Contents

Introduction. 7
Purpose of this book 8
Prerequisites 9
Overview of the tools for Notes database and document manipulation. 10
When to call the C API from LotusScript 13
LotusScript and C functions 15
Use of C functions. 16
Information sources 17
Source of the problem 18
Calling C from LotusScript. 19
Use of C samples and sequence of conversion . 21
Getting started — OSLoadProgram sample 23
Summary 29
Crash course in C for LotusScript programmers31
What is C 32
Basic structure of C program or "Hello World". 33
How C differs from LotusScript. 36
Datatypes 37
typedef 38
Pre-processor directives. 39
#include 39
. 39
#define 39
#undef. 39
#ifdef name. 40
#ifndef name. 40
#endif 40
#else. 40
#pragma. 40
\ (backslash) 40
Comments 41
Constants 42
Bits and bytes 43
Hexadecimal notation 43
Signed and unsigned variables 43
Shifts 46
Operators 48
Variables, structures, arrays and the use of pointers 51
Simple variables. 51
Arrays and strings 51
Pointers 53
Casts. 55
Conversion of C headers to LotusScript. 57
Summary 66
LotusScript to Lotus C API basics 67
Introduction to C API call declarations 68
Basic declaration transformation 68
Parameter conversion 69
Dealing with call-back functions and passing null pointers. 72
Multiple declarations. 74
Data and reference types 76
Data structures 79
Endian issues 81
Data alignment. 83
Alignment in LotusScript 84
Alignment in C 84
Structures and byte datatype in R5. 86
Calling conventions and parameters. 88
Summary 92
Implementation, limitations and special considerations 93
Inherent limitations 94
Memory handling 95
Handling of Notes data buffers 95
Memory allocation and reallocation. 97
Reading and writing data to and from Notes documents 98
Reading an item. 98
Moving data in memory and reading the data buffers 99
Writing a data buffer 101
Wrapper functions and classes 103
Hiding multi-platform issues using wrappers. 103
Samples of wrappers 103
Encapsulating memory management in classes 109
Error handling 112
National character sets and necessary conversions. 113
LMBCS and Unicode conversion 113
Conversion by using a declaration 113
Use of OSTranslate 114
National character support 115
Multi-platform considerations 118

Platform identification 118
Handles 119
Endians. 119
Alignment matching 120
Pointers 124
Synopsis of platform specifics. 126
iSeries specifics 127
Problems 127
Workaround — wrapper srvpgm 127
Using Lotus C API from Visual Basic 132
Short overview of the used Notes structures . . 134
BLOCKID 134
DBREPLICAINFO 135
CD-record basics, text and rich text handling. 135
CDLINK2, CDLINKEXPORT2 and CDRESOURCE 136
IDTable 137
Special purpose items 137
Use of NotesPeek. 142
Use of NSD logs to diagnose problems 144
Summary 149
Ready-to-use samples 151
Getting special documents (About, Using, Icon). . 152
Conversion of C API sample code for the use in LotusScript 154
Introduction. 154
Steps to convert Lotus C API Toolkit sample to Notes LotusScript agent 154
Programmatic creation of About document157
Item information and Memory management. . 165
Declarations 165
Main agent code 167
Class memoryManager 169
Checking users' access rights. 171
Getting server statistics 174
Retrieving and updating Db replica information, timedate handling. 177
Declarations 177
Agent code 179
Class memoryManagerExt 182
Selective replication settings 184
Declarations 184
Main agent code 186
Used classes 195
Class aServer 195
Class srcServer. 195
Class recServer 196
Class ReplicationTable 198
Delete Agent Data 201
Declarations section 201
Main Agent code 202
Delete without leaving a replication stub 205
Setting OLE Object flags (editing $OLEOBJECT fields) 207
Retrieving and modifying $Fonts 211
Retrieving and modifying doclinks 218
Declarations 218
Main agent code 219
Editing reference type items 225
Importing RTF File 230
Main agent code 231
Getting an information on available servers . . . 236
Retrieving unread documents in views and updating unread records 239
Declarations 239
Main agent code 239
Class unreadTable 240
Getting available shares (Win32) 245
FileOpen Dialogue (Win32) 248
Init session (Visual Basic) 250
Retrieval of Notes Rich Text and callbacks (Visual Basic). 252
Refreshing design, callbacks and string conversion (Visual Basic) 255
Summary 260
Conclusion 261
Index 263
Appendix i
Data types and Symbol values ii
Functions xiii
ASCII table xxxvi