

The Smartest Way To Get Certified™



The perfect supplement
to all certification
study guides



Visual C++ 6 Distributed

Exam 70-015

**Microsoft
Certified
Solution
Developer**

James M. Lacey



中国水利水电出版社
www.waterpub.com.cn



MCSD Visual C++ 6 Distributed

James M. Lacey

Reprint Copyright

Original English language edition published by The Coriolis Group LLC, 14455N. Hayden Drive, Suite 220, Scottsdale, Arizona 85260 USA, telephone (602)483-0192, fax (602) 483-0193. Copyright © 2000 by The Coriolis Group. English language reprinted copyright © 2001 by China WaterPower Press. All rights reserved.

Original Copyright

©2000 The Coriolis Group. All Rights Reserved.

This book may not be duplicated in any way without the express written consent of the publisher, except in the form of brief excerpts or quotations for the purposes of review. The information contained herein is for the personal use of the reader and may not be incorporated in any commercial programs, other books, databases, or any kind of software without written consent of the publisher. Making copies of this book or any portion for any purpose other than your own is a violation of United States copyright laws.

Limits Of Liability And Disclaimer Of Warranty

The author and publisher of this book have used their best efforts in preparing the book and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book.

The author and publisher shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims of productivity gains.

Trademarks

Trademarked names appear throughout this book. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, the publisher states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

北京市版权局著作权登记号: 01-2000-1750

图书在版编目 (CIP) 数据

MCSD Visual C++ 6 Distributed 考前冲刺/ (美) 莱西 (Lacey, J.M.) 著.

—北京: 中国水利水电出版社, 2001.5

MCSD Visual C++ 6 Distributed Exam Cram

ISBN 7-5084-0537-4

I. M... II. 莱... III. C语言—程序设计—资格考核—自学参考资料
—英文②计算机通信网—交换器—工程技术人员—资格考核—自学参
考资料—英文 IV. TP312

中国版本图书馆CIP数据核字 (2000) 第82210号

书 名	MCSD Visual C++ 6 Distributed考前冲刺
作 者	[美] James M. Lacey
出版、发行	中国水利水电出版社 (北京市三里河路6号100044) 网址: www.waterpub.com.cn E-mail: mchannel@public3.bta.net.cn (万水) sale@waterpub.com.cn
销 售	电话: (010) 68359286 (万水)、63202266 (总机)、68331835 (发行部) 全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京蓝空印刷厂
规 格	787×1092毫米 16开本 30.25印张 692千字
版 次	2001年5月第一版 2001年5月北京第一次印刷
印 数	0001—2000册
定 价	48.00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社发行部负责调换
版权所有·侵权必究

Our readers tell us why **Certification Insider Press is #1**

I just wanted to thank you for helping me become an MCSE. I've self-studied for all six exams using your *Exam Cram* books. I use your books to narrow down all the information into "hot topics" that almost always show up on the tests. The price is right and the layout is great. Keep up the good work!

—Eric Stanfield, *MCSE*

I was drowning in the sea of lengthy, wordy books. I am impressed with the *Exam Cram* books. They are concise and show you exactly where to study further.

—Miriam Stryzik

Thanks for the helpful Cram Sheet in your books. I am going to purchase ALL of the *Exam Crams* while I am working toward my MCSE.

—Robert Vierthaler, *PC Support Analyst*

I just wanted to say thanks to your company; I have used your *Exam Cram* books for every certification. I am now concentrating on Novell. Thanks!

—Steve Guillot, *A+, N+, MCP+I, MCSE, CCNA*

So far, your informative, no-nonsense books have been an invaluable source of information on the course toward my MCSE. Keep up the good work and thanks for the help!

—Ian Clark

This is by far the BEST study material for MCSE preparation. I have used every type of tool out there; however, I passed my last two tests by studying ONLY the *Exam Crams*. That is awesome. Thanks.

—Bruce Durbin

See for yourself why we're #1 and drop us a line at cipq@coriolis.com—we look forward to hearing about your testing successes!



Visual C++ 6 Distributed The Cram Sheet

This Cram Sheet contains the distilled, key facts about the Visual C++ 6 Distributed exam. Review this information last thing before entering the test room, paying special attention to those areas where you feel you need the most review. You can transfer any of the facts onto a blank piece of paper before beginning the exam.

DEVELOPMENT ENVIRONMENT

1. Compiler switches control the stack construction for function call definitions:
 - **/Gd**—Use **__cdecl**, which specifies the C calling convention (the default setting).
 - **/Gr**—Use **__fastcall**, which specifies that arguments are passed in registers.
 - **/Gz**—Use **__stdcall**, which specifies that arguments are pushed on the stack from right to left.
2. Compiler switch **/GZ** can help catch release build errors in debug.
3. The SourceSafe **Share** command allows a single instance of a component to be used with multiple projects.
4. The SourceSafe **Branch** command allows tracking the individual directions of a file as a project. The Branch command breaks any sharing links.
5. Pinning a SourceSafe file prevents you from making changes to the file.
6. Use **Share**, **Pin**, and **Branch** to create service pack projects (bug fixes).

MFC ARCHITECTURE—DOCUMENT/VIEW

7. The AppWizard-generated **InitInstance** function in a **CWinApp**-derived class initiates the creation of the main frame window, the document, and the

view. The document template ties together these three objects. Each combination of frame window, document, and view types requires a unique document template. Each document template creates and manages all the documents of its type.

8. The document object stores the applications data. **CDocument**, derived from **CCmdTarget**, is the base class for document objects.
9. The view object displays a visible representation of the data. **CView**, derived from **CWnd**, is the base class for view objects.
10. MFC supports two types of threads: user interface (UI) threads, which have a message pump, and worker threads, which don't. Use **AfxBeginThread()** to launch a new thread. **CWinThread** objects encapsulate threads of execution. **CWinThread** is derived from **CCmdTarget**.

MFC ARCHITECTURE—DYNAMIC LINK LIBRARIES

11. An MFC regular DLL exports C-style functions and can be used by Visual Basic, Delphi, and other languages that can call C-style functions.
12. MFC extension DLLs are used to export functions and classes that enhance MFC. Use the **AFX_EXT_CLASS** macro to export a class or function.
13. The preprocessor symbol **_AFXEXT** must be defined for extension DLLs.

MFC ARCHITECTURE—USER INTERFACE

14. The **CMainFrame::OnCreate** function, generated by the AppWizard, creates and docks the standard status bar and toolbar. Here, you will normally add toolbars and modify the status bar configuration.
15. Menus, keyboard accelerators, toolbar buttons, and other UI objects generate command messages (**WM_COMMAND**).
16. **WM_COMMAND** messages sent to an SDI frame window are routed through the active view object, the active view's document object, the document object's document template, the main frame window object, the application object, and the default window procedure.
17. Regular window messages go directly to the window for which they are intended.
18. Use **CreateEx()** to create toolbar objects when extended styles need to be present during the construction of the embedded control.
19. After creating a toolbar, four more actions are required to place a dockable toolbar in your application: load the toolbar resource, enable docking for the toolbar, enable docking for the frame window, and dock the toolbar to the frame window.
20. Add ToolTips to a toolbar by adding **CBRS_TOOLTIPS** to the toolbar style, creating string table resources containing the ToolTip text, and setting the string resource IDs so that they match the toolbar buttons.
21. Use **CStatusBar::SetIndicators ()** to specify the number of panes a status bar will contain.
22. Use **CStatusBar::SetPaneText()** or **CCmdUI::SetText()** to display text in a status bar pane.
23. Standard controls, such as **CButton** and **CEdit**, send notifications to their parents by using **WM_COMMAND**.
24. Most common controls, such as **CProgressCtrl** and **CSpinButtonCtrl**, enclose their notifications in **WM_NOTIFY**.
25. **CDataExchange** provides dialog data exchange and validation (DDX/DDV).
26. Override **DoDataExchange()** and use **DDX_** functions to transfer data between controls and

data members. Use **DDV_** functions to validate the values entered into controls.

27. Register new ActiveX controls by running **RegSvr32**, passing it the control's file name on the command line.
28. To create an ActiveX control at runtime, insert the component into the project, add an embedded ActiveX control wrapper class member to the appropriate window class, add an ID for the control, add code to call the control's **Create()** function, add the event sink declaration and definition macros, and add the necessary event message handlers and prototypes.

MFC ARCHITECTURE—DRAWING, PRINTING, AND PRINT PREVIEW

29. A **CView**-derived class overrides **OnDraw**. Whenever the framework receives a **WM_PAINT**, it creates a **CPaintDC** object and calls **OnDraw**, passing it a pointer to the **CPaintDC** object.
30. **OnDraw** can be used for displaying to a window, printing, and print previewing.
31. The framework calls five **CView** virtual functions at various stages of the printing process.
 - **OnPreparePrinting**—Override to call **DoPreparePrinting()**, which is responsible for displaying the Print dialog box and creating the printer DC. This function is called at the beginning of the print job.
 - **OnBeginPrinting**—Override to allocate fonts and other resources needed for the print job. You can also set the maximum page count here. This function is called just before printing starts.
 - **OnPrepareDC**—Override to modify the device context. This function is called before each page is printed.
 - **OnPrint**—Override to print headers, footers, and other page elements that are not drawn by **OnDraw** or to do all printing here instead of in **OnDraw**. This function is called to print (or preview) one page.
 - **OnEndPrinting**—Override to deallocate resources allocated in **OnBeginPrinting**. This function is called when printing ends.

COMPONENT OBJECT MODEL (COM)

32. All interfaces must be derived directly or indirectly from **IUnknown**, which has three methods: **QueryInterface()**, **AddRef()**, and **Release()**.

33. Automation interfaces are derived from **IDispatch**. **IDispatch** exposes four methods: **GetTypeInfoCount()**, **GetTypeInfo()**, **GetIDsOfNames()**, and **Invoke()**.
34. Dual interfaces support both vtable binding and the **IDispatch** interface.
35. In-process servers, which are usually housed in a DLL file, run in the same address space as their clients.
36. Local servers are housed in EXE files, run in their own address space, and have their own execution context.
37. Register in-process servers by running **RegSvr32**, passing it the control's file name on the command line. To unregister the server, run **RegSvr32** with the /u switch.
38. Register a local server by running the EXE specifying the /RegServer switch. To unregister the server, use the /UnregServer switch.
39. Local servers publish a component's class factory dynamically at runtime and revoke it at shutdown. It does this by calling the **CoRegisterClassObject()** and **CoRevokeClassObject()** functions.
40. To use the full IDL syntax, define **_WIN_DCOM** or ensure that **_WIN32_WINNT** is set to a value equal to or greater than 0x0400.
41. For a client application to use **CoCreateInstanceEx()**, either **_WIN32_WINNT=0x0400** or **WIN32_DCOM** must be defined and **objbase.h** must be included.
42. The single-thread apartment (STA) requires a Windows message pump.
43. A process can have any number of STAs, but only one multithreaded apartment (MTA).
44. Threads within an MTA can use direct interface pointers, but interface pointers that cross apartment boundaries must be marshaled.
45. All MTA-compatible objects must be thread-safe.
46. An object that supports aggregation provides two **IUnknown** implementations—delegating and nondelegating. When it's being created as an aggregated object, it will cache the outer object's **IUnknown** pointer.
47. **_bstr_t** class encapsulates a **BSTR**, **_variant_t** class encapsulates a **VARIANT**, **_com_error** class encapsulates an **HRESULT** error code, and

_com_ptr_t smart pointer encapsulates an interface pointer.

48. ATL provides two smart pointers to manage interface pointers: **CComPtr** and **CComQIPtr**. ATL **CComBSTR** encapsulates a **BSTR**.

DATA SERVICES

49. The ADO model consists of seven objects (**Connection**, **Command**, **Recordset**, **Error**, **Property**, **Parameter**, and **Field**) and four collections (**Errors**, **Properties**, **Parameters**, and **Fields**).
50. The ADO **Connection** object manages a physical connection to the database. It allows you to open a connection to a data source, execute SQL statements or database stored procedures, and establish transactions for data updates. The **Connection** object contains an **Errors** collection and a **Properties** collection.
51. You can use the ADO **Command** object to execute SQL statements or stored procedures, define and execute repetitive SQL commands that require a set of SQL parameters, and source a **Recordset**. The **Command** object contains a **Parameters** collection and a **Properties** collection.
52. The ADO **Recordset** object is designed to manipulate data held in a data source. It represents a set of records from the data provider and provides the ability to scroll through the records as well as add, edit, and delete them. The **Recordset** object contains a **Fields** collection and a **Properties** collection.
53. Three ways of creating an ADO recordset are **Recordset::Open()**, **Command::Execute()**, and **Connection::Execute()**.
54. ADO provides four types of recordset cursors:
 - *Dynamic*—Unrestricted movement; allows you to add, modify, or delete records
 - *Keyset*—Similar to dynamic; additions and deletions made by other users are not visible.
 - *Forward*—Identical to dynamic, but only scrolls forward.
 - *Static*—Like dynamic, but changes made by other users are not visible.

SQL AND DATABASE

55. The **SELECT** statement is used for retrieving data. The major clauses of a **SELECT** statement must be in a specific order:

SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, COMPUTE [BY].

56. The desired columns are specified right after the **SELECT** keyword.
57. The **FROM** keyword specifies the name of the table or tables from which to retrieve the data.
58. The **WHERE** clause specifies search criteria or filtering conditions:
 - Use **IS NULL** to test for **NULL**. Remember, **NULL** means no value, not 0 or spaces. A **NULL**—when compared in any manner to any value, including another **NULL**—is always false.
 - Use the **IN** operator to test for a set of values.
 - Use the **LIKE** operator when specifying wildcards. The percentage sign (%) means match any number of occurrences of any character. The underscore (_) means match a single character. The bracket ([]) wildcard is used to indicate a set or range of characters.
 - Use **BETWEEN** to find everything within two specific values or string patterns.
59. The **GROUP BY** clause is used to summarize data. It divides data into logical sets so that aggregate calculations can be performed on each group.
60. The **HAVING** clause is used to filter which groups to include and which to exclude. The **HAVING** clause is used in conjunction with the **GROUP BY** clause.
61. The **ORDER BY** clause sorts the results into either ascending or descending order based on a list of column names. An **ORDER BY** clause is nearly always specified when using a **GROUP BY**.
62. The **COMPUTE [BY]** clause is used to create subtotals of data in the output.
63. Aggregate functions operate on a set of rows to calculate and return a single value.
64. Aggregate functions cannot be mixed with nonaggregate columns (columns not in the **GROUP BY** clause) in the **SELECT** list.
65. A join takes two tables and builds a result set from them by multiplying the rows from one table to the rows of the other table.
66. An equi-join combines two or more tables based on the testing of equality between the tables. Equi-joins can be performed by listing two or more tables in the **FROM** clause and specifying the equality operation in the **WHERE** clause. Also known as an inner join.
67. The **AS** key word allows you to specify aliases for table names.
68. An outer join takes two tables and returns the records of one table where there is no matching record in the other table. A **LEFT OUTER JOIN** returns records in the table on the left that don't match any entries in the table on the right. A **RIGHT OUTER JOIN** returns records in the table on the right that don't match any entries in the table on the left. A **FULL OUTER JOIN** is a combination of a **LEFT OUTER JOIN** and a **RIGHT OUTER JOIN**.
69. A subquery is a query that is executed inside the main or outer query.
70. A **VIEW** presents an alternative logical representation of data. **INSERT**, **DELETE**, and **UPDATE** can only be used with views that are defined only for queries on one table. The query can have no **GROUP BY** clause, no **HAVING** clause, no **COMPUTE BY** clause, no aggregate functions, and no **SELECT DISTINCT**. Also, any columns excluded from the **VIEW** must have a default value or be **NULL**-able.
71. A trigger is an event-driven SQL program. Triggers are attached to a specific table for a specific operation: **INSERT**, **DELETE**, or **UPDATE**. When data changes, the trigger fires.
72. Stored procedures can be called by a client program, a trigger, or another stored procedure. They can return as many result sets, of any size, as needed.

To Ann, my loving wife

20

About The Author

James M. Lacey is a software developer and technical trainer specializing in C++, COM, Windows, client/server, and n-tier software development. He has more than 25 years of experience in the computer industry in both hardware and software engineering. This experience includes embedded systems, operating systems, compilers, graphic applications, client/server applications, and database systems.

James is a Microsoft Certified Solution Developer and Trainer. In addition to his own consulting practice, James teaches at Boston University and Worcester Polytechnic Institute. He prefers the challenge of designing and developing total solutions for small- to medium-sized businesses. His consulting practice keeps him abreast of current technology trends and allows him to bring real-world experience to the classroom.

James and his wife, Ann, live in central Massachusetts. He can be contacted at jlacey@rclink.net.

Acknowledgments

I would like to acknowledge and say thank you to all the people who supported me and helped me in writing this book. To all the folks at The Coriolis Group starting with Shari Jo Hehr, Acquisitions Editor: Thank you for bringing me on board and staying with me through the entire project. To Meredith Brittain, Project Editor, who was helpful in providing guidance when assistance was needed and always had a positive attitude that was much appreciated. To Catherine Oliver, Copy Editor, for asking pointed questions and making excellent suggestions that helped clarify my writing. To Nathan Lewis, Technical Reviewer, for his work to ensure technical accuracy by providing suggestions, asking for clarification, and finding outright mistakes in my writing.

I'd also like to thank all the people behind the scenes with whom I never had direct contact but who were instrumental in getting the book to press: Kim Eoff, Production Coordinator; Cynthia Caldwell, Marketing Specialist; Jesse Dunn, for cover design; and April Nielsen, for interior design.

A special thank you to Gene Olafsen and Phil Janus for the chapters they provided.

Finally, to my wife, Ann, for her loving support.

Introduction

Welcome to *MCSD Visual C++ 6 Distributed Exam Cram*! This book aims to help you get ready to take—and pass—the Microsoft certification test numbered 70-015, “Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0.” This Introduction explains Microsoft’s certification programs in general and talks about how the *Exam Cram* series can help you prepare for Microsoft’s certification exams.

Exam Cram books help you understand and appreciate the subjects and materials you need to pass Microsoft certification exams. *Exam Cram* books are aimed strictly at test preparation and review. They do not teach you everything you need to know about a topic (such as the ins and outs of C++ programming and database development or all the nitty-gritty details involved in using the Visual C++ 6 integrated development environment [IDE] and Visual SourceSafe). Instead, *Exam Cram* books present and dissect the questions and problems I’ve found that you’re likely to encounter on a test. I’ve worked from Microsoft’s own training materials, preparation guides, and tests, as well as from a battery of third-party test preparation tools and practice exams. My aim is to bring together as much information as possible about Microsoft certification exams.

Nevertheless, to completely prepare yourself for any Microsoft test, I recommend that you begin by taking the Self-Assessment included in this book immediately following this Introduction. This tool will help you evaluate your knowledge base against the requirements for an MCSD under both ideal and real circumstances.

Based on what you learn from that exercise, you might decide to begin your studies with some classroom training or by reading one of the many study guides available. I also strongly recommend that you install, configure, and work with the software or environment that you’ll be tested on, because nothing beats hands-on experience and familiarity when it comes to understanding the questions you’re likely to encounter on a certification test. Book learning is essential, but hands-on experience is the best teacher of all.

The Microsoft Certified Professional (MCP) Program

The MCP program currently includes the following separate tracks, each of which boasts its own special acronym (as a would-be certificant, you need to have a high tolerance for alphabet soup of all kinds):

- *MCP (Microsoft Certified Professional)*—This is the least prestigious of all the certification tracks from Microsoft. Passing any of the major Microsoft exams (except the Networking Essentials exam) qualifies an individual for the MCP credential. Individuals can demonstrate proficiency with additional Microsoft products by passing additional certification exams.
- *MCP+SB (Microsoft Certified Professional + Site Building)*—This certification program is designed for individuals who are planning, building, managing, and maintaining Web sites. Individuals with the MCP+SB credential will have demonstrated the ability to develop Web sites that include multimedia and searchable content and Web sites that connect to and communicate with a back-end database. It requires passing two of the following three exams: “Designing and Implementing Commerce Solutions with Microsoft Site Server 3.0, Commerce Edition,” “Designing and Implementing Web Sites with Microsoft FrontPage 98,” and “Designing and Implementing Web Solutions with Microsoft Visual InterDev 6.0.”
- *MCSD (Microsoft Certified Solution Developer)*—The MCSD credential reflects the skills required to create multitier, distributed, and COM-based solutions, in addition to desktop and Internet applications, using new technologies. To obtain an MCSD, an individual must demonstrate the ability to analyze and interpret user requirements; select and integrate products, platforms, tools, and technologies; design and implement code and customize applications; and perform necessary software tests and quality assurance operations.

To become an MCSD, you must pass a total of four exams: three core exams and one elective exam. Each candidate must choose one of these three desktop application exams—“70-016: Designing and Implementing Desktop Applications with Microsoft Visual C++ 6.0,” “70-156: Designing and Implementing Visual FoxPro 6.0,” or “70-176: Designing and Implementing Visual Basic 6.0”—*plus* one of these three distributed application exams—“70-015: Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0,” “70-155: Designing and Implementing Distributed Applications with Visual FoxPro 6.0,” or “70-175: Designing and Implementing Distributed Applications with Visual

Basic 6.0.” The third core exam is “70-100: Analyzing Requirements and Defining Solution Architectures.”

Elective exams cover specific Microsoft applications and languages, including Visual Basic, Visual C++ 6 Distributed (the subject of this book), the Microsoft Foundation Classes, Access, SQL Server, Excel, and more. You cannot use the same exam to satisfy two requirements. For example, if you take and pass “Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0” to satisfy the core “Distributed Application Development” requirement, you cannot also use the exam to satisfy your elective requirement.

If you’re on your way to becoming an MCSD and have already taken some exams, visit www.microsoft.com/train_cert/ for information about how to proceed with your MCSD certification under this new track. Table 1 shows the requirements for MCSD certification:

- *MCDBA (Microsoft Certified Database Administrator)*—The MCDBA credential reflects the skills required to implement and administer Microsoft SQL Server databases. To obtain an MCDBA, an individual must demonstrate the ability to derive physical database designs, develop logical data models, create physical databases, create data services by using Transact-SQL, manage and maintain databases, configure and manage security, monitor and optimize databases, and install and configure Microsoft SQL Server.

To become an MCDBA, you must pass a total of five exams: four core exams and one elective exam. The required core exams are “Administering Microsoft SQL Server 7.0,” “Designing and Implementing Databases with Microsoft SQL Server 7.0,” “Implementing and Supporting Microsoft Windows NT Server 4.0,” and “Implementing and Supporting Microsoft Windows NT Server 4.0 in the Enterprise.”

The elective exams that you can choose from cover specific uses of SQL Server and include “Designing and Implementing Distributed Applications with Visual Basic 6.0,” “Designing and Implementing Distributed Applications with Visual C++ 6.0,” “Designing and Implementing Distributed Applications with Microsoft Visual FoxPro 6.0,” “Designing and Implementing Data Warehouses with Microsoft SQL Server 7.0,” and two exams that relate to Windows NT: “Internetworking with Microsoft TCP/IP on Microsoft Windows NT 4.0” and “Implementing and Supporting Microsoft Internet Information Server 4.0.”

Note that the exam covered by this book can be used as the elective for the MCDBA certification. Table 2 shows the requirements for the MCDBA certification.

Table 1 MCSD Requirements*

Core

Choose 1 from the desktop applications development group

Exam 70-016	Designing and Implementing Desktop Applications with Microsoft Visual C++ 6.0
Exam 70-156	Designing and Implementing Desktop Applications with Microsoft Visual FoxPro 6.0
Exam 70-176	Designing and Implementing Desktop Applications with Microsoft Visual Basic 6.0

Choose 1 from the distributed applications development group

Exam 70-015	Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0
Exam 70-155	Designing and Implementing Distributed Applications with Microsoft FoxPro 6.0
Exam 70-175	Designing and Implementing Distributed Applications with Microsoft Visual Basic 6.0

This solution architecture exam is required

Exam 70-100	Analyzing Requirements and Defining Solution Architectures
--------------------	--

Elective

Choose 1 from this group

Exam 70-015	Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0
Exam 70-016	Designing and Implementing Desktop Applications with Microsoft Visual C++ 6.0
Exam 70-019	Designing and Implementing Data Warehouses with Microsoft SQL Server 7.0
Exam 70-024	Developing Applications with C++ Using the Microsoft Foundation Class Library
Exam 70-025	Implementing OLE in Microsoft Foundation Class Applications
Exam 70-029	Designing and Implementing Databases with Microsoft SQL Server 7.0
Exam 70-055	Designing and Implementing Web Sites with Microsoft FrontPage 98
Exam 70-057	Designing and Implementing Commerce Solutions with Microsoft Site Server 3.0, Commerce Edition
Exam 70-069	Application Development with Microsoft Access for Windows 95 and the Microsoft Access Developer's Toolkit
Exam 70-091	Designing and Implementing Solutions with Microsoft Office 2000 and Microsoft Visual Basic for Applications
Exam 70-097	Designing and Implementing Database Applications with Microsoft Access 2000
Exam 70-105	Designing and Implementing Collaborative Solutions with Microsoft Outlook 2000 and Microsoft Exchange Server 5.5
Exam 70-152	Designing and Implementing Web Solutions with Microsoft Visual InterDev 6.0
Exam 70-155	Designing and Implementing Distributed Applications with Microsoft FoxPro 6.0
Exam 70-156	Designing and Implementing Desktop Applications with Microsoft Visual FoxPro 6.0
Exam 70-165	Developing Applications with Microsoft Visual Basic 5.0
Exam 70-175	Designing and Implementing Distributed Applications with Microsoft Visual Basic 6.0
Exam 70-176	Designing and Implementing Desktop Applications with Microsoft Visual Basic 6.0

*This is not a complete listing—you can still be tested on some earlier versions of these products. However, we have tried to include the most recent versions so that you may test on these versions and thus be certified longer. We have not included any tests that are scheduled to be retired.

Core exams that can also be used as elective exams can be counted only once toward certification. The same test cannot be used as both a core and an elective exam.

Table 2 MCDBA Requirements

Core

All 4 of these are required

Exam 70-028	Administering Microsoft SQL Server 7.0
Exam 70-029	Designing and Implementing Databases with Microsoft SQL Server 7.0
Exam 70-067	Implementing and Supporting Microsoft Windows NT Server 4.0
Exam 70-068	Implementing and Supporting Microsoft Windows NT Server 4.0 in the Enterprise

Elective

Choose 1 from this group

Exam 70-015	Designing and Implementing Distributed Applications with Microsoft Visual C++ 6.0
Exam 70-019	Designing and Implementing Data Warehouses with Microsoft SQL Server 7.0 Exam
Exam 70-059	Internetworking with Microsoft TCP/IP on Microsoft Windows NT 4.0
Exam 70-087	Implementing and Supporting Microsoft Internet Information Server 4.0
Exam 70-155	Designing and Implementing Distributed Applications with Microsoft Visual FoxPro 6.0
Exam 70-175	Designing and Implementing Distributed Applications with Microsoft Visual Basic 6.0

- *MCSE (Microsoft Certified Systems Engineer)*—Anyone who has a current MCSE is warranted to possess a high level of expertise with Windows NT (version 3.51 or 4) and other Microsoft operating systems and products. This credential is designed to prepare individuals to plan, implement, maintain, and support information systems and networks built around Microsoft Windows NT and its BackOffice family of products.

To obtain an MCSE, an individual must pass four core operating system exams, plus two elective exams. The operating system exams require individuals to prove their competence with desktop and server operating systems and networking components.

You must pass at least two Windows NT-related exams to obtain an MCSE: “Implementing and Supporting Microsoft Windows NT Server” (version 3.51 or 4) and “Implementing and Supporting Microsoft Windows NT Server in the Enterprise” (version 3.51 or 4). These tests demonstrate an individual’s knowledge of Windows NT in smaller, simpler networks and in larger, more complex, and heterogeneous networks, respectively.

Note: The Windows NT 3.51 version will be retired by Microsoft on June 30, 2000.

.....

You must pass two additional tests as well. These tests relate to networking and desktop operating systems. At present, the networking requirement can be satisfied only by passing the Networking Essentials test. The desktop operating system test can be satisfied by passing a Windows 95, Windows NT Workstation (the version must match whichever core Windows NT curriculum you are pursuing), or Windows 98 test.

The two remaining exams are electives. An elective exam may fall in any number of subject or product areas, primarily BackOffice components. However, it's also possible to test out on electives by taking advanced networking tests such as "Internetworking with Microsoft TCP/IP on Microsoft Windows NT" (but the version of Windows NT involved must match the version for the core requirements). If you are on your way to becoming an MCSE and have already taken some exams, visit www.microsoft.com/mcp/certstep/mcse.htm for information about how to complete your MCSE certification.

In September 1999, Microsoft announced its Windows 2000 track for MCSE and also announced retirement of Windows NT 4 MCSE core exams on 12/31/2000. Individuals who wish to remain certified MCSEs after 12/31/2001 must "upgrade" their certifications on or before 12/31/2001. The details are too complex to discuss here; to obtain those details, please visit www.microsoft.com/mcp/certstep/mcse.htm.

Whatever mix of tests is completed toward MCSE certification, individuals must pass six tests to meet the MCSE requirements. It's not uncommon for the entire process to take a year or so, and many individuals find that they must take a test more than once to pass. The primary goal of the *Exam Cram* series is to make it possible, given proper study and preparation, to pass all Microsoft certification tests on the first try.

- *MCT (Microsoft Certified Trainer)*—Microsoft Certified Trainers are deemed able to deliver elements of the official Microsoft curriculum based on technical knowledge and instructional ability. Therefore, it's necessary for an individual seeking MCT credentials (which are granted on a course-by-course basis) to pass the related certification exam for a course and to take the official Microsoft training on the subject, as well as to demonstrate an ability to teach. MCT candidates must also possess a current MCSE.

The teaching skill criterion may be satisfied by proving that one has already attained training certification from Novell, Banyan, Lotus, the Santa Cruz Operation, or Cisco, or by taking a Microsoft-sanctioned workshop on instruction. Microsoft makes it clear that MCTs are important cogs in the Microsoft training channels. Instructors must be MCTs before Microsoft will allow them to teach in any of its official