

EDITORIAL BOARD

Steve Benfield, Richard Brooks, Joe Celko, Kourous Gorgani,
Judah Holstein, Peter Horwood, Bahadir Karuv, Ph.D.,
David McClanahan, Bernie Metzger, John Olson,
Sean Rhody, Lijun Yang

Editor-in-Chief: John Olson

Art Director: Jim Morgan

Executive Editor: Scott Davison

Managing Editor: Hollis K. Osher

Senior Editor: M'lou Pinkham

Production Editor: Brian Christensen

Internet Editor: Andy Blum

Technical Editor: Bernie Metzger

Watcom SQL Editor: Joe Celko

DataWindows Editor: Richard Brooks

WRITERS IN THIS ISSUE

Michael Barlotta, Jason Cohen, Gerald Giggs, Brian J. Grimm,
Bob Hendry, Lucy Kenton, David McClanahan, Bernie Metzger,
Michael Ogrinz, John Olson, Sean Rhody, Peter Vassilatos

SUBSCRIPTIONS

For subscriptions and requests for bulk orders,
please send your letters to Subscription Department

Subscription Hotline: 800 513-7111

Cover Price: \$14/issue.

Domestic: \$149/yr. (12 issues)

Canada/Mexico: \$169/yr.

All other countries: \$179/yr.

(International subscriptions by U.S. Banks or Money Orders only)

Back Issues: \$12 each

Publisher, President and CEO: Fuat A. Kircaali

Vice President, Production: Jim Morgan

Vice President, Marketing: Carmen Gonzalez

Accounting: Ignacio Arellano

Graphic Designers: Robin Groves

Alex Botero

WebMaster: Robert Diamond

Senior Web Designer: Chad Stiler

Advertising Assistant: Robyn Forma

Customer Service: Sian O'Gorman

Paula Horowitz

Online Customer Service: Mitchell Low

EDITORIAL OFFICES

SYS-CON Publications, Inc.

39 E. Central Avenue, Pearl River, NY 10965

Telephone: 914 735-1900

Fax: 914 735-3922

POWERBUILDER DEVELOPER'S JOURNAL is published
monthly (12 times a year) for \$119.00 by SYS-CON Publications, Inc., 39 E. Central
Ave., Pearl River, NY 10965-2306. Periodicals Postage rates are paid
at Pearl River, NY 10965 and additional USPS offices. POSTMASTER:
Send address changes to: POWERBUILDER DEVELOPER'S JOURNAL, SYS-CON
Publications Inc., 39 E. Central Ave, Pearl River, NY 10965-2306.

© COPYRIGHT

Copyright © 1999 by SYS-CON Publications, Inc. All rights reserved. No part of
this publication may be reproduced or transmitted in any form or by any means,
electronic or mechanical, including photocopy or any information storage and retrieval
system, without written permission. For promotional reprints, contact reprint coordinator,
SYS-CON Publications, Inc., reserves the right to revise, republish and authorize its
readers to use the articles submitted for publication.

ISSN # 1078-1889

All brand and product names used on these pages are trade names,
service marks or trademarks of their respective companies.
SYS-CON Publications, Inc., is not affiliated with the companies
or products covered in PowerBuilder Developer's Journal.

FROM THE EDITOR

By John Olson, Editor-in-Chief



Wake Up and Smell the Coffee!

Are you caught up in the hype of the millennial hysteria, frantically analyzing your systems and devising contingency plans? Or are you waiting and watching, hoping that nothing will happen, but "ready" to leap into action if it does? Many companies are adopting a "wait and see" approach. If nothing happens, the managers will smile, give each other high fives and count all the money they saved in labor, training, analysis tools and consulting. But can they safely assume their systems will prove to be compliant?

Many of us assume that since both PowerBuilder and our databases are Y2K compliant, the software we have written is also compliant. Unfortunately, having compliant tools just means that the tools themselves won't be the cause of Y2K failures. Code written with these tools can be riddled with Y2K bugs. A good analogy is that of writing object-oriented code in PowerBuilder. Sure, PowerBuilder provides the ability to write object-oriented code, but it doesn't force code to be object oriented. PowerBuilder itself is compliant, but developers can very easily write noncompliant code without knowing any better. Unfortunately, this is true of most developers because they are unaware of the issues involved in writing Y2K-compliant code. Without proper education they not only write noncompliant code, but they aren't equipped to find, much less fix, the problems. Peter Vassilatos has written a powerful article that delves deeply into the subject of Y2K-compliant PowerBuilder code. In a related article, Brian Grimm, one of the developers of the PFC, writes about Y2K compliance of the PFC. I strongly encourage you to read both articles.

We cannot assume our systems are Y2K compliant. No one can. Even development teams that have stringent coding standards often let bugs, including Y2K bugs, slip through the code reviews. It's a fact of life in the coding world - no software is bugless. Unfortunately, unless comprehensive and astute analysis is targeted at Y2K bugs, they won't be found until they are experienced in a production environment. The resulting damage to data could be catastrophic, even life threatening.

I have a client who has a system critical to the health of thousands of people. It's truly a "critical" system. Data corruption or failures in this type of system could lead to deaths and, in fact, has in the past. Many of us on this project were confident that the system, built with PowerBuilder 6.5 and Sybase SQL Server System 11, was Y2K compliant. However, out of 300,000 lines of code we found over 250 high-risk areas that needed to be closely scrutinized. Of those, I determined that 50 were noncompliant. I expected to find a few, not 50. In this life-critical system even ONE could have dire consequences. But 50! This experience shocked me into recognizing the potential for widespread system disruptions.

Are you confident that your systems are compliant? Have you performed comprehensive Y2K analysis and testing of your systems? Those questions are one and the same. Confidence can come only through proper analysis and testing. If you haven't done the proper testing, then you might be confusing confidence with hope.

What happens if January 1 comes and a company that takes the "wait and see" approach finds that one or more of their critical systems has failed? They call in their programming staffs and work hard until the problems are fixed. In the worst-case scenario the system is totally offline, and they are forced to revert to paper systems. Backup paper systems are at best too slow but are also less accurate. A major airline recently stated they have their paper

(continued on page 38)

"Owing to past neglect, in the face of the plainest warnings, we have now entered upon a period of danger. The era of procrastination, of half measure...of delays, is coming to its close. In its place, we are entering a period of consequences....We cannot avoid this period; we are in it now. Unless...this body resolves to find out the truth for itself, it will have committed an act of abdication of duty without parallel."

—Winston Churchill