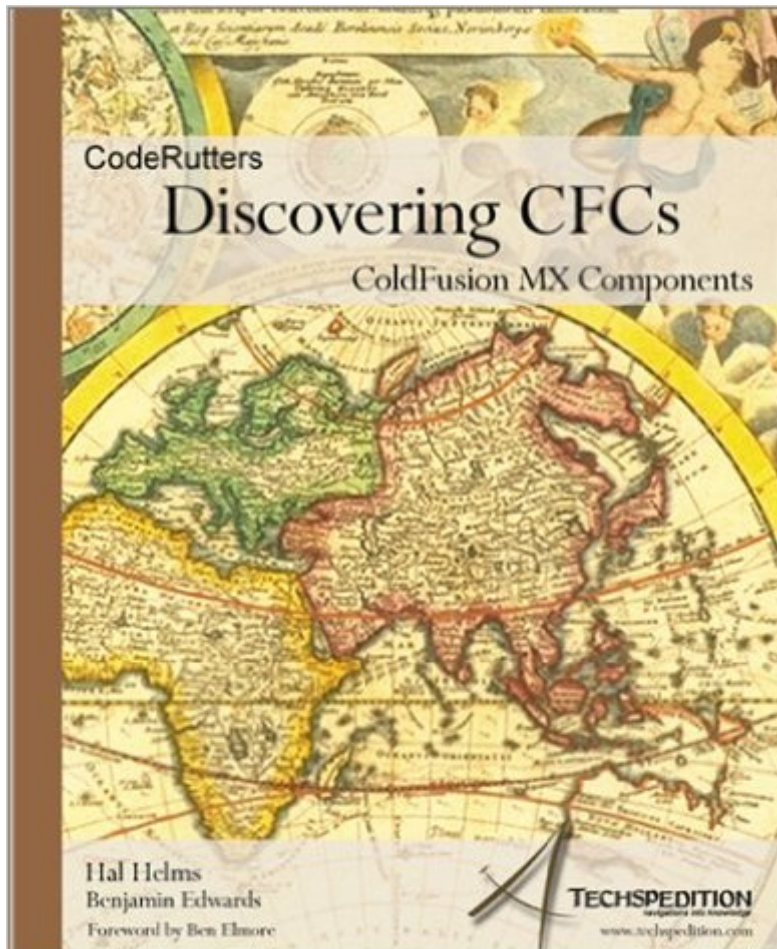


The book was found

# Discovering CFCs: ColdFusion MX Components



## Synopsis

Discovering CFCs begins with object oriented concepts and explains how to use these with ColdFusion MX components. You will discover how to: \*encapsulate your code into CFCs \*use object inheritance to maximize code reuse \*create compositional/aggregational relationships between CFCs \*go beyond the limitations of CFCs \*integrate CFCs with traditional ColdFusion code

## Book Information

Paperback: 168 pages

Publisher: Techspedition (October 9, 2002)

Language: English

ISBN-10: 0972078649

ISBN-13: 978-0972078641

Product Dimensions: 9.1 x 7.2 x 0.5 inches

Shipping Weight: 11.2 ounces

Average Customer Review: 3.6 out of 5 stars [See all reviews](#) (12 customer reviews)

Best Sellers Rank: #5,852,566 in Books (See Top 100 in Books) #68 in [Books > Computers & Technology > Programming > Web Programming > Cold Fusion](#) #4340 in [Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design](#) #9673 in [Books > Computers & Technology > Web Development & Design > Web Design](#)

## Customer Reviews

Description "Discovering CFCs" is a slim book of some 160 pages. Curiously, the paragraphs are numbered. There are some 350 paragraphs in total, plus an appendix chapter. Quite a lot of space is given to diagrams, pictures, and code examples. There is an accompanying workbook in PDF that can be purchased from the publishers techspedition. The authors Hal Helms and Ben Edwards are known to the ColdFusion community as a writer in "ColdFusion Developer's Journal" and as one of the people behind Fusebox. The book has ten chapters. The first is an introduction and the last a short story. The eight core chapters explain the object-oriented concepts, how to do object-oriented programming with CFCs, and the limitations of CFCs with some workarounds. The code examples are often first in Java and then in CFC. In comparison to the Java code it becomes obvious how painful it is to program CFCs, because of the markup-language syntax and the limitations of the language. Helms and Edwards provide good discussion of some of those limitations, and provide workarounds for them. Among the limitations they work around are \* no private properties, \* no constructors, \* no super. They also show how to use custom attributes and the meta-data to make

self-documenting code. The book uses basic UML without making a big deal of it. The last chapter is a fictitious conversation between three programmers about the pros and cons of object-oriented programming. It has a slight entertainment value, but doesn't really add much. The book is aimed at CF developers who want to use CFCs. That is, they want to learn object orientation and how to do it with ColdFusion Components. Critique The book is trying to do three things at once: 1. Teach object-oriented programming. 2. Teach how to program CFCs. 3. Discuss limitations of CFCs and provide workarounds for them. It would be better to do each of those things separately. A reader new to OO is likely to be confused by being taught an OO concept like super-class, then being shown an example in Java (a language that is besides the point of the book), then being shown the CFC code that takes half a page to do what Java does in three lines, and then the poor reader has to digest the limitations of CFCs and the suggested workarounds. What will you get out of the book? \* Readers who don't know OO need better examples and less syntax problems of CFC. They should not read this book, but one of the good books about OO using Java as example language. \* Readers who know OO from Java or C++ or Smalltalk will find how to write CFCs. For them the OO stuff is superfluous. The book gives an introduction on how to write CFCs, but it does not cover the whole CFC syntax, so they will need the MX manual anyway. The book shows how to overcome some CFC weaknesses and that is useful. As concerns style, I personally don't like so many footnotes, especially if they explain important things like the definition of "method signature". Some examples are unrealistic: a cat that knows it can sleep, eat, and play, is taken as an example of an object with methods.

CFC are maybe the most important improvement coming from ColdFusion MX, bringing concepts borrowed from OOP to CFML. This book digs deep into CFC, covering both their great features and their shortcomings and it even uncovers a few bugs along the way. The author's main targets are developers used to CFML and procedural programming that need to get acquainted with the new, OOP-like, paradigm introduced by CFC. Still, seasoned OOP programmers may benefit from it as well, since CFC implementation diverges in many ways from many traditional OOP concepts. Compared to other books of similar size it can be a little overpriced, but you should consider Techspedition is a very small, independent, publisher that really deserve our support

Discovering CFCs is an essential read for any developer that wants to keep up with the movement to Object-Oriented Programming. While ColdFusion isn't OOP, and CFCs don't make it so, they are a step in that direction. This book clearly lays out the principles of OOP and shows where CFCs

comply and depart. After reading this book, you'll have a much clearer understanding of CFCs and be ready to use them in your applications. Kudos to the authors for making a tough subject accessible and useful.

I was a bit daunted at the prospect of learning anything about object oriented programming. The introduction of CFCs in CFMX seemed to really force my hand, and BOY was I glad to find this book focused solely on the topic. A lot of the other CFMX books I saw were huge and covered all aspects of CFMX. This book was a focused treatment of CFCs in depth and done in a way that let me approach the basics of OO in a comfortable environment.

Hard to believe this book had only two authors. A real disappointment from Hal Helms after reading this Fusebox books. Save your money and wait for something more concise and focused to come out.

This book is completely worthless. You are better off reading the documentation in ColdFusion about components. The examples are very simplistic and do not help to address why and how to use components in real-world applications. There are also dozens of mistakes, including spelling, grammar and in the code examples which often make the examples difficult to follow. The numbered paragraphs are useless and seem to serve primarily as a way to increase the number of pages in the book. (If you do not include the code examples, this book probably contains only about 20 pages of writing.) If you do buy this book, I implore you to skip the "Closing Thoughts" section which is an imaginary conversation between 3 developers that rambles on for 10 pages yet says nothing of value. I will say the book is aptly titled: it helps you "discover" CFCs, but it does little to make them useful. I am considering sending this book back and asking for a refund. It's that bad.

I reckon this book is fantastic. I have gotten up to speed with objects in no time. I am writing CF now that has objects doing inheritance and composition and it seems so straightforward. Before this book I was bamboozled. The power of the OO stuff is awesome. I can't imagine now not being able to do things this way. It's only a small book but that is its great strength. You don't have to wade through hundreds of pages so learning is quick. I can't recommend this book highly enough. The only issue I have had with this book is the super keyword. The book is written for version 6.0 and shows a way to get around there being no super keyword available. As of version 6.1 there is a super keyword so that part of the text is obsolete. Anyway, it's only a few pages and it will still work

anyway.

Do not buy this book, it is so out of date and teaches the wrong way of doing things..

[Download to continue reading...](#)

Discovering CFCs: ColdFusion MX Components Visual Developer Developing Custom Delphi 3 Components: Master the Art of Creating Powerful Delphi 3 Software Components Developing Custom Delphi Components: Master the Art of Creating Powerful Delphi Software Components Sound (Discovering Science) (Discovering Science) Discovering Genesis: Content, Interpretation, Reception (Discovering Biblical Texts (DBT)) Living in the Light: 22 Creative Components Including Services, Dialogues, Monologues, Skits, Dramas, Mediations, and a Litany Dan Appleman's Developing COM/ActiveX Components With Visual Basic 6 Creating Lightweight Components with ATL Software Components Ada Software Components With Ada: Structures, Tools, and Subsystems (The Benjamin/Cummings Series in Ada and Software Engineering) Programming Delphi Custom Components Toolkit for Delphi: A Collection of Components, Utilities, and Documentation for Borland's Programming Language Expert Oracle Application Express Plugins: Building Reusable Components (Expert's Voice in Oracle) Software Engineering with Reusable Components Oop: Building Reusable Components with Microsoft Visual Basic .Net CommonKADS Library for Expertise Modelling, Reusable Problem Solving Components (Frontiers in Artificial Intelligence and Applications) Apache Jakarta Commons: Reusable Java(TM) Components Reuse of Off-the-Shelf Components: 9th International Conference on Software Reuse, ICSR 2006, Torino, Italy, June 12-15, 2006, Proceedings (Lecture Notes in Computer Science) Reusable Ada Components Sourcebook (The Ada Companion Series) Using UML: Software Engineering with Objects and Components (2nd Edition)

[Dmca](#)