

Centura Pro

Visit us at www.ProPublishing.com

Hot Ideas for Centura® Developers

I'll Call You Back

Christian Schubert

A callback procedure is a Windows mechanism to handle asynchronous events—events you trigger somehow (by calling a function) but which execute at some time unknown to you in the future. In some ways they're much like messages but with at least two differences:

- Callback procedures don't need window handles.
- Callback procedures can carry more than two parameters, an important advantage.

That advantage is the main reason callback procedures are used. The maximum number of parameters in sample callbacks that I researched was six; the minimum was just one.

SQLWindows

32

The problem with callbacks in SQLWindows

Some useful functions found in the Windows SDK require the address of a callback procedure. This is a problem with SQLWindows, because you can't determine the address of a SAL function at runtime. So the solution for the callback problem is going to require some C coding. However, I'll provide a "one size fits all" solution that can be used for almost any purpose and doesn't require you to dive into the depths of C programming or even own a C compiler.

The solution

The solution should have the following features:

- It must be written in C to be able to provide a pointer to a function.

Using Windows callback procedures in SQLWindows has always been difficult. There are several approaches to mastering them, including coding special DLLs for each purpose or using the object compiler. There was no global solution—until now!

- It must fit for any possible number of parameters of callback procedures.
- It must provide a way to write the code for the callback procedure in SAL.

The solution is a small DLL written in C. It contains its own procedure for each number of parameters required and relies on the fact that *every* parameter of *every* callback function uses four bytes. This is true for any type of pointer: handle, DWORD, long, and so on. (I've found no exception to this rule so far.)

The callback procedure hands over its parameters to a SAL function by calling `SalCompileAndEvaluate`. The name of the SAL function to be called can be set by the application.

December 1999

Volume 4, Number 12

- 1 I'll Call You Back
Christian Schubert
- 2 The Planets are in Alignment
Mark Hunter
- 5 JobShop!
- 6 New Progress in Progress Bars
Eugene Toperman
- 7 Report from the Field: Centura Grabs Hold of the Future
Martin Knopp
- 10 Manage Form Windows Better with `VisWinsWindow()`
Michael Lamon
- 12 Centura Tip: Dynamic Tool Tips for Windows 98
Bill Grzanich, SAM User



Continues on page 3

The Planets are in Alignment

Mark Hunter

The deadline for this issue was held open to include news from the Berlin conference, and good news it was! Check out Martin Knopp's impressions in this issue. At nearly the same time, Centura's stock price skyrocketed in response to important new working agreements with SAP and Nocom. And eSNAPP was well received at the Comdex conference. Everything is pointing in the right direction for Centura today.

There's a fascinating new service being provided by www.centurahosting.com. They have a Web-connected server with unlimited-license SQLBase, where you can locate your databases and applications for remote access. It raises some unique questions. What happens when you give a client administrator authority to their database, yet you need to lock out all the other databases on the same server, even to the point of making them invisible to tools like SQLTalk? We'll cover this service in depth in an upcoming issue.

At the Berlin Conference, there was some talk about a "wrecking crew test" for the Matterhorn release of Centura Team Developer. I asked Charity Silkebakken, who manages CTD and other tools, for some clarification. Her response:

The Wrecking Crew is composed of internal Centura offices and selected customers that receive the release candidate of the product and just hammer the heck out of it. This exposes things that the beta program doesn't, because until a product is declared a release candidate, people are afraid to convert their real applications to the new version. Once we have real customer applications running with a new version, a lot of things come to light. Any regressions from the previous version that are found are fixed before shipping the product. (We don't fix every bug; we can't. We do triage. We try to fix every regression.) So far, CTD 1.5.1, SQLWindows 5.0.5, and SQLBase 7.5.1 have all had a Wrecking Crew. None of those had beta programs, so Matterhorn will be different. I am not sure if I will be able to allow all beta testers to participate. There will be a boatload of beta testers for this program, and we can't realistically manage a Wrecking Crew of more than, say, 25 customers. Therefore it is likely that only those who are most active in the beta program (asking questions, reporting new bugs, offering suggestions) will be invited to participate in the Wrecking Crew.

OK, beta testers, get your feedback ready! Happy holidays, and good luck to everyone with your Y2K runs!

CP

Finishes Big Mark Hunter, Wraps it Up Dian Schaffhauser, Brings it to an End Shelley Doyle, Sends it Off Paul Gould, Lays it Down Mocha

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 © 1999 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 530-265-4082. 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.

Contact Us

Centura Pro on the Web
www.ProPublishing.com

Editorial Department

Phone: 818-249-1364

Fax: 818-246-0487

E-mail: huntersoftware@netscape.net

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

I'll Call You Back ...

Continued from page 1

How it works

Let's look at the simplest kind of callback procedure: one parameter. Let's say the procedure to handle the callback in our SAL application is named MyCallback. It's defined in the form window frmTest and looks like this:

```
◆Function: MyCallback
  ◇Description:
  ◇Returns
  ◆Parameters
    ◇Number: nParam1
  ◇Static Variables
  ◇Local variables
  ◆Actions
    ◇! do something useful here...
```

We have to tell the DLL what SAL function should be called and how many parameters it uses. We do this by calling SetCallback:

```
◇Call SetCallback ( "frmTest.MyCallback", 1, TRUE,
  pCallbackProc, hModule )
◇If pCallbackProc = 0
  ◆! something went wrong, bail out
  ...
```

SetCallback takes the following parameters:

- *sProc*—Name of the SAL function to be called. It's always a good idea to fully qualify the functions name.
- *nNumParams*—Number of parameters to be used.
- *bReportErrors*—Report errors when calling SalCompileAndEvaluate.

Listing 1.

```
◆Function: SHBrowseForFolder
  ◇Description:
  ◇Export Ordinal: 0
  ◆Returns
    ◇Number: LONG
  ◆Parameters
    ◆structPointer
      ◇Window Handle: HWND ! parent windows of browsing dialog
      ◇Number: LPVOID ! pointer to start directory (not used here)
      ◇Receive String: LPSTR ! buffer containing the folder the user chose
      ◇String: LPSTR ! text to be shown in the dialog
      ◇Number: UINT ! flags, control appearance and behavior
      ◇Number: LPVOID ! pointer to callback function
      ◇Number: LPARAM ! application defined value for callback
      ◇Receive Number: INT ! image of selected folder (not used here)
```

Listing 2.

```
Number: nWnd ! window handle of the dialog
Number: nMsg ! type of callback message
Number: lParam ! value dependant on the message type
Number: lpData ! applications defined value (used to distinguish different callers)
```

- *pProc*—Receive parameter; contains the pointer to the callback procedure if everything was OK; otherwise, it's zero.
- *pModule*—Receive parameter; contains the module handle needed by some SDK functions.

The C callback procedure pointed to by pProc after calling SetCallback looks like this:

```
void CALLBACK CallbackProc1(DWORD dwParam1)
{
  TransferParams ( 1, dwParam1 );
}
```

TransferParams builds the following string and executes it via SalCompileAndEvaluate:

```
frmTest.MyCallback ( 12345 )
```

We assumed that dwParam1 contained the value 12345. The values of the parameters are converted to numeric constants when building the statement. The callback's parameters can now be used within our SAL function.

Samples

I include two sample applications to show how callbacks can be used.

The first one covers a topic that is frequently asked for in Centura's newsgroups and forums: the Windows folder-browsing dialog (see [Figure 1](#) on page 4). The SDK function used here is *SHBrowseForFolder*. It takes a pointer to a structure that contains information about the desired action (a description of the parameters follows):

```
typedef struct _browseinfo {
  HWND hwndOwner;
  LPCITEMIDLIST pidlRoot;
  LPSTR pszDisplayName;
  LPCSTR lpszTitle;
  UINT ulFlags;
  BFFCALLBACK lpfn;
  LPARAM lParam;
  int iImage; }
BROWSEINFO
```

This structure can be defined as an external function, as shown in [Listing 1](#).

SHBrowseForFolder is embedded into a functional class that also contains the required variables and the callback function. You need the callback to be able to set the initial folder and do some validation. The callback function takes four parameters, as shown in [Listing 2](#).

The callback is triggered for various reasons. Once it's sent right before the dialog is shown (nMsg =

BFFM_INITIALIZED) and lets us modify the initial directory, the title of the dialog, or the status text. It's also triggered when the user selects a folder (nMsg = BFFM_SELCHANGED) or clicks OK and the edit box contains an invalid folder (nMsg = BFFM_VALIDATEFAILED).

As you see there are many possibilities for influencing the dialog.

SHBrowseForFolder returns the pointer to a shell identifier list containing the folder. It must be extracted to a string by using the function *SHGetPathFromIDList*. Please note that the return value of *SHBrowseForFolder* points to a chunk of memory allocated by the shell. It would be the responsibility of the calling application to release this memory. However, the technique needed here can't be used with *SQLWindows* (at least I haven't found out how, but I'm still researching it). It would require us to deal with pointers to class interfaces (as returned by *SHGetMalloc*, just for those of you who are interested). So if you use *SHBrowseForFolder* the way we did here, you'll have a small (but probably acceptable) memory leakage.

The second sample (Figure 2) shows the definition of a keyboard system hook, which also requires a callback procedure. This procedure gets all keyboard events before the application gets them. This could be handy if you need special accelerators and don't want to (or can't) use the Visual Toolchest accelerator functions.

The callback function required here takes three parameters. We're only interested in two of them: If *nCode* is less than zero, we shouldn't process the event. *lParam* contains information about the keys that caused the event. We use *GetKeyNameText* to get a descriptive name of the key and display it.

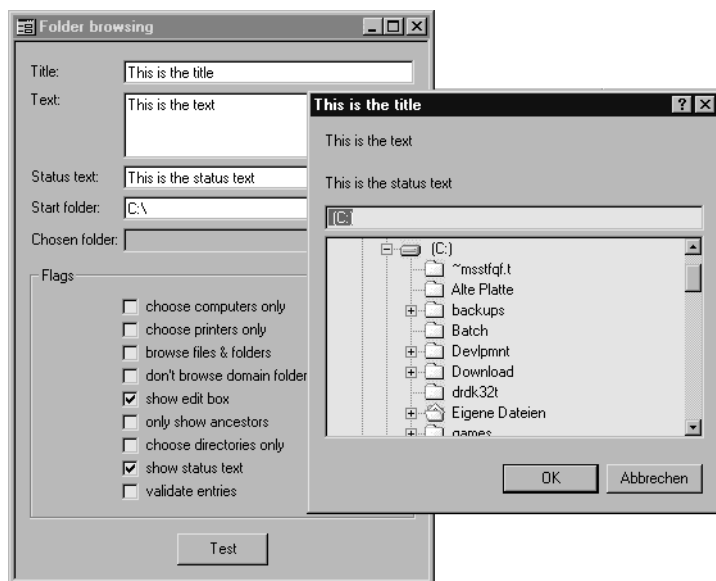


Figure 1. The shell folder browser.

Using more than one callback simultaneously

If you ever need to have more than one callback procedure at the same time, you can use the following trick to accomplish this:

1. Copy the file *callback.dll* to *callback2.dll* (or any other filename you like).
2. Add the following external function to your code:

```
◆Library name: Callback2.dll
◆Function: SetCallback
◇Description:
◇Export Ordinal: 1
◇Returns
◆Parameters
◇Number: INT
◇String: LPVOID
◇Number: INT
◇Boolean: BOOL
◇Receive Number: LPLPVOID
◇Receive Number: LPHANDLE
```

3. Now you have another callback available. You may repeat these steps if more simultaneous callbacks are needed.

Caveats

There are also some things to watch for when handling callback procedures.

- Be careful when writing hook procedures. You might get in trouble if your procedure generates events that trigger a new call of the hook procedure. This will almost certainly crash your application. Sometimes it's not even enough to provide a bypass mechanism to prevent this message stack overflow. I didn't manage to get a window message hook to work, which is probably due to the way *SQLWindows* handles its own function calls.

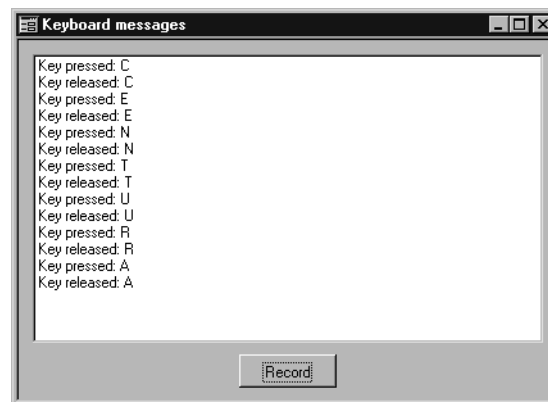


Figure 2. The output window of the keyboard hook.

- The method of using callback procedures as shown here only works with callback functions using four-byte parameters. Though I haven't found any callback procedures of a different kind, there's no proof that they don't exist. So have a close look at the definition of the callback procedure before using it.
- If you reuse a callback procedure by overwriting the definition of the SAL function, make sure that the old callback won't be triggered any longer, or the SAL function might be called with a wrong number of parameters.
- After calling *SetCallback*, check that *pProc* isn't zero. If you submit a NULL pointer to a function, you might get a GPF.

- I've seen applications that use fake callback procedures for communication between different programs. These procedures aren't defined as CALLBACK (in C) and have a totally different way of placing their parameters on the stack. Therefore you'll also get some nice GPFs if you try to use them. **CP**

Download *Callback.zip* from www.propublishing.com or find it on this month's Companion Disk.

Christian Schubert works as Senior Developer at GODEsys GmbH, Mainz, Germany. He has been using Centura products since 1993 and develops sales management applications. He can be reached at christian.schubert@mainz.netsurf.de.



Congruity Corp.

SQLWindows/32 programmers

SQLWindows/32 programmers needed for multiple contracts! Congruity Corp. is looking for bright, motivated individuals interested in the growing client/server to Web marketplace. We develop technologies that glue database applications to the Internet. We provide for our customers the best solutions and tools on the market today to ensure seamless application development. Projects include: Web enabling, integration, client/server application development, database, testing, etc. Join our rapidly growing company! For more information, visit our Web site www.congruitycorp.com. EOE.

Contact

mike@congruitycorp.com, fax 408-559-2024

Geac SmartEnterprise Solutions

Senior Software Engineer

Responsible for intranet/Internet application/JavaBean development including analysis, design, and development of business software using Java, Java Servlets, Java Server Pages, and HTML. Perform moderately complex programming tasks within appropriate schedule, task, and quality guidelines set in conjunction with the manager of the project. Candidate will use a proven software design and development methodology. Development experience with each of the following environments is required: Centura, SQLWindows, Windows 95, 98, or NT, Microsoft Internet Explorer or Netscape browsers with Web applications. Any mainframe application, business application, or middleware experience highly desired. Will be team member on the Net Foundations / Connector

Foundation project teams working with other software developers, application experts, testers and documentation resources.

Required skills

- Java, HTML programming experience
- Experience using OO Design and Development methodologies (UML, CRC, Booch...)
- Experience developing software for an intranet/Internet environment
- Familiar with using common software deliverables (data models, design documents, requirements documents, etc.)
- Basic understanding of design and coding techniques
- Effective written and verbal communications
- Time management
- Basic knowledge and understanding of the relevant technologies and platforms
- Ability to function in a team environment

Desired skills

- Centura programming
- Java Servlets, Java Server Pages
- CORBA, COM, or RMI development
 - Mainframe knowledge
 - Understanding of business applications or product being developed

Contact

Huey Gerald Associates, Inc.,
770-973-8944, 800-410-1927
Hga@bellsouth.net;
hueygeraldassociates.com.

List your Centura-related positions in a future issue of *Centura Pro!* Send details to huntersoftware@netscape.net. Include your contact details. We guarantee publication of only those positions that sound truly state of the art.

New Progress in Progress Bars

Eugene Toperman

My corrections allow the progress dialog to be a better “partner” with your application. The progress dialog is a process that works at the same time as the main process. The main process does useful work, and the progress dialog’s process does the monitoring job. It’s clear that the progress dialog’s process should *not* create a heavy load on the main process. If the progress dialog cycle of monitoring takes too much time, we have a problem: Performance of the main process degrades significantly. For example, if the progress dialog is used to show population of a table window, and each row fetched from the database causes progress to be updated, we can spend too much time on the very useful but secondary monitoring process.

SOLWindows

16 / 32

What’s new?

I’ve made changes to my original progress dialog. The core is still the same—you can read my original article about the progress dialog to get to know the basics. Let’s talk about changes.

The progress dialog now works in a very different mode. The old version runs in a straightforward way; you send the PAM_Progress message with some frequency that you define, and the progress dialog just shows you where you are. Things are changed in the current version. Now when you send PAM_Progress, the new version only stores the progress value (zero to 100 percent) inside the dialog in local variable nPROGRESS. The actual time of refreshing the progress bar depends on your choice—now you can set refreshing frequency in seconds using message PAM_SetRefreshTime. Set wParam of this message to desired frequency in seconds, sending the message any time you want. When the progress dialog is at the SAM_Create phase, this frequency is set to the default value, 5.0 seconds. Once again, you can change

The January 1997 article, “Making Progress (Bars),” described a solution to show progress in long-running tasks. This solution was designed as a modeless dialog. Times are changing, and some corrections need to be made in order to optimize performance of the progress dialog.

this refreshing frequency, and the pace of refreshing will be changed.

The progress dialog now works in an asynchronous manner, not in a synchronous manner as before. You set refreshing frequency and the progress dialog “steals” valuable CPU time from the main process only when you allow it, by setting the optimal value for refreshing frequency. So you can tune performance of monitoring, which is

very important. Coming back to our famous population of a table window, if fetching and processing time for each row is comparable to progress-showing time, you need to update progress less frequently, and the new version allows you to do it.

So, what you can do in addition to the things described in the January 1997 article? Set refreshing frequency any time you want like this:

```
call SalSendMessage( dlgProgress,
PAM_SetRefreshTime, 10, 0 )
```

where 10 (seconds) is refreshing time. Every 10 seconds the progress dialog will refresh the progress bar.

You can also send the PAM_Progress message *not* to show progress but to inform the progress dialog about changes in progress, and the progress dialog will take care of actual refreshing when refresh time comes.

Last, you can change refreshing frequency on the fly, and progress will measure your process more/less frequently, producing a more detailed progress picture. See the sample source files for more details. **CP**

Download Newbar.zip from this issue’s Table of Contents at www.ProPublishing.com or find it on this month’s Companion Disk.

Eugene Toperman is a Senior Software Developer in the R&D division of BEZ Systems, Inc. Reach him at BEZPLUS@BEZ.COM.

Report from the Field:

Centura Grabs Hold of the Future

Martin Knopp

The week of November 15th, 1999 was quite a busy one for Centura: two conferences, Comdex, and a nice increase in its stock price caused by some very interesting announcements (the eSnapp launch, the SAP deal in Japan, and a Wireless Access Protocol project with Nocom). What you'll find here is a summary and comments about the Customer Conference held in Berlin.

Please take into account that it isn't just a simple summary about the sessions. It also contains what I read between the lines and some of my own thoughts.

In the beginning

The first day consisted of some general sessions and keynotes starting with **Heinz Höreth, Directing Manager, Centura Germany**. He did a short introduction and all the boring organizational stuff that's needed to run a conference with about 350 attendees. And that's the first great thing—the last big conference in Europe two years ago in Munich was attended by some 100 people. Seeing the increase in attendance was the first sign that Centura is back on track and growing again.

The next session, called "2 Years with Centura—Successful Turnaround," was presented by **Scott Broomfield, CEO of Centura**. To underline the theme of this conference to be a "German Conference" Scott did his first few sentences in German. After that he gave an update on the strategy and vision of Centura, and it was a relief to see that there are no strange things coming up. Centura is simply continuing what was started two years ago. Right now the company is completely focused on Microsoft DNA architecture, component-based development, Web-enabling applications, embeddable databases, and solutions for appliance computing. All of these topics were further discussed in the sessions during the next two days.

Couldn't make it to the Centura Customer Conference in Berlin? That's OK—the author will make you feel like you were there.

What was great to see in Scott's session is that during the past two years every single word was about getting back credibility. Right now that is no longer an issue; so everything is focused on the future and on the vision of the company. That was not only my feeling about the presentation but also what I felt from the audience and all the people I talked to. Not a single word about the credibility of Centura but a lot of ideas for the future.

Next in line was a keynote speech presented by **Chris Atkinson, Director Business Solutions Group, Microsoft EMEA**. He talked about "Microsoft's WinDNA 2000 Architecture." Seeing this presentation in the agenda shows that Centura and Microsoft are committed to really live the partnership that they announced some months ago. We learned about Microsoft's vision for the DNA architecture.

Now it became a little bit more technical in a session named "Centura Products—Technology and Future Directions," held by **Joe Falcone, CTO of Centura**. Joe joined the company about a year ago, and I think he is the right man in the right position now. I don't want to go into much detail about this session here, as you'll find below detailed sections for the different topics. Only let me tell you there are already codenames for CTD releases coming after "Matterhorn": "Trek" is the next and "Fizz" is coming afterwards.

Last but not least for this day was a session of **Charity Silkebakken, Manager Tools Development of Centura**. She introduced the new Centura Complementary Software Partners Program (CSPP). This program is meant for companies who develop third-party products adding value to the Centura product line—especially the tool. There's a new, old person in charge of this program: the well-known **David Burke**. He does all the communication with the CSPP members. Centura will provide co-marketing and technical assistance (up to

getting into their sourcecode if need be). Starting with the "Matterhorn" release of CTD we'll see demo versions of the CSPP member products directly shipped with CTD on the same CD. See the section about CSPP for further details.

After a general Q&A session, there was a great evening event organized by Centura, so maybe you should attend the conference next year so you won't miss that! At this point my congratulations to **Claudia Frotscher, Centura Germany**, who was, as far as I know, responsible for the organization of this great conference and this evening event.

Tools and Internet strategy

The hottest topic here was "Matterhorn," the upcoming release of CTD. Be careful; it's not yet certain that it will be called CTD 2.0. Maybe we'll see a CTD 2000 or SQLWindows 2000. What is sure is that there is available a beta version in existence, and the final release will be delivered in Q1/2000. The major goal of this release is called **COM server generation** and this part is already working. We saw some great demos, and I already created COM servers using the beta myself! Additionally the

current ActiX integration features are also reworked so that "Matterhorn" will support events for non-visual COM components, a feature missing in the current CTD 1.5.1. To make all this happen (it's already in the beta), we get **dynamic instantiation**, both for functional classes and child window controls. For all the techies there's a "new" and "this" function in "Matterhorn." Third, "Matterhorn" will become to be an OLE DB consumer. But

you won't need to learn much new stuff (sure, there are some new Sql* functions to give access to some OLE DB functionality—sessions for example, which make it possible to have more than one transaction at a time, if I understand the concept correctly). You'll continue to use the same Sql* functions you're using right now, but behind the scenes "Matterhorn" can use the native routers from Centura or routers from an OLE DB provider. What that means for the future is obvious—there will now be new native routers, as all major database vendors will ship OLE DB providers for their database products. Centura told us that the performance of its own OLE DB provider for SQLBase is much the same as the native router—so why should we not switch to the modern OLE DB concept?

ReportBuilder saw some enhancements but only minor things. No big rework there. The good thing is that Centura (namely, Joe Falcone and Charity Silkebakken) is well aware that the reporting capabilities of CTD need to be improved; so I hope to see some changes there in "Trek" or at least "Fizz" (the codenames for the next steps of CTD after "Matterhorn"). The only reason we didn't see improvements in "Matterhorn" is a limit in resources.

Regarding Internet strategy, it's obvious for me that Centura is going the "ASP+COM" way. What that means to me is that they position "Matterhorn" as the tool of choice to develop business logic COM components to be used by any Internet development tool/solution that's COM-compliant. That's what the Microsoft DNA architecture suggests anyway; so it's sensible for Centura to follow. What's currently known as Centura Web Developer (CWD) will remain and be improved (both from the point of scalability and from the point of adding better layout capabilities); but in "Matterhorn" it will no longer be called CWD, and it won't be available as a standalone product—it will be called "Centura Web Extensions" and ship only as part of "Matterhorn."

Last but not least, there was one statement in Joe Falcone's session on the first day that I haven't yet mentioned because I think it fits better here. In presenting his vision of the future, he talked about a CTD runtime for running non-visual CTD applications on Linux. To make it clear, they're not going to port CTD to Linux, and they're not going to port the whole runtime to Linux. Right now they're not even working on or planning anything for Linux. But the company is recognizing that Linux might become a platform of choice for running business logic components on a Linux-based server. Please, don't misunderstand what I write here—you won't see this happen during the next several months; but take it as an indicator that Centura is very closely monitoring the market. There's one thing that makes this move more complicated than it might sound—"Matterhorn" will be able to create COM components. Unfortunately, there's no COM on Linux; so Centura either has to wait until there will be a COM

Berlin Quote . . .

Charity Silkebakken

The report that I gave to Jeremy over the phone was that the mood of this conference was quite good. I think it was quite an improvement over the Orlando conference. Attendance was very good, the stock price was up, everyone was impressed with the presentation and the demos of CTD "Matterhorn," and the Complementary Software Partner Program was very well received. In short, everyone was stoked. There was also a lot of interest in what we are doing with eSnapp and our ventures with SAP. I think we all felt like we were finally free of the "old Centura" and were making a fresh start. I enjoyed meeting the members of Ice Tea Group and the other Complementary Software Partners face-to-face, and look forward to a new kind of relationship with our development partners.

implementation on Linux, which might never happen—or if it did would never get credibility in the Linux world since it comes from Microsoft. So going this route would mean for Centura also going the step to implement either CORBA or EJB as the component model. Maybe we should start a thread on the forum.wishlist newsgroup to discuss our needs and pros/cons of possible solutions.

What's going on with net.db? net.db will be continued, and the new release commonly known as net.db 2.0 but internally known as "Unity" is in beta already. It gives you much better layout capabilities as well as the most important new feature—business logic integration. Sorry, but I did not figure out for sure how this will be established—we heard a lot about "Unity" being able to access COM components for this job but saw a presentation from **Gerd Marinitsch, Consultant for Centura Germany**, who only demonstrated a direct link to CTD. So we all stayed confused about that. And, to be honest, net.db isn't that interesting for me, so I didn't try to dig deeper to figure out the truth. Anyone interested in net.db should also read the following CSPP sections.

Another great session, talking about the "Interface for R/3," was presented by **Uli Klemm of Data Design**. You might wonder why it wasn't presented by someone from Centura, but there's a simple explanation. The Centura Interface for SAP R/3 (CIR/3) was developed by a third-party vendor and only purchased by Centura, and it is now maintained by Data Design, who will also do the further product development. Uli said the future version is focused on providing better classes on the CTD side to make parameter handling easier and add support for accessing BAPI components. Right now it's only based on Remote Function Calls (RFC) to SAP. I think it's a great solution, and there were some people around in the audience who have already developed successful solutions. Just to think that the interface is BAPI-validated by SAP! By the way, it sounds perfectly suited to become another product moved to the CSPP.

Sorry, but I didn't find the time to attend the session about SQLHost.

Complementary Software Partners Program (CSPP)

I already mentioned the basic goal of this program, so let me now list all current participants (in alphabetic order) here:

- Congruity Corp.
- Dataline Software
- Fecher
- Ice Tea Group, LLC (ITG)
- Lynx
- Metex Systems, Inc.
- Softworks

The first session of a CSPP member that I could attend was held by **Eberhard Fecher of fecher GmbH**. Fecher is

participating in the CSPP with Building Blocks, a class library and application framework, including code generators for CTD. In his great presentation he gave a good overview about Fecher GmbH, the current customer base of Building Blocks (including some big companies), and his vision of where Building Blocks should be positioned in the future. That's taking it one step further into providing a framework for developing business components—a strategy that perfectly fits with where Centura is heading with its tool.

The next session was held by **Carsten Barner Nielsen, Ice Tea Group, LLC**. He gave a short overview about the five ITG members and talked a little bit about the products of ITG. The hottest one of the "Intelli" product line is IntelliSal, a great add-on for CTD that gives you the comfort of what is known as IntelliSense in Microsoft Visual Basic and Visual C++. You no longer have to remember all the functions of your classes nor all the parameters of each function. IntelliSal will pop up lists of functions and tooltips with the parameter list right while you're typing your code! The second hot product is Ice Tea Active Pages (ITAP). It's a product that lets you bring your CTD applications to the internet. It's using the same concept as Microsoft Active Server Pages but directly links to your CTD application—it's hard to believe, it's hard to explain, but it works great and it is very easy to install and use. It's scalable, fault tolerant, and secure. The new ITG Web page, which will be online in the next few weeks, is based on ITAP! Additionally there are some other products available from ITG, and also XSal and XSalCOM from Gianluca Pivato. My own ReportWizard (as an ITG-approved product) will be available via the new ITG eCart coming with the new homepage during the next weeks.

Sorry to all the CSPP members that only got mentioned in the list but not in further detail; I had to make some tough decisions about which sessions to attend! You can learn more about all the CSPP members and their products through a link on Centura's Web site. What's interesting is to see Dataline Software mentioned in the CSPP program. Dataline is the company that does the net.db development. So I wouldn't be too surprised if

Berlin Quote . . .

Carsten Barner Nielsen

In my opinion it was a great success. Centura has regained credibility with their customers. They are facing the right direction with bringing corporate data to hand-held devices, and they are cooperating with the huge guys like Microsoft a lot. So I'm quite optimistic about the future. (I wasn't so optimistic some months ago, so they "converted" me, too !)

net.db ceases to be a Centura product and gets moved completely to the CSPP. To be honest, as far as I know, the product management will go back to Dataline; but because they're part of the CSPP, it will continue to perfectly work together with CTD. I think this isn't a bad idea, since it will free some resources at Centura to focus on its core product in the tool sector and keep net.db available via Centura for everyone who has a need for it!

Database products

There was one important deal during the last year regarding the database sector of Centura. In July 1999

Centura acquired Raima Corp. Raima is a vendor of two database products. One is called Raima Database Manager (RDM), which is a small, fast, robust database engine well positioned in the market of embedded databases. It's currently used by a lot of big companies (such as Hewlett Packard and Hughes/AOL). RDM is available on WindowsCE, Linux, and various runtime OSs, and has just been ported to PalmOS. See the section about appliances to read more about RDM.

The second product is

Velocis, a client/server database. It's available on different OSs such as Windows NT and Linux. Because there's an ODBC driver for Velocis and CTD can be used with ODBC databases, you can develop applications with CTD running against a Velocis database on Linux. Congruity Corp. showed this configuration together with Centura at Comdex.

One could be confused because SQLBase and Velocis might be positioned in the same segment. That's only

partly true. SQLBase has the SafeGarde technology as a big advantage against Velocis, whereas Velocis will give you SMP support in the upcoming 3.0 version. And if you're moving away from Windows, you'll find SQLBase available for Novell and Velocis for Linux and other Unix derivatives. Since both products have been available for quite some time and haven't yet gotten into a competitive situation, as we were told during the conference, it shouldn't be a problem for the future either.

What can we expect for the future? The feature that Centura was most specific about is that they will add the patent-pending SafeGarde technology (currently only available in SQLBase) to both RDM and Velocis.

SafeGarde is an end-to-end security technology that encrypts both the network traffic and the database file itself using algorithms up to 128-bit 3DES without too much loss of performance (in the range from 0.1 percent up to about 8 percent based on the platform and algorithm that's used).

In the long term it's possible that we'll see SQLBase and Velocis end up in one product—but most likely not within the next version or two.

Appliance computing

Last, let me sum up what we heard about the newest strategic sector of Centura: appliance computing. Because it's different from rest of the Centura products, it won't be handled only via the conventional channel. Right at the beginning of the conference Heinz Höreth announced **Ralf Gronkowski of Centura Germany** as the manager of the Centura Appliances Division (I'm not sure if this is the official name but at least it describes what it is).

There are two Centura products available that fit into this sector. One is the **Raima Database Manager (RDM)** and the other one is **eSnapp**.

As I already mentioned, RDM is a small, fast, robust database engine already used for years now as an embedded database solution by major companies. eSnapp is a connectivity solution for appliances like PDAs, handhelds, smartphones, and anything else that has some kind of computer in it and some kind of network connection (be it wired or wireless). What eSnapp gives you right now is a database connectivity API that's much like ODBC in the Windows world, which can be used to access any ODBC datasource via an eSnapp Application Server. We saw a demonstration of a WindowsCE Palmtop connection via eSnapp, via a wireless LAN to a SQLBase Server database. There was a press release during the last week announcing a cooperative agreement with SAP in Japan where Centura is extending the mySAP.com solutions based on this technology.

What was most interesting for me was that there were people around in the audience that came to Ralf directly after his presentation, telling him that they have current projects where eSnapp perfectly fits their needs.

Continues on page 12

Berlin Quote...

Sven O. Rimmelspacher

When I listened to Scott Broomfield's keynote speech, I was kind of happy, because he gave me the feeling that everything's on the right track. Almost everything he "promised" at the DevCon in Munich some two years ago, was fulfilled. There is no big euphoria (well, there is because of the stock price) but there are very good and serious visions.

The Matterhorn release seems to be a pretty good product and the so called wrecking-crew test they plan is a good option. The product portfolio is getting some kind of "round" now with the Raima products and the new department that deals with the fifth wave and appliance products. I'm happy that this is a separate department which does not influence the basic company directions too much (I hope).

Manage Form Windows Better with VisWinIsWindow()

Michael Lamon

I prefer to use form windows instead of modal dialog boxes; this allows users simultaneous access to multiple windows within applications. Due to the asynchronous nature of creating form windows, I often run into the problem of determining if a window has already been created or destroyed by the user.

The most common remedy for this problem is to test whether the window handle assigned to the form window from a previous call to `SalCreateWindow()` is equal to

`hWndNULL`. For example, to ensure that duplicate instances of the same form window can't be created from a menu item, try this:

SQLWindows

16 / 32

```
◆Popup Menu: File
  ◆Menu Item: &Search
    ◆Menu Settings
      ◇Enabled when: hWndSearch = hWndNULL
    ◆Menu Actions
      ◇Set hWndSearch =
        SalCreateWindow(frmSearch, hWndMDI)
```

Or perhaps in a more complex situation where the form window to test is determined dynamically at runtime, do this:

```
◆Pushbutton Class: CpbSearchNaturalLanguage
  ◆Instance Variables
    ◇Window Handle: hWndSearch
  ◆Functions
    ◆Function: GetSearchWindow
      ◆Returns
        ◇Window Handle:
  ◆Message Actions
    ◆On SAM_Click
      ◆If hWndSearch = hWndNULL
        ◇Set hWndSearch = ..GetSearchWindow()
      ◇!
      ◇Call SalSendMessage(hWndSearch, PM_DoSearch,
        MODE_NaturalLanguage, lParam)
    ◆If NOT SalIsWindowVisible(hWndSearch)
      ◇Call SalShowWindow(hWndSearch)
```

Is your code turning into spaghetti from having to reset window handles back to `hWndNULL` when the windows they represent are destroyed? Use this tip to wipe your window handle initialization mess off your plate.

This solution, however, has the drawback of assuming that the window handle to test (`hWndSearch` in the examples above) will be reset to `hWndNULL` when the form window in question is destroyed. In the menu item example, `hWndSearch` would have to be declared at a scoping level accessible to both

`frmSearch` and the menu item actions (which unfortunately in many cases is the global scope) so that on `SAM_Destroy` of `frmSearch`, `hWndSearch` could be reset to `hWndNULL`.

Resetting the window handle in the second example is more complex. I would have to move the local declaration of `hWndSearch` to the global variable section and ensure that all the possible windows returned by `GetSearchWindow()` will reset `hWndSearch` on `SAM_Destroy`. Or I would have to invent an involved notification scheme to let instances of `CpbSearchWindowNaturalLanguage` know when search windows are destroyed.

A much cleaner solution is to substitute the `hWndNULL` test on the window handle with a call to `VisWinIsWindow()`. `VisWinIsWindow(hWnd)` returns `TRUE` if the window handle `hWnd` is associated with a valid window that hasn't been destroyed; `FALSE`, otherwise. Note that `hWnd` doesn't have to equal `hWndNULL` for `VisWinIsWindow(hWnd)` to recognize that the form window associated with `hWnd` has been destroyed.

Using `VisWinIsWindow()` does away with the need to reset the window handle corresponding to a form window to `hWndNULL` when that form window is destroyed. The window handle can be declared locally, and no complex notification schemes are needed to ensure that the window handle is properly initialized. Here are the previous examples rewritten to use `VisWinIsWindow()`:

```

◆Popup Menu: File
◆Menu Item: &Search
◆Menu Settings
  ◇Enabled when: NOT VisWinIsWindow(hWndSearch)
◆Menu Actions
  ◇Set hWndSearch =
    SalCreateWindow(frmSearch, hWndMDI)

◆Pushbutton Class: CpbSearchNaturalLanguage
◆Instance Variables
  ◇Window Handle: hWndSearch
◆Functions
  ◆Function: GetSearchWindow
    ◆Returns
      ◇Window Handle:
◆Message Actions
  ◆On SAM_Click
    ◆ If NOT VisWinIsWindow(hWndSearch)
      ◇Set hWndSearch = ..GetSearchWindow()
    ◇!
    ◇Call SalSendMsg(hWndSearch, PM_DoSearch,
      MODE_NaturalLanguage, lParam)
    ◆If NOT SalIsWindowVisible(hWndSearch)
      ◇Call SalShowWindow(hWndSearch)

```

An alternative to `VisWinIsWindow()` is the function `SalGetWindowState(hWnd)`. This returns the CTD constant `Window_Invalid` if the form window associated with `hWnd` has been destroyed. I prefer `VisWinIsWindow()` myself because its syntax fits concisely with the functionality needed to solve the original problem.

This technique is so effective that I've stopped comparing window handles to `hWndNULL` in favor of using `VisWinIsWindow()` in all my code. **CP**

Michael Lamon is an independent consultant working in the Boston area. He's currently on contract with the first company to develop turnkey solutions in Centura Team Developer and SQLBase for independent schools. He can be reached at softinfo@parclata.com.

Centura Grabs Hold ...

Continued from page 10

As Ralf outlined in his presentation, the appliance computing sector is very important for Centura because of the tremendous increase in availability of such devices proposed for the next few years.

What features can we expect for the future? Right now, eSnapp provides connectivity to database applications. As everything moves into component-based development, the logical step forward is to provide connectivity to business components and that's what's planned for future versions. We'll see functionality that provides connectivity from appliances to COM components. Additionally, we'll see a notification functionality (from the eSnapp Application Server to the small device) and some features to better support offline operations (some kind of replication technology).

At the moment eSnapp is available on a limited basis for some kinds of pilot projects. General availability will be in Q1/2000.

Why I sound so upbeat

I hope I've provided the same kind of overview Centura offered during the conference. It might sound a little bit too positive. Sure, I would like to see some more things, especially in the tools sector, but I'm realistic about the resources. In my opinion, Centura is executing the right strategy and—with "Matterhorn"—will deliver a great tool for our needs. Let's hope the company removes the most annoying bugs that are now in 1.5.x—I'm sure it will do so! **CP**

Martin Knopp is a Senior Consultant at Pict & Byte GmbH, Vienna, Austria. Contact him at mknopp@byte.at or visit the Pict & Byte web site at www.byte.at.

Dynamic Tool Tips for Win98

Centura *Tip!*

Bill Grzanich—In the October, 1997 issue of *Centura Pro* you published "Dynamic Tool Tips," an article about implementing dynamic tool tips for a table class. I've successfully used this technique in my applications built with CTD 1.1.2 for Windows 95. Now, however, I have users attempting to use the app under Windows 98, and I've discovered that the call to the Windows API function `SendMessageTbGetMetrics` no longer works. Do you have any information on what I might need to change to provide this functionality under Windows 98? Thanks very much.

SAM User replies ...

All you need to do to make it work on Windows 98 is to change the export ordinal for `SendMessageTbGetMetrics` to the value of 503. Now you have to deal with another issue: Do you create and maintain separate Windows 95 and Windows 98 versions of your application, or add platform detection logic to your application? If you go with the latter case, then you could have two external functions declared under `user32.dll`: `SendMessageTbGetMetrics95` (with the originally published export ordinal) and `SendMessageTbGetMetrics98` (with the 503 export ordinal) and call whichever function is determined to be appropriate at runtime.