



Note: Before using this information and the product it supports, read the information in "Notices" at the end of this document.

First Edition (December, 2005)

This edition applies to IBM® Lotus® Domino® Designer 7 (product number L-GHUS-5RWNHM), and to all subsequent releases and modifications, until otherwise indicated in new editions.

© Copyright International Business Machines Corporation 1994, 2005. All rights reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Introduction to LotusScript. . . 1

What is LotusScript?	1
Advantages of LotusScript.	1
Working with scripts	2
Working in the script editor	3
Working with Lotus software	4
Determining which product file is being used	4
Debugging applications.	5

Chapter 2. Script and Statement

Construction Rules 7

Statement construction rules	7
Example	7
Literal number construction rules	7
Literal string construction rules	8
Identifier construction rules	9
Escape character for illegal identifiers	9
Examples	9
Labels	9
Keywords	10
Alphabetical listing of LotusScript keywords	10
Special characters	13

Chapter 3. Data Types, Constants, and Variables. 17

Summary of LotusScript data types	17
Data type conversion	18
Numeric operations	18
Argument passing	18
Variant variables.	18
Explicit data type conversion	19
Automatic data type conversion	20
Constants and Variables	22
Scope of declarations	23
Name conflicts and shadowing	23
Module scope	23
Procedure scope	24
Type or class scope	24
Constants	25
Built-in constants	25
Constants defined in LSCONST.LSS	26
Constants defined in LSPRVAL.LSS	26
Product-specific constants	26
User-defined constants	26
Variables	29
Declaring scalar variables explicitly	29
Declaring scalar variables implicitly	33
Examples of scalar variables	34
Arrays	36
Fixed arrays	39
Dynamic arrays	43
Lists	45
Working with lists	47
Variants	49
Boolean values	51

Dates/Time	52
Referring to Variants	54

Chapter 4. Expressions and Operators 57

Overview of expressions and operators	57
LotusScript operators	57
Operator order of precedence	58
Examples	58
Table of numeric operators	59
Arithmetic Operators	60
Exponentiation operator	61
Negation operator	62
Multiplication operator	62
Division operator	63
Integer division operator	63
Mod operator.	64
Addition operator	64
Subtraction operator	65
Relational (comparison) operators	66
Syntax	66
Elements	66
Return value	66
Usage	67
String comparison	67
Data type conversion	68
Examples	68
Logical Operators	69
Bitwise Operators	70
Boolean Operators	71
Not operator	73
And operator.	73
Or operator	74
Xor operator	75
Eqv operator	76
Imp operator	77
Table of string operators	78
String concatenation operators	79
String relational(comparison) operators	80
Like operator.	81
Is operator.	83
Syntax	83
Elements	83
Usage	83
Example	84
IsA operator	84
Syntax	84
Elements	84
Usage	84
Example	84

Chapter 5. Procedures: Functions, Subs, and Properties 85

Procedures	85
Functions	85
Defining functions	86

Passing arguments by reference and by value	87
Assigning a return value to a function	89
Executing a user-defined function	90
Values that a function can manipulate	93
Subs.	96
Defining subs.	96
Executing a sub	97
Specialized subs.	98
Properties	99
Declaring and defining properties	99
Using properties	100

Chapter 6. File Handling. 103

File operations	103
Sequential files	103
Opening sequential files	103
Writing to sequential files	104
Reading from sequential files	104
Random files	105
Opening random files	105
Defining record types	105
Writing to random files in LotusScript	106
Reading from random files	106
Binary files	107
Opening binary files	107
Using variable-length fields	107
Writing to binary files	107
Reading from binary files	107
Reading, writing, and closing files	108
Opening files	109
Reading from files and writing to them.	109
Closing files	111

Chapter 7. Error Processing 113

Types of errors	113
Run-time error processing	113
Informational functions used in run-time errors	113
Statements used in run-time errors	116

Chapter 8. User-Defined Data Types and Classes 125

Overview of user-defined data types and classes	125
User-defined data types	126
Declaring a variable of a user-defined data type	126
Referring to member variables.	127
Conserving memory when declaring member variables	127
Working with data stored in files.	128
User-defined classes	129
Benefits of classes	129
Base classes	130
Declaring member variables	130
Defining member properties and methods.	130
Public and Private class members	133
Private class members	133
Initializing member variables	133
Public class members.	134
Referring to members of an object	134
Testing object references.	135
Deleting objects	136

Managing memory for objects	136
Derived Classes	137
Property and method overriding	139
Arrays and lists of classes	143
Working with object reference variables	144
Creating objects	144
Using the Set statement	145
Using Variants to hold object references	146
Language cross-reference	146

Chapter 9. Managing Flow in Scripts 147

Flow of execution	147
Flow control statements	147
Comments and the compiler directive	147
Declarations	147
Definition statements.	148
Errors	148
Statement labels	148
Block statements	148
Selecting one or the other with the If...Then...Else statement.	148
Specifying multiple test conditions with the If...Then...ElseIf statement	150
Making a choice with the Select Case statement	152
Branching statements.	153
Transferring control with the GoTo statement	153
Using the If...GoTo...Else statement to transfer unconditionally.	154
Conditional control transfer with the On...GoTo statement.	154
Transferring control within the same procedure with the GoSub, On...GoSub, and Return statements	155
Iterative statements	157
Do and Do...While loops	157
For...Next loops	159
ForAll loops for lists and arrays	163
Using the While statement	167
Early termination statements	167
Stopping procedure execution early using the End statement	167
Using the Exit statement for early procedure termination	168

Chapter 10. Managing Asynchronous Web Agents in Domino 171

Introduction to multithreading and synchronization in LotusScript	171
Advantages of thread-safe agents.	171
Agents run serially	171
Threaded agents	172
Synchronization functions	172
How synchronization works	172
Running asynchronous agents on the Domino server	175
Thread-safe LSX, C/C++ code	175
Thread-specific bugs	175
Creating and destroying locks	175

Chapter 11. Beyond Core LotusScript 177

Lotus software environments	177
Determining which product file is being used	177
Product classes and objects	177
Interacting with the user	180
MsgBox on Notes server context	182
Interacting with other programs	182
Functions and statements for working with other programs	182
OLE Automation	184
Calling external C language functions	185
Example	186
Declaring C functions	186
Passing arguments to C functions	187
Passing strings	188
Passing arrays, types, and objects	190
Using user-defined data type variables	192
Return values	193
Calling C language functions extended example	194
LS2J: Connecting with Java	195
About LS2J	195
System requirements	195
Using LS2J	195
LS2J classes	200
Data type mappings	225
LS2J extended example	227

Chapter 12. LotusScript Language

Reference 233

Abs function	233
Syntax	233
Elements	233
Return value	233
Usage	233
Language cross-reference	233
Examples: Abs function	233
ACos function	233
Syntax	234
Elements	234
Return value	234
Usage	234
Language cross-reference	234
Examples: ACos function	234
ActivateApp statement	234
Syntax	234
Elements	234
Usage	234
Examples: ActivateApp statement	235
ArrayAppend function	235
Syntax	235
Elements	235
Return value	235
Usage	235
Error handling	235
Extended examples: array and String functions	236
ArrayGetIndex function	239
Syntax	239
Elements	239
Return value	239
Usage	239
ArrayReplace function	240
Syntax	240

Elements	240
Return value	240
Usage	240
ArrayUnique function	242
Syntax	242
Elements	242
Return value	242
Usage	243
Language cross-reference	243
Examples: ArrayUnique function	243
Asc function	243
Syntax	243
Elements	244
Return value	244
Examples: Asc function	244
ASin function	244
Syntax	244
Elements	244
Return value	244
Language cross-reference	244
Examples: ASin function	244
ATn function	245
Syntax	245
Elements	245
Return value	245
Language cross-reference	245
Examples: ATn function	245
ATn2 function	245
Syntax	245
Elements	245
Return value	245
Language cross-reference	246
Examples: ATn2 function	246
Beep statement	246
Syntax	246
Usage	246
Examples: Beep statement	246
Bin function	247
Syntax	247
Elements	247
Return value	247
Usage	247
Examples: Bin function	247
Boolean data type	247
Usage	247
Examples: Boolean data type	248
Bracket notation	248
Syntax	248
Elements	248
Usage	248
Examples: Bracket notation	249
Byte data type	249
Usage	249
Examples: Byte data type	249
Call statement	250
Syntax 1	250
Syntax 2	250
Syntax 3	250
Syntax 4 (functions only)	250
Elements	250
Usage	251

Referencing a function that returns an array, list, or collection	251	Return value	263
Examples: Call statement	251	Language cross-reference	263
CBool function	252	Examples: CLng function	263
Syntax.	252	Close statement	264
Elements	252	Syntax.	264
Return value	252	Elements	264
Examples: CBool function	252	Usage	264
CByte function	253	Examples: Close statement	264
Syntax.	253	CodeLock function	264
Elements	253	Syntax.	264
Return value	253	Elements	264
Examples: CByte function	253	Return values	264
CCur function	254	Usage	265
Syntax.	254	Extended examples: lock functions	265
Elements	254	CodeLockCheck function	266
Return value	254	Syntax.	266
Examples: CCur function	254	Elements	266
CDat function	255	Return values	266
Syntax.	255	Usage	266
Elements	255	CodeUnlock function	266
Return value	255	Syntax.	266
Usage	255	Elements	266
Language cross-reference	256	Return values	266
Examples: CDat function	256	Usage	267
Cdbl function	256	Command function	267
Syntax.	256	Syntax.	267
Elements	256	Return value	267
Return value	256	Usage	267
Language cross-reference	256	Examples: Command function.	267
Examples: Cdbl function	257	Const statement	267
ChDir statement	257	Syntax.	267
Syntax.	257	Elements	267
Elements	257	Functions that can be evaluated as LotusScript constants	268
Usage	257	Usage	268
Examples: ChDir statement.	258	Examples: Const statement	269
ChDrive statement	258	Cos function.	269
Syntax.	258	Syntax.	269
Elements	258	Elements	269
Usage	258	Return value	269
Examples: ChDrive statement	258	Language cross-reference	270
Chr function.	258	Examples: Cos function	270
Syntax.	258	CreateLock function	270
Elements	258	Syntax.	270
Return value	259	Elements	270
Usage	259	Return values	270
Examples: Chr function	259	Usage	270
CInt function	259	CreateObject function.	270
Syntax.	259	Syntax.	270
Elements	260	Elements	270
Return value	260	Return value	271
Language cross-reference	260	Usage	271
Examples: CInt function.	260	Examples: CreateObject function	272
Class statement.	260	CSng function	272
Syntax.	260	Syntax.	272
Elements	261	Elements	272
Usage	261	Return value	272
Examples: Class statement	262	Language cross-reference	272
CLng function	263	Examples: CSng function	272
Syntax.	263	CStr function	273
Elements	263	Syntax.	273

Return value	273	Elements	282
Language cross-reference	273	Usage	284
Examples: CStr function	273	Passing arguments	284
CurDir function	273	Using LMBCS or Unicode strings	284
Syntax	273	Calling exported library functions in 32-bit versions of Windows	285
Elements	273	Examples: Declare statement (external C calls)	285
Return value	274	Declare statement (forward reference)	285
Usage	274	Syntax	285
Language cross-reference	274	Elements	285
Examples: CurDir function	274	Usage	287
CurDrive function	274	Examples: Declare statement (forward reference)	287
Syntax	274	Deftypestatements	287
Return value	274	Syntax	287
Examples: CurDrive function	274	Elements	288
Currency data type	274	Usage	288
Usage	275	Examples: Deftype statements	289
Examples: Currency data type	275	Delete statement	289
CVar function	275	Syntax	289
Syntax	275	Elements	289
Elements	275	Usage	289
Return value	275	Examples: Delete statement	289
Examples: CVar function	275	DestroyLock function	290
DataType function	276	Syntax	290
Syntax	276	Elements	290
Elements	276	Return values	290
Return value	276	Usage	290
Usage	276	Dim statement	290
Language cross-reference	277	Syntax	290
Examples: DataType function	277	Elements	291
About data types	277	Usage	292
Date function	278	Explicit declarations and implicit declarations	292
Syntax	278	Specifying the data type	292
Return value	278	Declaring arrays	292
Usage	278	Declaring lists	293
Examples: Date function	279	Declaring object reference variables	293
Date statement	279	Initializing variables	293
Syntax	279	Visibility of declarations	293
Elements	279	Examples: Dim statement	294
Usage	279	Dir function	295
Examples: Date statement	279	Syntax	295
DirectoryName function	279	Elements	295
Syntax	279	Return value	296
Elements	280	Usage	296
Return value	280	Examples: Dir function	296
Language cross-reference	280	Do statement	296
Examples: DateNumber function	280	Syntax 1	296
DateValue function	280	Syntax 2	296
Syntax	280	Elements	297
Elements	281	Usage	297
Return value	281	Terminating the loop	297
Usage	281	Language cross-reference	297
Language cross-reference	281	Examples: Do statement	297
Examples: DateValue function	281	Dot notation	297
Day function	281	Syntax 1	297
Syntax	281	Syntax 2	298
Elements	281	Elements	298
Return value	282	Usage	298
Language cross-reference	282	Examples: Dot notation	298
Examples: Day function	282	Double data type	298
Declare statement (external C calls)	282	Usage	298
Syntax	282		

Examples: Double data type	299	Examples: Execute function and statement.	308
End statement	299	Exit statement	309
Syntax.	299	Syntax.	309
Elements	299	Elements	309
Usage	299	Usage	309
Language cross-reference	299	Language cross-reference	310
Examples: End statement	299	Examples: Exit statement	310
Environ function	299	Exp function	310
Syntax 1	299	Syntax.	310
Elements	300	Elements	310
Return value	300	Return value	310
Language cross-reference	300	Usage	311
Examples: Environ function	300	Language cross-reference	311
EOF function	300	Examples: Exp function	311
Syntax.	300	FileAttr function	311
Return value	301	Syntax.	311
Usage	301	Elements	311
Examples: EOF function.	301	Return value	311
Erase statement.	301	Examples: FileAttr function.	311
Syntax.	301	FileCopy statement	312
Elements	301	Syntax.	312
Usage	302	Elements	312
Examples: Erase statement	302	Usage	312
Erl function	302	Examples: FileCopy statement.	312
Syntax.	302	FileDateTime function	312
Return value	302	Syntax.	312
Usage	302	Elements	313
Examples: Erl function	302	Return value	313
Err function	303	Examples: FileDateTime function	313
Syntax.	303	FileLen function	313
Return value	303	Syntax.	313
Usage	303	Elements	313
Language cross-reference	303	Return value	313
Examples: Err function	303	Examples: FileLen function.	313
Err statement	304	Fix function	313
Syntax.	304	Syntax.	313
Elements	304	Elements	314
Usage	304	Return value	314
Examples: Err statement.	304	Usage	314
Error function	304	Examples: Fix function	314
Syntax.	304	For statement	314
Elements	304	Syntax.	315
Return value	305	Elements	315
Language cross-reference	305	Usage	315
Examples: Error function	305	Executing the loop the first time	315
Error statement.	305	Executing the loop more than once	315
Syntax.	305	Exiting the loop early	315
Elements	305	Nested For loops	315
Usage	306	Language cross-reference	316
Examples: Error statement	306	Examples: For statement.	316
Evaluate function and statement	306	ForAll statement	316
Syntax.	306	Syntax.	316
Elements	307	Elements	316
Return value	307	Usage	317
Examples: Evaluate function and statement	307	Exiting the loop early	317
Execute function and statement	307	Using <i>refVar</i>	317
Statement Syntax	307	Language cross-reference	317
Function Syntax	307	Examples: ForAll statement.	318
Elements	307	Format function	319
Return value	308	Syntax.	319
Usage	308	Elements	319

Return value	319	GoTo statement.	339
Formatting codes	319	Syntax.	339
Numeric formats	319	Elements	339
Named numeric formats.	319	Usage	339
Custom numeric formatting codes	320	Examples: GoTo statement	339
Date/time formats.	321	Hex function	340
Named date/time formats	321	Syntax.	340
Custom date/time formatting codes.	322	Elements	340
String formatting codes	323	Return value	340
Formatting dates and times in Asian languages	324	Usage	340
Date/time format codes	324	Examples: Hex function	340
Examples: Format function	325	Hour function	340
Fraction function	327	Syntax.	340
Syntax.	327	Elements	341
Elements	327	Return value	341
Return value	327	Language cross-reference	341
Usage	327	Examples: Hour function	341
Examples: Fraction function	327	If...GoTo statement	341
FreeFile function	327	Syntax.	341
Syntax.	327	Elements	341
Return value	327	Usage	342
Usage	328	Language cross-reference	342
Examples: FreeFile function	328	Examples: If...GoTo statement	342
FullTrim function	328	If...Then...Else statement	342
Syntax.	328	Syntax.	343
Element	328	Elements	343
Return value	328	Usage	343
Usage	328	Language cross-reference	343
Function statement	329	Examples: If...Then...Else statement	343
Syntax.	329	If...Then...Elseif statement	343
Elements	329	Syntax.	343
Usage	330	Elements	344
Examples: Function statement	330	Usage	344
Get statement	331	Language cross-reference	344
Syntax.	332	Examples: If...Then...Elseif statement	344
Elements	332	%If directive.	344
Usage	332	Syntax.	344
Examples: Get statement	333	Elements	345
GetFileAttr function	334	Usage	345
Syntax.	334	Examples: %If directive	346
Elements	334	IMESetMode function	347
Return value	334	Syntax.	347
Usage	334	Elements	347
Examples: GetFileAttr function	334	Return values	348
GetObject function	335	Usage	348
Syntax.	335	Examples: IMESetMode	348
Elements	335	IMestatus function	348
Return value	335	Syntax.	348
Usage	335	Return value	348
Examples: GetObject function	336	Usage	348
GetThreadInfo function	337	Example	349
Syntax.	337	Implode function	349
Elements	337	Syntax.	349
Return values	337	Elements	349
Usage	337	Return value	349
Examples: GetThreadInfo function	338	Usage	350
GoSub statement	338	Error handling	350
Syntax.	338	Examples: Implode function	350
Elements	338	%Include directive.	350
Usage	338	Syntax.	350
Examples: GoSub statement	338	Elements	350

Usage	351	Usage	362
Examples: %Include directive	351	Examples: Integer data type	363
Input # statement	351	IsArray function	363
Syntax.	351	Syntax.	363
Elements	351	Elements	363
Usage	352	Return value	363
Examples: Input # statement	353	Examples: IsArray function.	363
Input function	353	IsDate function	363
Syntax.	353	Syntax.	363
Elements	354	Elements	363
Return value	354	Return value	363
Usage	354	Usage	364
Examples: Input function	354	Examples: IsDate function	364
InputB function	354	IsDefined function.	364
Syntax.	354	Syntax.	364
Return value	355	Elements	364
Usage	355	Return value	364
Examples: InputB function	355	Usage	364
InputBox function	355	Examples: IsDefined function	365
Syntax.	355	IsElement function	365
Elements	355	Syntax.	365
Return value	356	Elements	365
Usage	356	Return value	366
Language cross-reference	356	Usage	366
Examples: InputBox function	356	Examples: IsElement function	366
InputBP function	356	IsEmpty function	367
Syntax.	357	Syntax.	367
Return value	357	Elements	367
Usage	357	Return value	367
Examples: InputBP function	357	Examples: IsEmpty function	367
InStr function	357	IsList function	367
Syntax.	357	Syntax.	367
Elements	357	Elements	367
Return value	358	Return value	367
Usage	358	Examples: IsList function	368
Language cross-reference	358	IsNull function	368
Examples: InStr function	359	Syntax.	368
InStrB function	359	Elements	368
Syntax.	359	Return value	368
Elements	359	Usage	368
Return value	359	Language cross-reference	368
Usage	360	Examples: IsNull function	368
Examples: InStrB function	360	IsNumeric function	368
InStrBP function	360	Syntax.	368
Syntax.	360	Elements	369
Elements	360	Return value	369
Return value	360	Usage	369
Usage	361	Language cross-reference	369
Examples: InStrBP function.	361	Examples: IsNumeric function.	369
InStrC function.	361	IsObject function	370
Syntax.	361	Syntax.	370
Elements	361	Elements	370
Return value	361	Return value	370
Usage	361	Examples: IsObject function	370
Int function	362	IsScalar function	370
Syntax.	362	Syntax.	370
Elements	362	Elements	370
Return value	362	Return value	370
Usage	362	Examples: IsScalar function.	371
Examples: Int function	362	IsUnknown function	371
Integer data type	362	Syntax.	371

Elements	371	Usage	379
Return value	371	LenC function	379
Examples: IsUnknown function	371	Syntax	380
Join function	372	Elements	380
Syntax	372	Return value	380
Elements	372	Usage	380
Return value	372	Let statement	380
Usage	372	Syntax	380
Error handling	372	Elements	380
Language cross-reference	372	Usage	381
Examples: Join function	373	Example: Let statement	381
Kill statement	373	Line Input # statement	381
Syntax	373	Syntax	381
Elements	373	Elements	382
Usage	373	Usage	382
Example: Kill statement	373	Example: Line Input # statement	382
LBound function	373	ListTag function	382
Syntax	373	Syntax	382
Elements	373	Elements	382
Return value	373	Return value	382
Usage	373	Usage	382
Example: LBound function	374	Example: ListTag function	383
LCase function	374	LOC function	383
Syntax	374	Syntax	383
Elements	374	Elements	383
Return value	374	Return value	383
Usage	374	Example: LOC function	383
Example: LCase function	374	Lock and Unlock statements	384
Left function	374	Syntax	384
Syntax	374	Elements	384
Elements	374	Usage	384
Return value	374	Example: Lock and unlock statements	384
Example: Left function	375	LOF function	385
LeftB function	375	Syntax	385
LeftBP function	375	Elements	385
Syntax	375	Return value	385
Elements	375	Usage	385
Return value	375	Example: LOF function	385
Example: LeftBP function	375	Log function	386
LeftC function	376	Syntax	386
Syntax	376	Elements	386
Elements	376	Return value	386
Return value	376	Usage	386
Usage	376	Example: Log function	386
Example: LeftC function	376	Long data type	386
Len function	376	Usage	386
Syntax	376	Example: Long data type	386
Elements	376	LSet statement	387
Return value	377	Syntax	387
Usage	377	Elements	387
Examples: Len function	377	Usage	387
LenB function	377	Example: LSet statement	387
Syntax	378	LTrim function	387
Elements	378	Syntax	387
Return value	378	Elements	388
Usage	378	Return value	388
Examples: LenB function	378	Example: LTrim function	388
LenBP function	379	MessageBox function and statement	388
Syntax	379	Function Syntax	388
Elements	379	Statement Syntax	388
Return value	379	Elements	388

Return value	389	On Error statement	398
Usage	390	Syntax	398
Examples: MessageBox function and statement	390	Elements	398
Mid function	391	Usage	398
Syntax	391	How does On Error work?	399
Elements	391	How does the error-handling routine work?	399
Return value	391	Where are error numbers and messages	
Language cross-reference	391	defined?	399
Examples: Mid function	391	Language cross-reference	399
Mid statement	391	Examples: On Error statement	399
Syntax	391	On Event statement	400
Elements	391	Syntax	400
Usage	392	Elements	400
Language cross-reference	392	Usage	401
Examples: Mid statement	392	Examples: On Event statement	401
MidB function	392	On...GoSub statement	401
MidB statement	392	Syntax	401
MidBP function.	392	Elements	401
Syntax.	393	Usage	402
Elements	393	Examples: On...GoSub statement	402
Return value	393	On...GoTo statement	402
Examples: MidBP function	393	Syntax	402
MidC function	393	Elements	402
Syntax.	393	Usage	403
Elements	393	Examples: On...GoTo statement	403
Return value	394	Open statement	403
Usage	394	Syntax	404
Minute function	394	Elements	404
Syntax	394	Usage	406
Elements	394	Examples: Open statement.	406
Return value	394	Option Base statement	407
Language cross-reference	394	Syntax	407
Examples: Minute function	394	Elements	407
MkDir statement	395	Usage	407
Syntax	395	Examples: Option Base statement	407
Elements	395	Option Compare statement	407
Usage	395	Syntax	407
Examples: MkDir statement	395	Elements	408
Month function.	395	Usage	408
Syntax	395	Examples: Option Compare statement.	408
Elements	395	Option Declare statement	409
Return value	396	Syntax	409
Language cross-reference	396	Usage	409
Examples: Month function.	396	Examples: Option Declare statement	410
Name statement	396	Option Public statement	410
Syntax	396	Syntax	410
Elements	396	Usage	410
Usage	396	Examples: Option Public statement.	410
Examples: Name statement	397	Print statement	410
Now function	397	Elements	410
Syntax	397	Usage	411
Return value	397	Examples: Print statement	411
Usage	397	Print # statement	412
Language cross-reference	397	Syntax	412
Examples: Now function	397	Elements	412
Oct function.	397	Usage	412
Syntax	397	Examples: Print # statement	413
Elements	397	Property Get/Set statements	413
Return value	397	Syntax	413
Usage	398	Elements	414
Examples: Oct function.	398	Usage	415

Using Property Get	415	RightBP function	428
Using Property Set	415	Syntax	428
Referencing a property that returns an array, list, or collection	416	Elements	428
Passing a property to a function	416	Return value	429
Examples: Property Get/Set statements	416	Language cross-reference	429
Put statement	416	Examples: RightBP function	429
Syntax	417	RightC function	429
Elements	417	Syntax	429
Usage	417	Elements	429
Examples: Put statement	418	Return value	429
Randomize statement.	419	Usage	429
Syntax	419	Examples: RightC function	429
Elements	419	Rmdir statement	430
Usage	419	Syntax	430
Examples: Randomize statement.	419	Elements	430
ReDim statement	419	Usage	430
Syntax	420	Examples: Rmdir statement	430
Elements	420	Rnd function	430
Usage	420	Syntax	430
Examples: ReDim statement	421	Elements	430
Rem statement	421	Return value	430
Syntax	421	Usage	430
Elements	422	Language cross-reference	431
Usage	422	Examples: Rnd function	431
Language cross-reference	422	Round function.	431
Examples: Rem statement	422	Syntax	431
%Rem directive.	422	Elements	431
Syntax	422	Return value	431
Elements	422	Language cross-reference	432
Usage	422	Examples: Round function.	432
Language cross-reference	423	RSet statement	432
Examples: %Rem directive.	423	Syntax	432
Replace function	423	Elements	432
Syntax.	423	Usage	432
Elements	423	Examples: RSet statement	432
Return value	424	RTrim function	433
Usage	424	Syntax	433
Language cross-reference	425	Elements	433
Examples: Replace function.	425	Return value	433
Reset statement.	425	Examples: RTrim function	433
Syntax	425	Run statement	433
Usage	425	Second function	433
Examples: Reset statement.	425	Syntax	433
Resume statement.	426	Elements	433
Syntax	426	Return value	434
Elements	426	Language cross-reference	434
Usage	426	Examples: Second function	434
Examples: Resume statement	426	Seek function	434
Return statement	427	Syntax	434
Syntax.	427	Elements	434
Usage	427	Return value	434
Examples: Return statement	427	Usage	435
Right function	427	Examples: Seek function	435
Syntax	427	Seek statement	435
Elements	428	Syntax	435
Return value	428	Elements	435
Usage	428	Usage	436
Language cross-reference	428	Examples: Seek statement	436
Examples: Right function	428	Select Case statement.	436
RightB function.	428	Syntax	436
		Elements	437

Usage	437	Examples: Select Case statement	437	Syntax	448
SendKeys statement	438	Elements	448	Usage	449
Syntax	438	Examples: Spc function	449	Split function	449
Usage	438	Syntax	449	Elements	449
Examples: SendKeys statement	440	Return value	450	Usage	450
Set statement	440	Error Handling:	450	Language cross-reference	450
Syntax 1: Create an object and assign a reference	440	Examples: Split function	451	Sqr function	451
Elements	440	Syntax	451	Elements	451
Syntax 2: Copy an existing object reference to another variable	440	Return value	451	Language cross-reference	451
Elements	441	Examples: Sqr function	451	Stop statement	451
Syntax 3: Associate a product object with a variable	441	Syntax	451	Syntax	451
Elements	441	Elements	451	Usage	451
Usage	441	Language cross-reference	451	Str function	452
Language cross-reference	442	Examples: Set statement	442	Syntax	452
Examples: Set statement	442	SetFileAttr statement	442	Elements	452
SetFileAttr statement	442	Syntax	442	Return value	452
Syntax	442	Elements	442	Usage	452
Elements	442	Language cross-reference	442	Language cross-reference	452
Usage	443	Examples: SetFileAttr statement	443	Examples: Str function	452
Examples: SetFileAttr statement	443	Sgn function.	443	StrCompare function	452
Sgn function.	443	Syntax	443	Syntax	453
Syntax	443	Elements	443	Elements	453
Elements	443	Return value	444	Return value	453
Return value	444	Language cross-reference	444	Language cross-reference	453
Language cross-reference	444	Examples: Sgn function	444	Examples: StrCompare function	453
Examples: Sgn function	444	Shell function	444	StrConv function	454
Shell function	444	Syntax	444	Syntax	454
Syntax	444	Elements	444	Elements	454
Elements	444	Return value	445	Return value	454
Return value	445	Usage	445	Language cross-reference	455
Usage	445	Language cross-reference	445	Examples: StrConv function	455
Language cross-reference	445	Examples: Shell function	445	StrLeft function.	455
Examples: Shell function	445	Shellid function	445	Syntax	455
Shellid function	445	Syntax	445	Elements	455
Syntax	445	Elements	445	Return value	456
Elements	445	Return value	446	Language cross-reference	456
Return value	446	Usage	446	Examples: StrLeftBack function	456
Usage	446	Examples: Shellid function	446	Syntax	456
Examples: Shellid function	446	Sin function	446	Elements	456
Sin function	446	Syntax	446	Language cross-reference	457
Syntax	446	Elements	447	StrRight function	457
Elements	447	Return value	447	Syntax	457
Return value	447	Language cross-reference	447	Elements	457
Language cross-reference	447	Examples: Sin function	447	Language cross-reference	457
Examples: Sin function	447	Single data type	447	StrRightBack function	457
Single data type	447	Usage	447	Syntax	457
Usage	447	Examples: Single data type	447	Elements	458
Examples: Single data type	447	Sleep statement.	447	Language cross-reference	458
Sleep statement.	447	Syntax	447	StrToken function	458
Syntax	447	Elements	447	Syntax	458
Elements	447	Usage	448	Space function	448
Usage	448	Examples: Sleep statement	448	Syntax	448
Examples: Sleep statement	448	Space function	448	Elements	448
Space function	448	Syntax	448	Language cross-reference	448
Syntax	448	Elements	448	Return value	448
Elements	448	Return value	448	Examples: Space function	448
Return value	448	Examples: Space function	448	Spc function.	448

Elements	458	Examples: TimeNumber function	471
Return value	459	Timer function	471
Usage	459	Syntax	471
Error Handling	459	Return value	471
Language cross-reference	459	Usage	471
Examples: StrToken function	459	Examples: Timer function	471
String data type	460	TimeValue function	471
Usage	460	Syntax	471
Examples: String data type	460	Elements	471
String function	460	Return value	471
Syntax	460	Usage	472
Elements	461	Language cross-reference	472
Return value	461	Examples: TimeValue function	472
Examples: String function	461	Today function	472
Sub statement	461	Syntax	472
Syntax	461	Return value	472
Elements	461	Usage	472
Usage	462	Language cross-reference	472
Examples: Sub statement	462	Examples: Today function	472
Sub Delete	463	Trim function	472
Syntax	464	Syntax	472
Usage	464	Elements	473
Examples: Sub Delete	464	Return value	473
Sub Initialize	464	Language cross-reference	473
Syntax	465	Examples: Trim function	473
Usage	465	Type statement	473
Examples: Sub Initialize	465	Syntax	473
Sub New	465	Elements	473
Syntax	465	Usage	474
Elements	465	Defining types	474
Usage	466	Declaring type members	474
Examples: Sub New	466	Declaring a type variable	474
Sub Terminate	467	Referring to type members	474
Syntax	467	Examples: Type statement	475
Usage	467	TypeName function	475
Examples: Sub Terminate	467	Syntax	476
Tab function	468	Elements	476
Syntax	468	Return value	476
Elements	468	Language cross-reference	476
Usage	468	Examples: TypeName function	477
Language cross-reference	468	UBound function	477
Examples: Tab function	468	Syntax	477
Tan function	469	Elements	477
Syntax	469	Return value	477
Elements	469	Usage	478
Return value	469	Examples: UBound function	478
Language cross-reference	469	UCase function	478
Examples: Tan function	469	Syntax	478
Time function	469	Elements	478
Syntax	469	Return value	478
Return value	469	Usage	478
Usage	469	Language cross-reference	478
Examples: Time function	470	Examples: UCase function	478
Time statement	470	UChr function	478
Elements	470	Syntax	479
Examples: Time statement	470	Elements	479
TimeNumber function	470	Return value	479
Syntax	470	Examples: UChr function	479
Elements	470	Uni function	479
Return value	470	Syntax	479
Usage	470	Elements	479

Return value	479
Usage	479
Examples: Uni function	479
Unlock statement	479
Use statement	480
Syntax	480
Elements	480
Usage	480
Loading a used module	480
Referring to Public names in a used module	480
Declaring Public names	480
Examples: Use statement	480
UseLSX statement	480
Syntax	480
Elements	481
Usage	481
Examples: UseLSX statement	481
UString function	482
Syntax	482
Elements	482
Return value	482
Usage	482
Language cross-reference	482
Examples: UString function	482
Val function	482
Syntax	482
Elements	482
Return value	483
Usage	483
Language cross-reference	483
Examples: Val function	483
Variant data type	483
Usage	483
Examples: Variant data type	484
Weekday function	485
Syntax	485
Elements	485
Return value	485
Usage	485
Language cross-reference	485
Examples: Weekday function	485
While statement	485
Syntax	485
Elements	486
Usage	486
Language cross-reference	486
Examples: While statement	486
Width # statement	486
Syntax	486
Elements	486
Usage	487
Examples: Width # statement	487
With statement	487
Syntax	487
Elements	487
Usage	488
Examples: With statement	488
Write # statement	488
Syntax	488
Elements	488
Usage	489

Examples: Write # statement	489
Year function	490
Syntax	490
Elements	490
Return value	490
Language cross-reference	490
Examples: Year function	491
Yield function and statement	491
Syntax	491
Return value	491
Usage	491
Examples: Yield function and statement	491

Appendix A Language and Script Limits 493

Limits on numeric data representation in LotusScript	493
Limits on string data representation in LotusScript	493
Limits on array variables in LotusScript	494
Limits on file operations in LotusScript	494
Limits in miscellaneous source language statements in LotusScript	494
Limits on compiler and compiled program structure in LotusScript	495
Storage size of data	495

Appendix B Platform Differences 497

OS/2 platform differences in LotusScript	497
Language construct differences	497
File system differences	497
Other differences	497
UNIX platform differences in LotusScript	498
Language construct differences	498
File system differences	499
Other differences	499
Macintosh platform differences in LotusScript	500
Language construct differences	500
File system differences	500
Other differences	501
OS/400 platform differences in LotusScript	501
Language construct differences	501
File system differences	502
Other differences	502

Appendix C LotusScript/REXX Integration 505

Appendix D LotusScript Aliases 507

Appendix E MIME Charset Names 509

Appendix F Compile-time Error Messages 511

DELETE not valid on: <name>	511
Too many nested INCLUDEs	511

File contains too many source lines	511	Arguments not legal in declaration of: <sub name>	520
Illegal OPTION BASE after array declaration . . .	511	Undefined label: <label name>	520
Illegal OPTION DECLARE after implicit declaration	511	Illegal data type for argument: <argument name>	521
Too many items specified in input/output statement.	511	Too many arguments for: <subprogram name> . . .	521
Illegal value for OPTION BASE	512	Cannot subclass: <class name>	521
Too many labels specified in ON...GOTO statement	512	Derived class may not be PUBLIC when parent is PRIVATE: <class name>	521
SUB NEW arguments do not match parent's SUB NEW arguments	512	Illegal use of NEW or DELETE	521
Name previously declared: <name>	512	DIM required on declarations in this scope . . .	521
Class is not a parent of this class: <class name>	513	Illegal PRIVATE declaration of: <name>	521
Illegal name for class or type: <name>	513	Illegal PUBLIC declaration of: <name>	522
Public symbol is declared in another module: <name>	513	Name was forward declared as something else: <name>	522
Member is not a subprogram: <member name> . . .	514	Duplicate forward declaration: <name>	522
Illegal executable code at the module level . . .	514	Storage class or visibility does not match forward declaration: <subprogram name>	522
Illegal PUBLIC instance of PRIVATE class or type: <instance name>.	514	Return type does not match forward declaration: <function name>	522
Illegal type suffix on name: <name>	514	Number of arguments does not match forward declaration: <subprogram name>	523
ISELEMENT argument is not a list or variant: <name>	514	Argument does not match forward declaration: <argument name>	523
Illegal scope for PUBLIC or PRIVATE on: <name>	514	Illegal function return type for: <function name>	523
Illegal constructor clause on: <sub name> . . .	515	Method was declared as something else in a parent: <method name>	524
Parent SUB NEW has arguments, SUB NEW is required for: <class name>	515	Method signature does not match parent method: <method name>	524
Illegal USE or UseLSX statement after declaration	515	PROPERTY GET and SET must have same storage class and visibility.	524
Member declared in a parent class	515	Illegal property type for: <property name>	525
Event handler must be a LotusScript SUB or FUNCTION: <handler name>	516	PROPERTY GET and SET must have same data type	525
Member of PUBLIC class or type is instance of a PRIVATE class or type: <member name>.	516	Property was declared as something else in a parent: <property name>	525
FORALL alias variable was previously declared: <name>	516	Property type does not match parent property: <property name>	525
FORALL alias variable already in use: <variable name>.	516	Illegal pass by value: <argument name>	526
CASE ELSE must be the last CASE in a SELECT statement.	517	Illegal STATIC on: <name>	526
TYPE declaration has no members	517	Illegal external argument: <argument name> . . .	526
Declaration of external subprogram is not legal inside a class	517	Illegal construction of type instance: <instance name>	526
Illegal use of array or list element as FORALL target	517	Class or type name not found: <name>	527
Illegal use of property: <property name>.	517	Illegal range specifier.	527
Wrong data type for argument <argument name> in event handler <event handler name>	518	Illegal DEFtype statement after declaration. . . .	527
Maximum array dimensions (8) exceeded: <array name>.	518	Duplicate range specifier	528
Illegal array bound for: <array name>	518	Label is illegal outside of a subprogram	528
Array size exceeds maximum: <array name> . . .	518	Error number must be INTEGER constant: <name>	528
Illegal specification of array bounds for: <array name>.	518	Error number must be INTEGER	528
Declaration not valid in TYPE scope: <name>. . .	518	Illegal ON ERROR statement	528
Statement is illegal in TYPE block: <keyword> . .	519	Statement is illegal outside of a subprogram . . .	528
Statement is illegal in CLASS block: <keyword> . .	519	Not a product class: <name>	529
TYPE may not have instance of itself as a member: <instance name>.	519	Not a product class instance: <name>	529
Out of memory.	519	Not an event name: <name>	529
Size of data cannot exceed 64K in this scope . . .	520	Not a sub or function name: <name>	529
Size of data cannot exceed 32K in this scope . . .	520	Illegal REDIM on: <name>	529
Illegal constant expression for: <CONST name> . .	520	Illegal RESUME statement	530
		FOR count variable already in use: <name>	530
		FORALL alias variable is not of same data type: <name>	530
		FOR count variable must be a scalar variable: <name>	530

Illegal type suffix on FORALL alias variable: <name>	530	Illegal EXIT <EXIT type>	543
Not a PUBLIC member: <name>	531	Illegal OPTION PUBLIC after declaration	543
Illegal reference to FORALL alias variable: <name>	531	Illegal use of ERASE	543
Type suffix does not match data type: <name>	531	SET may only be used on class instance assignments	543
Not a member: <name>	531	Illegal pass by value	543
Variable not declared: <name>	531	Wrong number of arguments to constructor for class: <class name>	544
Illegal single-line IF	531	Illegal reference to array or list: <array or list name>	544
Name does not match FOR count variable: <name>	532	Illegal type suffix on keyword: <keyword>	545
Not an array, list, collection or variant: <name>	532	Compiler statement stack overflow at: <token name>	545
ME not valid outside of class scope	532	Maximum allowable code size exceeded	545
.. not valid outside of class scope	532	Maximum allowable data size exceeded	545
Reference must contain exactly one subscript: <name>	532	Maximum allowable symbol table size exceeded	545
Illegal parenthesized reference: <name>	533	PUBLIC is not allowed in this module	545
Wrong number of array subscripts for: <array name>	533	Illegal call to: <sub name>	546
Not an instance name: <name>	533	Empty parentheses not legal on: <name>	546
Bounds must be specified in REDIM of: <array name>	533	Illegal use of parentheses	546
Variable required: <name>	533	Class not specified on BIND into: <name>	546
Named product class instance not valid here	534	Illegal Directive	547
Illegal reference to: <name>	534	Unterminated %IF, %ELSEIF, or %ELSE directive	547
Numeric overflow	534	Illegal character after directive	547
Numeric underflow	535	LIB name must be a string constant	547
Illegal numeric constant	535	USE or USELSX name must be a string constant	547
Illegal product constant: <name>	535	EVALUATE argument must be a string constant	548
Name too long: <name>	535	Illegal second parenthesized expression	548
Token is too long	535	Statement is illegal in a subprogram	548
Declaration may not contain type suffix and data type: <name>	535	Illegal use of UNICODE or LMBCS keyword	548
Illegal string length constant for: <name>	536	UNICODE and LMBCS strings must be declared BYVAL	549
Illegal use of NEW on array or list declaration: <name>	536	Too many nested WITHs	549
INCLUDE filename must be a string constant	536	Illegal use of escape character in identifier: <name>	549
Cannot open included file: <file name>	536	Illegal use of escape character	549
Unterminated %REM block	536	Error in EVALUATE macro	549
Unterminated string constant	536	Name previously referenced in this scope	549
Unterminated multiline string	537	Wrong number of arguments for event handler: <sub name>	550
Unterminated square bracket reference	537	Property is read-only: <property name>	550
Illegal character after continuation character	537	Missing array subscript or collection index for: <name>	550
Illegal character after %INCLUDE directive	537	Missing argument to constructor for: <class name>	550
SET required on class instance assignment	537	Missing array bound for: <array name>	551
Unterminated <keyword> block	538	LEN argument must be a variable or string expression	551
Unexpected: <token>; Expected: <token>	538	Missing collection index for: <name>	551
Parser stack overflow at: <token name>	539	Missing list subscript for ISELEMENT argument: <list name>	551
Unknown statement	539	Cannot assign into collection item	552
Maximum number of errors reached	539	Cannot forward declare CLASS or TYPE	552
PROPERTY SET not defined for: <property name>	539	CLASS or TYPE declaration may not be inside a control block	552
PROPERTY GET not defined for: <property name>	539	Procedure declaration may not be inside a control block	552
Duplicate option	539	Product class does not have a New method: <class name>	552
Missing argument for: <function name>	540	Collection item is not an instance	552
Expected expression before end of argument list for: <function name>	540	Illegal on declarations in this scope: <keyword>	552
Wrong number of arguments for: <name>	540	Wrong return type in event handler <handler name>	553
LISTTAG argument is not a FORALL alias variable	540	Event handler must be a FUNCTION	553
Type mismatch on: <name>	540		
Illegal BYVAL on arguments to: <subprogram name>	542		
Illegal TO in reference to: <name>	542		
Illegal BYVAL	542		
Duplicate label: <label name>	542		

Event handler must be a SUB	553
Conflicting option	553
PROPERTY GET and SET arguments do not match: <property_name>	553
Number of arguments do not match for PROPERTY GET and SET <property_name>	553
Property signature does not match parent property: <property_name>	553
Type suffix character required on: <name>	554
Must be a sub: <procedure_name>	554

Appendix G Run-time

Error Messages 555

User-defined error	555
RETURN without GOSUB	555
Illegal function call	555
Overflow	556
Invalid ^ operator operands	556
Out of memory	556
Subscript out of range	556
Expression out of range	557
Duplicate PUBLIC name in USE module: <module name>	557
Division by zero	557
Type mismatch	557
Out of string space	557
No RESUME	558
RESUME without error	558
Out of stack space	558
Sub or function not defined	558
Error in loading DLL	558
Bad DLL calling convention	558
Internal error	559
Bad file name or number	559
File not found	559
Bad file mode	559
File already open	559
Device I/O error	559
File already exists	560
Bad record length	560
Disk full	560
Input past end of file	560
Bad record number	560
Bad file name	560
Too many files	560
Device unavailable	561
Permission denied	561
Disk not ready	561
Cannot rename with different drive	561
Path/file access error	561
Path not found	561
Object variable not set	561
FOR loop not initialized	562
Invalid pattern string	562
Invalid use of NULL	562
Cannot destroy active instance	562
File not writable	562
File not readable	563
Illegal file number	563
File not open	563
Conflicting modes supplied	563

Unable to open file	563
Illegal operation for file mode	563
Data too big for record	563
Bad attribute	564
Cannot set attribute for file	564
List item does not exist	564
Cannot find module <module_name>	564
Cannot find external name <name>	564
Type mismatch on external name <name>	564
Module already loaded	564
Invalid module file	565
Compiler error	565
Opcode <opcode_name> not implemented	565
Named product object does not exist	565
ADT error: Control procedure missing	565
Bad argument to external function	565
Unsupported argument type to external function	565
Unsupported return type for external function	565
External function not found	566
Event handler not attached	566
Module in use	566
Illegal circular USE: <module_name>	566
Too many calls into module	566
LISTTAG argument not a list element	566
Illegal REDIM of fixed array	567
Array size exceeds maximum limit	567
Illegal LIKE pattern	567
Error in constant expression evaluation	567
Operation not supported on this platform	568
Type suffix does not match actual data type	568
Instance member does not exist	568
Variant does not contain an object	568
Variant does not contain a container	568
Wrong number of arguments for method	568
Name used as a method is not a method	569
Illegal use of sub	569
Illegal use of function	569
Illegal use of property	570
Illegal use of read-only property	570
List reference must contain exactly one subscript	570
Illegal DELETE	570
Not a product object	571
Event does not exist	571
Event handler argument count mismatch	571
Event handler argument type mismatch	571
Not a PUBLIC member	572
Missing argument	572
Operation is disallowed in this session	572
Attempt to access an uninitialized dynamic array	572
Error loading USE or USELSX module	573
Wrong number of collection indices	573
Not a collection object	573
Collection item not found	573
Underflow	573
SET required on class instance assignment	573
Invalid Collection item	574
Automation-Object error	574
Automation-Object cannot create	574
Automation-Object file name error	574
Automation-Object member not found	574
Automation-Object argument count	574

Automation-Object argument type mismatch . . .	575
ForAll container invalid or modified	575
Out of system stack space	575
Illegal REDIM	575
Error creating product object	576
Error accessing product object property	576
Error accessing product object method	576
Error accessing product object	576
Error in EVALUATE macro	576
Event handler return type mismatch.	576
Event handler procedure type mismatch	576
Wrong number of arguments for PROPERTY.	576

Illegal use of MEMBER	576
PROPERTY SET not defined	577
PROPERTY GET not defined	577
String too large.	577
Variable is read-only	577
Unknown class instance.	577
Cannot assign into collection item	577
Wrong number of array subscripts	577

Index 579