

# Outside Looking In

**Mark Hunter**

Well, it's a big month. Pro Publishing has teamed up with Pivato Consulting to release XSalCOM. This major new product gives Centura Team Developer customers the ability to create COM automation servers from their present applications, with no code changes required. That means CTD 1.1 and 1.5, not some future 2.0 version. We have been testing and documenting XSalCOM for a long time, but we didn't announce any alpha or beta versions. It's complete, debugged, and functional right now.

I don't want to turn this editorial into an advertisement for XSalCOM. If the timing works out, there will be a separate brochure enclosed with this issue. But the story of XSalCOM reveals some important information about how Centura Software thinks about their development tools. Originally this technology was offered to Centura in the hope that COM server functionality could be announced more than a year ahead of schedule. Centura turned it down, stating that they would rather provide COM server features within a smaller, better-designed runtime engine. That's why you won't see COM server capabilities from Centura Team Developer until the year 2000, and you will have to migrate to CTD 2.0—and hope for stability in that version—to get them. Meanwhile, XSalCOM customers will be deploying actual COM servers now, and when 2.0 finally

arrives, they can take their time choosing whether to migrate or not.

That position about the runtime engine has some validity. CTD does have a pretty large footprint, and it isn't well-suited to transaction servers. And Centura has told developers for many years that a better engine is necessary before things like COM servers, dynamic instantiation, and multi-threading can be accomplished.

But new developments have caused me to question all this conventional wisdom. This week, as I was finishing the XSalCOM documentation edits, Gianluca Pivato announced that he had found a way to do dynamic object instantiation with the current CTD runtime. It's possible to create dynamic objects derived from any class in your outline at runtime. Tens of thousands of objects, if you want. Object pointers. Typecasting. Arrays of objects, with different elements in the same array derived from different classes. No memory leaks. Dynamic instantiation was one of the main features of Tomahawk, and we were told that it was dead when Tomahawk was cancelled. Looks like that wasn't completely true.

How does he do it? We'll have details in an upcoming issue. And Pivato says that he has a multi-threading example working with the present runtime engine, too!

I am aware that Gianluca Pivato has about five times

*Continues on page 12*

Publisher markhunter.com Editor dianschaffhauser.com, Business Manager shelleydoyle.com, Production Editor paulgould.com, Woman's Best Friend mochamix.com

Centura Pro (ISSN: 1093-2100) is published monthly (12 times per year) by Pro Publishing, PO Box 2399, Nevada City, CA 95959.

POSTMASTER: Send address changes to Centura Pro, PO Box 2399, Nevada City, CA 95959.

Copyright © 1998 by Pro Publishing. All rights reserved. No part of this periodical may be used or reproduced in any fashion whatsoever (except in the case of brief quotations embodied in critical articles and reviews) without the prior written consent of Pro Publishing. Printed in the United States of America.

Centura Pro is a trademark of Pro Publishing. Other brand and product names are trademarks or registered trademarks of their respective holders.

This publication is intended as a general guide. It covers a highly technical and complex subject and should not be used for making decisions concerning specific products or applications. This publication is sold as is, without warranty of any kind, either express or implied, respecting the contents

of this publication, including but not limited to implied warranties for the publication, performance, quality, merchantability, or fitness for any particular purpose. Pro Publishing, shall not be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to be caused directly or indirectly by this publication. Articles published in Centura Pro reflect the views of their authors; they may or may not reflect the view of Pro Publishing. Opinions expressed by Centura Software employees are their own and do not necessarily reflect the views of the company.

**Subscription information:** To order, call Pro Publishing at 520-265-4082. Cost of domestic subscriptions: 12 issues, \$119; Canada: 12 issues, \$129. Other countries: 12 issues, \$139. Ask about source code disk pricing. Individual issues cost \$15. All funds must be in U.S. currency.

Call Centura Software Corp. at 650-596-3400.

If you have questions, ideas for bribing authors, or would just love to chat about what you're doing with Centura products, contact us via one of the means at right.

## We've Moved!

**Centura Pro on the Web**  
http://www.ProPublishing.com

**Editorial Department**

Phone: 818-249-1364

Fax: 818-246-0487

E-mail: mhunter@sprintmail.com

**Subscription Services**

Phone: 530-265-4082

Fax: 530-265-0368

E-mail: shelley@propublishing.com

**Mail**

Pro Publishing

PO Box 2399

Nevada City, CA 95959

# Outside Looking In . . .

*Continued from page 2*

more talent than typical developers like me. But remember—he has done all of this work *without* any technical assistance from Centura. In fact, when he asked them for some information about the internals of the runtime engine, he was politely refused. Imagine what would be possible if he did have such access.

So why haven't we seen this kind of innovation from inside Centura? One possible explanation is that there's

no one left at the company who participated in the CTD 1.x runtime engine development. The engineers who are there now may not really understand it. Is that the reason that Centura says a new runtime engine must come before any significant enhancements? Is it because their engineers don't know how to take advantage of their own code?

XSalCOM is an example of a strategic advantage that doesn't have to wait for a new runtime engine. There may be many others. Is it right for Centura to ignore the potential of the current CTD versions as it slowly works toward 2.0? **CP**