

# TABLE OF CONTENTS

<b>Splitting Your App Into DLLs .....</b>	<b>1</b>
The Objective	1
The Global DLL	3
The Program DLLs	5
The Main EXE	7
Summary	7
<b>Using Procedure Category to Split Apps into DLLs.....</b>	<b>9</b>
Getting exports right	13
The downside	15
Exporting procedures from an EXE	16
Using modules for organization	16
Other tips	17
Summary	17
<b>Who Calls Who - Keeping Track Of DLL Calling Order .....</b>	<b>19</b>
Rule Number One: DLL calls:	22
Rule Number Two: Moving the DLL compile order up:	23
Rule Number Three: Moving the DLL compile order down:	23
<b>Optimizing DLL Loading - Introduction to Rebasing .....</b>	<b>25</b>
Rebasing	26
A short history of rebasing	27
Please step out to my garage for an example	28
The Process and Penalties of Rebasing	30
For Further Information	32
<b>Optimizing DLL Loading - Rebasing Your DLLs.....</b>	<b>33</b>
How to Set the Image Base in Clarion	33
Determining What Address to Use	34
The Debug Advantage	38
Thirty-second time a template	40

Implementing rebasing step by step	41
Implementing rebasing gets easier	42
Summary	44
Source code	44
Resources	45
<b>Rebasing Third Party DLLs .....</b>	<b>47</b>
Rebase.exe Syntax	48
Using the CarlBase utility	49
Should you rebase with Rebase.exe?	50
A word of caution	51
A quick way to rebase your entire multi-DLL project	51
Creating your own rebase utility	52
Success!	52
Resources	53
<b>Calling By Address, STARTing By Address .....</b>	<b>55</b>
How the class works	57
STARTing DLL procedures	61
STARTing a Procedure	62
Summary	66
Source code	66
<b>Loading DLLs At Runtime - The Theory .....</b>	<b>67</b>
Libraries, Libraries, and more Libraries	68
Run-time Dynamic Linking	69
Function pointers	69
Need proof? Symbols, assembler mnemonics, and hexadecimal, oh my!	71
Load Library Class	72
<b>Loading DLLs At Runtime - Calling DLL Procedures .....</b>	<b>75</b>
Load the DLL	76
Assign the address	78
Assign the address	80
Call the procedure	82
Unloading the DLL	82
Source code	83

<b>Loading DLLs At Runtime - Calling Functions By Address.....</b>	<b>85</b>
Implementing the load library class	86
Two functions, different methods, same result	86
Source code	95

