alternate multiply value, or P-Mode value.

P-Mode

A collection of pixel processor parameters that controls how the pixel processor handles incoming cel pixels. You can use P-Mode, for example, to make some of the pixels in a cel translucent and some of them opaque or bright.

P-Mode bit

A bit derived from a coded cel pixel that chooses either of two P-Modes for processing the pixel in the pixel processor.

P-Mode preset

A named P-Mode set by a 3DO Custom Plug-In. Each plug-in provides a small number of P-Mode presets (translucent, dim, bright, opaque). You can create additional P-Mode presets and save them in a file.

PMV

See primary multiply value.

Portfolio

See 3DO Portfolio.

PPM tools

Portable pixmap tools, a set of public domain tools available on the Toolkit CD-ROM that lets you convert files of almost any format to the PPM (portable pixmap) format from UNIX. The PPM files can then be converted to 3DO format. PPM tools are sometimes called PBM tools.

preamble

A data structure that identifies the data type of the source: packed or unpacked, coded or uncoded, the number of bits per pixel, and more. The preamble is usually included at the beginning of the source data, but can in special cases be included in the cel control block.

primary divide value (PDV)

The value used to divide the primary source color value in the pixel processor. This division occurs after multiplication by the PMV. See primary multiply value.

primary multiply value (PMV)

The value used to multiply the primary source color value in the pixel processor. Compare primary divide value.

primary scalar

The value created by combining the primary multiply value and the primary divide value, used to scale the incoming primary source color value in the pixel processor.

primary source

The first of two sources of color data fed into the pixel processor. The primary source data can be scaled up or down by the pixel processor and then optionally merged with secondary source data. Compare secondary source.

privileged mode

A permission mode in which system tasks run. This mode gives the system task access to special kernel functions such as those that create device drivers.

Program Change

A MIDI channel message that specifies a new program (numbered from 0 to 127) for all receiving instruments tuned to the channel.

projector

A component of the cel engine that puts the cel's processed color values, along with optional VH (cornerweight) values, into an area in the frame buffer defined by the cel projection quadrilateral.

quadrilateral

See cel projection quadrilateral.

rasterization

The process of displaying an image as a set of horizontal scan lines on a television set.

ready queue

A queue of ready-to-run tasks awaiting execution by the CPU. Compare waiting queue.

Red Book audio

Audio stored in the compact disc format used for commercial recording releases ("record store" CD format).

register

See color register, left register, right register.

release stage

The time when a note is asked to finish. Compare stop stage.

Remote folder

A Macintosh directory regarded as the root (\$boot) directory of the 3DO file system when the Portfolio is running the 3DO Debugger.

In ARIA, you must set the directory that agrees with the Remote folder that the 3DO Debugger is using.

resolution

The width and height of an image, as measured in pixels. You can use the 3DO hardware to work in 320 x 240 and 640 x 480 resolution for NTSC, and 384 x 288 and 768 x 576 for PAL.

reverberation

An effect that adds reality to a sound, providing acoustic cues about the size and reverberant qualities of the sound's surroundings.

right register

A 16-bit register in the external output memory of the display signal processor. This register is connected to the right-channel DAC. The DAC reads the contents of the register and—44,100 times a second—converts it to the right channel of analog output audio.

sample

A digital waveform stored on disk or in memory. The audio folio initially searches for sample files in the \$samples directory. See also AIFF, alias.

sampld-sound instrument

A type of instrument that plays back sampled-sound tables.

sampled-sound pointer

The part of an instrument that lists a sampled-sound file to play.

scan line

One of 525 horizontal lines traced by an electron gun on an NTSC display, or one of 625 lines on a PAL display.

score

An ordered structure of events played back by the audio folio's high-level scheduler. A score can include musical information, sound effects information, and non-musical events such as external function calls.

ScoreChannel

Data structure used to keep track of the state of a synthesizer's channels.

ScoreContext

Data structure used to rack the overall playback of a music score.

screen

A structure that is a single VDL (Video Display List) with accompanying bitmap or bitmaps used to store image pixels.

script file

A file that starts the 3DO Debugger. Default scripts are provided for different versions of the 3DO hardware. After the script file has run, you can type in the Debugger Terminal window on the Macintosh. You can edit the script files to create a custom script that runs your program.

secondary divide value (SDV)

The value used to divide the secondary source color value in the pixel processor.

secondary source

The second of two sources of color data fed into the pixel processor. The secondary source color data can be scaled down by the pixel processor, or the data can be set to 0. The results are then merged with the primary source color data.

sequence

Contains an array of events to be executed in time. A sequence typically contains an array of MIDI messages from a MIDI file.

sequence class

Used to play sequences of notes—or other events, audio or otherwise. This class contains all of the juggler's methods and variables and adds connections to an event list, a list of timed events.

serial port bus

See SPORT.

signal

A kernel mechanism that allows one task to send one or more 1-bit flags to another task.

signal mask

A 32-bit word stored in a task's task control block (TCB) that specifies the bits of a 32-bit signal word on which the task can receive signals.

Slipstream

An optional hardware feature that replaces any background pixels with pixels from a digitized background image if an optional GenLock unit is attached.

sound file player

A collection of parameters that defines how the sound spooler plays back an AIFF file.

sound-synthesis instrument

A type of instrument that synthesizes audio signals from scratch.

source data

The bitmapped image data for a cel.

SPORT (Serial PORT)

An acronym for serial PORT bus, an additional serial bus available for VRAM that can take streams of bits from 512 x 512 arrays and generate video signals. It can also quickly copy pages of VRAM to new page locations in VRAM.

standard message

Another term for a message. Standard messages have no auxiliary buffer for data storage. They send eight bytes of data in the message that point to a separate data block. Also referred to as a pass-by-reference message.

start stage

The time when a note is asked to start. Compare release stage.

stereoscopic display

A method of displaying 3-D images on the 3DO Station by alternating left and right eye views in odd and even fields of the display, and then using LCD glasses to limit the viewer's left eye to the even field of the display and the right eye to the odd field of the display. Creating two images in slight parallax results in an illusion of depth.

stop stage

The time when a note is no longer playing.

stream

A woven set of chunk files with optional header description, block size, media type, subscribers, and possible control branch tables. An example of a stream is an audio chunk file woven with a Cinepak chunk file to make a movie stream.

streamblock

A stream file consists of streamblocks, which in turn consists of interleaved chunks. The DataStreamer performs disc reads on a streamblock-by-streamblock basis. See also 3DO DataStreamer, chunk.

stream file

A file that contains interleaved data that the 3DO DataStreamer library can parse and display. To create a stream file, use the Weaver tools and a Weaver script.

string

A sequence of bytes terminated by a NULL (0x00) byte interpreted as text.

submixer

A type of instrument that accepts output signals from other instruments, mixes the signals together, and then passes its output signals through outputs that can be fed to other instruments.

subpixel

The display generator increases the incoming frame buffer resolution of 320 x 240 to a resolution of 640 x 480 for NTSC and 384 x 288 to a resolution of 768 x 576 for PAL, by dividing each pixel into four subpixels.

super cel clipping

Technique used by the projector to save time by skipping cels that do not fall within the frame buffer boundaries. Compare super row clipping.

super row clipping

Technique used by the projector to save rendering time by skipping rows of pixels that do not fall within the frame buffer. Compare super cel clipping.

supervisor mode

A running mode that gives a task special permissions and access to system-only memory and resources. Only system tasks can run in supervisor mode. Compare user mode.

SVHS monitor

See S-Video monitor.

S-Video monitor

A monitor that is super-VHS-capable. Also called SVHS monitor.

system amplitude

The DAC amplitude of 0x7FFF.

system cycling

One type of the DSP cyclic running mode where the cycle time is set by the 3DO hardware's internal timers.

system memory pool

The entire amount of memory available in the system.

system task

One of the Portfolio operating-system's own tasks. A system task has direct access to system resources. Compare user task.

tag args

A list of parameters used to define an item structure. This list is used in the procedure call that creates the item structure.

target

The 3DO Station on which you run a program. Before doing this, you must compile the program on the host Macintosh.

target monitor

The part of the 3DO Debugger that performs low-level hardware control in response to the 3DO driver. The 3DO driver communicates with the target monitor through the NuBus communication board and launches the target monitor each time you reset the target hardware. The target monitor resides on the 3DO Station.

task control block (TCB)

A system data structure that contains the parameters of a running task.

tail anchor

A union that marks the end, or tail, of a linked list.

task priority

A value from 10 to 199 that sets a task's execution priority. 10 is the lowest priority, 199 is the highest priority. The lowest-priority tasks don't execute while higher-priority tasks are in the ready queue.

task state

A task's current running status, which can be either waiting, ready to run, or running.

TCB

See task control block.

Telecine process

Process of converting film that displays at 24 frames per second to video that displays at 30 fps by inserting intermediate, or composite, frames. During Telecine conversion, each set of four film frames is converted into interlaced video fields and the video fields are reconstructed into five video frames.

Terminal window

A window the 3DO Debugger displays on the Macintosh. From the Terminal window, you can communicate with the 3DO Station. Typing the name of an application (followed by Enter) in the Terminal window causes the application to execute on the 3DO Station.

thread

A task created and launched by another task. A thread shares the memory and resources of its parent task and dies if the parent task dies. See also memory sharing.

ticks

See audio ticks.

TIFF

Tag Image File Format, a file format supported by most scanner manufacturers and page layout programs. You can convert a TIFF file using the 3DO Animator tool.

time quantum

One slice of CPU time devoted to the execution of a single task.

title safe

Regardless of the size or brand of monitor you use, an entire animation or image will be visible on screen, without a black border, as long as there is a minimum resolution of 288 x 216 pixels.

Toolkit

See 3DO Toolkit.

translucent cel

A 3DO cel through which you can see the background. Translucency lets you simulate a wide array of effects—stained glass, spotlights, smoke, ghosts, and more.

transparent cel

A 3DO cel in which all black pixels (000) are not drawn, creating a transparent region for the cel.

transparent pixel

A cel pixel that is not projected into the frame buffer, and so doesn't affect the underlying frame buffer pixel. ^Iompare background pixel.

trigger mask

Type of event mask that lists all event types that will trigger the event broker to send an event message to the listener.

tuning table

A lookup table that provides a frequency for each standard pitch name given in a note call.

tweak

The action a task takes when it sends a value to change an instrument knob's setting.

uncoded cel

A 3DO cel containing pixels that have direct RGB colors—that is, values that are turned into 16-bit frame pixel values without passing through a PLUT. Compare coded cel.

unfolding

The process of turning an 8-bit uncoded cel pixel into a 16-bit value for projection into the frame buffer.

unpacked cel

A 3DO cel whose source data is not compacted, and can go directly into cel engine processing without passing through the data unpacker. Compare packed cel.

unpacker

See data unpacker.

user mode

A running mode that limits a task to write only to its own allocated memory. A task running in user mode does not have direct access to system resources. Compare supervisor mode.

user task

A task that does not have direct access to system resources.

VDAN

Video/32 animation file format, the native file format of the 3DO Animator tool. You can convert a VDAN file to 3DO file format using the 3DO Animator tool.

VDL

See video display list.

VDLP

See video display list processor.

vertical blank

The rendering time between the end of one field and the beginning of a new field. Also known as *vertical interrupt*. During this time the NTSC or PAL display's electron gun is turned off. Because many 3DO events are tied to the vertical blank, conditions can change between one field and the next. For comparison and illustration, see horizontal blank.

VH bits

See cornerweight bits.

VH value

See cornerweight value.

video display list (VDL)

A list of display control commands that tell the video display list processor how to control the display generator.

video display list processor (VDLP)

A graphics folio component that controls the display generator. It reads the video display list for control instructions.

video overlay

The image contained in the frame buffer when that image is laid over a background image supplied by Slipstream.

voice

The hardware resources necessary to play one note at a time.

VRAM (Video Random Access Memory)

The portion of 3DO's memory dedicated to storing video images and data structures. VRAM includes a SPORT bus for quick block-memory transfers. Standard DRAM does not.

VRAM bank

A contiguous one-megabyte section of VRAM. 3DO memory management restricts some operations to a single bank of VRAM.

W**waiting queue**

A queue of tasks to be executed by the CPU. When a designated external event occurs, a task in the waiting queue moves to the ready queue.

wrapper chunk

A wrapper chunk is one of the chunks supported by the 3DO file format. You can mark a file as a 3DO file using the 3DO wrapper chunk ID.

word

A unit of data. On the 3DO hardware, a word is 32 bits, or 4 bytes, wide.

Yellow Book CDs

Compact discs containing CD-ROM information stored in the Yellow Book format.