IUG TECHNICAL JOURNAL

THIRD QUARTER 2000

A PUBLICATION FOR USERS OF SYBASE, INC. PRODUCTS AND SERVICES

INSIDE

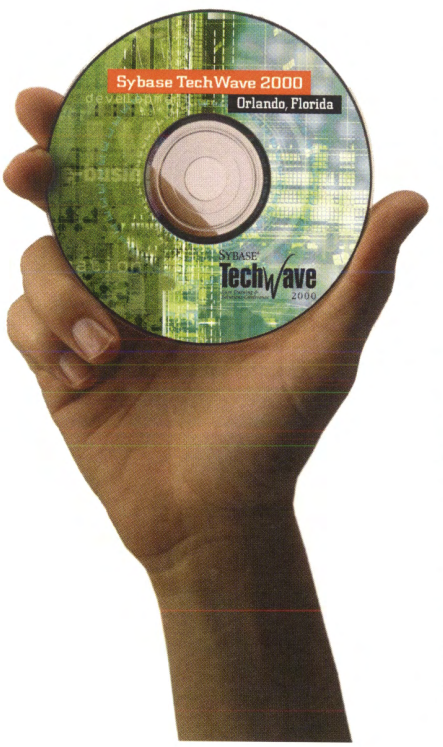
INTERNATIONAL

Sybase User Group

PRESIDENT'S MESSAGE

Dear ISUG Members,

The first half of the year has just flown by. I hope that as you read this, you are in attendance at or preparing to leave for the TechWave 2000 North America user conference taking place in Orlando, Florida. At this exciting event, you will be learning about the latest Sybase products, attending hard-hitting technical sessions, networking with your Sybase user colleagues, and hopefully having a truly wonderful time! And remember, if you can't make it to the conference this year, ISUG can provide you with a CD of the conference proceedings.



I am happy to say that Sybase has been doing some very exciting things so far this year! By now, everyone should be seeing the new corporate advertising campaign, and the press releases for the portal products. We on the ISUG board have been very busy as well, working to make your membership experience as rewarding as possible.

Last year, we introduced the ISUG CD containing several past issues of the *ISUG Technical Journal* and the public sessions from the TechWave 1999 North America Conference. We've put the logistics in place to do it again this year. Yes, we did hear from several people, and as a result, we will make the CD ISO-compliant so that it will also work on non-Intel and non-Windows-based machines.

ISUG is also well underway with our new membership campaign for 2000. Joining is easier than ever, as we have implemented a new membership application on the web with secured transactions for credit card processing. Come check out the site at www.isug.com and see what membership benefits ISUG has to offer.

We can also really use your help on this drive—please encourage friends and coworkers to join! Every new ISUG member gives us more clout with various divisions of Sybase. This allows us to provide you with the best service and benefits available to any members of a user group.

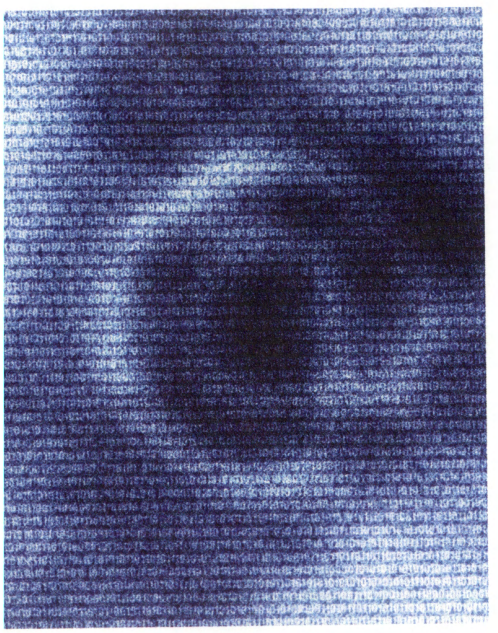
Be sure to stop by and visit us at the ISUG booth at TechWave 2000 this year. Also, Special Interest Group (SIG) meetings focusing on the major Sybase products will be held once again.

Additionally, ISUG will be announcing its annual recipient of the ISUG Award for Outstanding Achievement. This award goes to an individual, team, or organization using Sybase products (alone or with other products) to develop an application, function, tool or technique that can be emulated, utilized, or incorporated into another user's environment. If you know of anyone worthy of this award, please nominate them through the form provided on the ISUG website.

All the best,

Thom Lamb
ISUG President

Table of Contents



FEATURES

DEPARTMENTS

READER INFORMATION

ISUG Technical Journal

ISUG Technical Journal Director
Teresa Larson
Vetcentric.com
410.571.6790
tlarson@vetcentric.com

Managing Editor
Mary Freeman, Freeman Communications
510.525.4863
510.528.6958 Fax
Mary_Freeman@compuserve.com

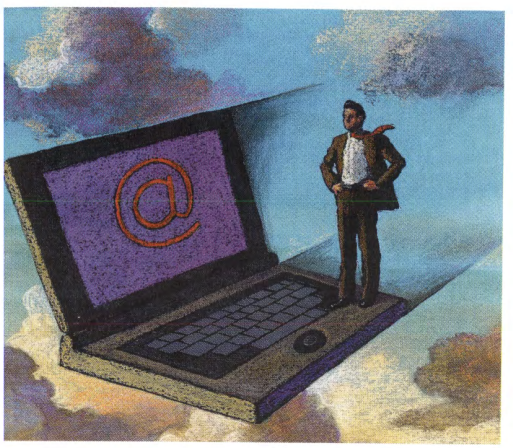
Art Direction
Jerry Jager, JagerCreative
jagercreativ@earthlink.net

Senior Technical Editor
Al Huntley, EDS

Technical Editors
Mark Dirrim, Vantive Corporation
Thomas Lamb, Automated Data Sciences
David Straiton, Sybase, Inc.

Managing Sybase SQL Server Using Your Web Browser

By Ed Barlow



Using perl/CGI scripts for database administration

Ed Barlow is a Sybase consultant who runs the Sybase Shareware site at www.edbarlow.com. He can be reached at sqltech@tiac.net

This article describes the use of perl/CGI scripts to allow administrators to access remote error logs and to manage database servers using any standard web browser. This approach gives production administrators functionality they do not normally have, but which is necessary in large production environments. The solution is secure, easy to create and deploy, and allows you to automatically access information that normally is retrieved using a manual process like telnet. This article describes the web browser/web server architecture, gives examples of perl/CGI scripts, gives an example of a useful tool to look at error logs, and introduces the free Webify.pm perl library.

Browsers are designed to serve information on remote servers, so it is a simple matter to envision managing your Sybase ASE servers through your web browser. There are several advantages to a web browser/web server architecture. Server-side scripts run on the same machines that hold your error logs and backup files, the machines where the database processing is actually occurring. The only normal interactive interface to this information on UNIX is the command line, which does not lend itself to automation; so this approach gives you information that is difficult to access normally. In fact, on Windows NT, you cannot access host information remotely (such as the output of task manager) although you can trivially access files (NT's "rsh" command and telnet facilities are not worth bothering with).

The web is also transparent. You can look at information that exists in multiple servers simply by changing your URL to point to another address. This allows a common interface that can access all your servers. Additionally, the web is open, so you can permit universal access to tools and real-time system-state information without any client configuration. You can allow the appropriate users to manage your servers, see if your nightly batches ran successfully, and even restart your servers from home through a secure connection (after some security configuration). You can also give your end users simple screens to check information they might require (like batch job status or server availability and performance). Finally, the web is easy to program yourself using perl/CGI scripts.

Both perl/CGI and Java can be used to serve information to your browser. For simple screens like those for systems administration, I recommend perl/CGI over Java. Java allows users to create a full-featured user interface, but programming it is an order of magnitude more complicated. This complexity does not necessarily have any benefit for the administrator. This article concentrates on how to use simple perl/CGI scripts to manage your servers.

Prior to writing perl/CGI screens to manage my systems, I kept telnet sessions open to all my machines. This left my desktop cluttered with unused windows. I only looked at error logs when there was a problem and I found it difficult to be proactive in managing my systems.

I now have a web-based morning review report which checks logs (error and backup logs) and over 100 potential database errors ranging from space warnings to security violations. This allows proactive systems management.

Architecture

A web server is daemon software (software running without a console) that handles requests from browsers. The protocol for a request is very simple, but beyond the scope of this document. Essentially, it is a server that listens on a socket and returns information when you send it a query. This is exactly like the Sybase data server, only the protocol is not TDS. The most popular web server is the free Apache server, which can be downloaded from www.apache.org. Requests to web servers are in the form of a URL, and the data being returned to the client is generally in the form of HTML, which the browser displays. Most of you will be familiar with the HTML markup language that provides basic page formatting. If you have never seen it, bring up a web page and **view -> page source**.

The server maps the URL into a file, which is either executed or simply returned. If the file is a script to be executed, it is known as a CGI script, and it must return a valid page of HTML text in standard output and nothing in standard error. The administrator defines a set of mappings for the web server to translate virtual directories to real directories. So, when your browser submits the URL http://www.edbarlow.com/links/link.pl, the following steps are performed:

1. The browser parses the URL into a machine name, port, and path.
2. A DNS lookup on edbarlow.com is performed to find the IP address and port number of the target web server.
3. The network routes the request to the appropriate machine and port (the default is port 80).
4. The web server get the request and translates the path into a directory (/links) and program (link.pl).
5. The virtual directory /links is translated (say, into /usr/local/apache/links) based on the web server configuration.
6. The server decides whether the target file (link.pl) is executable based on web server directory options and file permissions.
7. The program link.pl is executed and standard output is returned to the browser.

Perl is the most common scripting language for dynamic web pages and is freely available from www.perl.org. An NT version is available on www.activestate.com. Installation of both perl and Apache on NT is very easy (taking about 15 minutes after download). On UNIX, you can download precompiled binaries or source code which you must compile yourself. Expect to spend an hour or two setting up perl and Apache on a UNIX system. You do not need root permissions to install private versions of either of these packages on UNIX, but you will require root to install them in the default location (/usr/local) where everyone can access them.

The only complexity in installation is the set-up of sybperl, the perl library for Sybase. Sybperl comes with dblib and ctlib modules and can be found on http://www.mbay.net/~mpeppler. If you are using Activestate Perl for NT, use their ppm.pl utility to download sybperl. If you are on UNIX, you will need to download and compile it from www.cpan.com (of course you will not want to run the web server as a root-owned processor).

Perl and CGI Scripting to Manage Sybase

Perl is a simple scripting language that looks like a combination of C, the shell language, and sed. It is commonly used for simple administrative scripts because it is easy to learn and program, is robust and full-featured, and is very strong for string manipulation. I believe perl is the perfect language for DBAs.

The concept is to use perl/CGI to provide a user interface to simple perl programs that perform administration tasks. Breaking up the tasks this way makes it easier to debug your scripts. For example, you may write a program to filter your Sybase error log for serious errors, and then use a second script that creates a web screen to call this program. Additionally, many organizations already have perl scripts to perform Sybase tasks. Integration of these tasks into a web page gives them more usability, as users don't need to remember command line options or what the scripts do.

The CGI library that is built into perl is known as CGI.pm. This library provides many features necessary to write CGI scripts. Most HTML tags have simple corresponding functions—for example, the function **p()** returns the string **<p>** which corresponds to a paragraph in HTML. The function **h1(string)** returns **<h1>string</h1>** which indicates a first level heading.

A Simple CGI Example

The devil is in the details, so a simple example is in order. Here is a sample CGI screen:

```
use CGI qw/:standard/;          # load standard CGI routines
print header,                   # create the HTTP header
  start_html('hello world'),    # start the HTML
  h1('Heading'),                # level 1 header
  "Some Web Page Text",         # some normal text on the page
end_html;                       # end the HTML
```

This simple program will print a web page titled "hello world" with heading of "Heading" followed by the string "Some Web Page Text."

A More Complex Example

We will now look at an example in which a form calls an external program, specifically, a screen that allows us to look at the ASE error log. First, we write a basic program named filter_errorlog.pl that filters an ASE error log for errors. We then write a CGI program that provides a user interface for our program. Below is our filter_errorlog.pl program. (Note that the opening directive is for the Apache server; this example is from my own implementation. Otherwise, the hashbang should read **#!/usr/local/bin/perl**.)

```
#!C:\Perl\bin\perl.exe
# filter_errorlog.pl - a VERY basic filter for Sybase ASE error logs

# standard command line options
use    Getopt::Std;

# variables for options
use    vars qw($opt_i $opt_f $opt_t);

# check parameters
die "Usage: filter_errorlog.pl -i infile [-ft]" unless getopts('i:ft');
die "Must Pass Input File with -i\n" unless defined $opt_i;
die "File $opt_i does not exist\n" unless -e $opt_i;

# Get Current Datetime into variables and reformat to
# yyyy/mm/dd
my($sec,$min,$hr,$mday,$mon,$yr,$wday,$yday,$isdst)=localtime(time);

# year comes back offset by 1900
$yr+=1900;

# mon comes back 0-11 so add 1
$mon++;

# ensure 2 digits
$mon="0".$mon  if length($mon) == 1;
$wday="0".$wday if length($wday) == 1;
# make the string
my($datestring)="$yr/$mon/$mday";

# MAIN SECTION: open and read file, printing if necessary
open( INPUT, $opt_i ) or die "Cant Read File $opt_i : $!\n";
while (<INPUT>) {
        next unless ! defined $opt_f
                or /error:/i
                or /Msg:/i or /corrupt/i;
        next if defined $opt_t and ! /$datestring/;
        print;
}
close(INPUT);
```

The above program introduces some new concepts, but is quite functional for 18 lines of code. You pass the log file name with the -i option, the -f option indicates that you wish to only see errors, and -t indicates that you only wish messages generated today. The first section of this code uses the Getopt::Std library to parse command line options and store them in the variables $opt_i, $opt_f, and $opt_t. We then use the **localtime( )** function to parse the current date into some variables. These variables are reformatted to conform to the yyyy/mm/dd format in our errorlog. The months variable ($mon) is returned as 0-11, so we add 1 and prepend a 0 if necessary to get 01-12 (and prepend a 0 to day variable if it's one character as well).

The main section opens our log file and will print lines that match our parameters, even when the server is running. If -f was passed, we ignore lines unless they contain the strings **error:**, **msg:**, or **corrupt** (case insensitive). This is not sufficient for a full-featured log browser, but gives a good first cut. A more robust error log viewer is more complicated and can be downloaded with my perl utilities from http://www.edbarlow.com. If -t was passed, we ignore lines unless they contain our $datestring. In a real-world implementation, of course, you would limit your return set. As you can see, this program is simple and short. The next program is our CGI script to create a basic form to interface with filter_errorlog.

```
#!C:\Perl\bin/perl.exe
use    CGI qw/:standard -no_debug/;

print header(), start_html('Errorlog Viewer'),
        h1('System Errorlog'),
        start_form(),
        "Show Errors Only", radio_group(-name=>'errors',-
            values=>['no','yes']),br,"\n",
        "Today's Messages Only", radio_group(-name=>'today',-
            values=>['no','yes']),br,"\n",
        submit('View Error Log'),"\n";

if ( defined param() ) {
        print hr;

        $command="C:/perl/bin/perl.exe ";
        $command.= "filter_errorlog.pl -i\"C:/sybase/install/
            errorlog\"";
        $command.= " -e " if param('errors') eq "yes";
        $command.= " -t " if param('today')  eq "yes";

        # Run the command using perl's open $command.= " |";
        open(X,$command) or die "Cant Run Command: $!\n";
        while (<X>) { print $_,br; }
        close(X);

# You could also run the command using system() as follows:
        # print "<PRE>";
        # system ( $command );
        # print "</PRE>";
}
print hr,end_form(), end_html();
```

This script is broken into two parts. The first part prints a simple form with a title of "System Errorlog" and two radio buttons that allow you to select if you wish to show errors or to see todays messages. We know the form's button was pushed if the function **param( )** returns something. If so, we create a command string for our filter_errorlog.pl program. We can then run it either using the perl **open( )** syntax or using the **system( )** function (commented out). The function hr prints **<hr>**, which represents a horizontal rule in HTML. Because we're just getting input from radio buttons and building a command line from them, there are no security issues.

Applications of This Approach

You can use this approach to view error/log files, to run application queries, or to monitor your systems and data transfers. As mentioned earlier, I believe that the approach works best when you write external programs to do the work, and write a second script to handle the user interface. I find this architecture cleaner and easier to understand and debug. Use either **system( )** or **open( )** call to execute the external program that does the real work (see example). I have written CGI programs to generate web screens that will:

- View and filter error logs for common errors
- Monitor and restart nightly batches (data transfers, batch computation)
- Execute and monitor nightly backups
- Monitor and debug systems
- System reporting (including cross server reporting)
- Server performance tuning
- Monitor replication processing
- Monitor Open Servers and other application-specific log information
- Generate a morning review report containing anything suspicious found in the server, system, or log files.

When you are managing a number of servers, writing screens to perform these functions is often preferable to a command line interface. These jobs must be run on the systems that are running your data servers, which either means keeping telnet sessions open to all your servers or using web screens. While there are expensive enterprise monitoring systems that can be purchased, simple web screens give me clean access to any of the above features, no matter what the company requirements, political structure, or environment.

Additionally, I have written screens that do more common operations like:

- Hypertext linked DDL viewer
- Cross-server database comparison
- Backup and restores
- Cross-server reporting
- DDL interdependency reporting
- Performance and tuning using showplan
- Cross-server real-time monitoring
- Easy access to Sybase auditing
- Server to server DDL and data copying

Security

Security is of crucial importance to any web-based system because, by nature, the network is open and insecure. This is especially true if you open access to your applications to the public Internet.

There are several levels of security that you can build into your applications using perl/CGI scripting. The most restrictive is the hardware security built into your firewall, which can be used to restrict access by IP address. Most applications you write should not be accessible from outside your company (this is the normal default for most companies). If you wish to access these systems from home, you will need to contact your firewall administrator and have them add your IP address as a legitimate client for your web server. The same security can be configured in your web server software to prevent/allow access from an IP address by virtual directory (not dynamic IP addresses, however).

Additional security is provided by requiring users to enter a login and password. Because CGI is a stateless protocol, there is, however, a problem tracking login information between screens. A stateless protocol means that each screen has no explicit relationship to the next or prior screen. You must explicitly pass information between forms using form fields or URL parameters. For security reasons, you would not normally put the login and password in the URL, so you lose information about which login was used (and whether it was successful) when you draw a new screen. You can't keep track of this information in a form field because it would be insecure.

There are two good ways to address this situation. The first is to use a cookie and to keep track of what login was used. Cookies are valid until they expire, so it is recommended that you expire your cookies after two or so hours, or anyone who has access to the users' workstation will be able to access your web screens (this might or might not be a problem). The second approach is to keep track of the client's login information by using their IP address as a key. The client IP address is available in the environment variable REMOTE_ADDR, which is accessed in perl as $ENV{"REMOTE_ADDR"}. As with the cookie, you would need to expire logins after a period of time.

One key concept for security on these types of systems is to not permit dangerous activity. It is one thing to allow you to dump your databases, and another to allow users to load them. A final approach to security is to keep your system "open" (no password required), but to require an password for any "secure" operations. Simply ask for a login and password right before you perform each secure operation.

Other Notes

Web Serving: Of course, when you use this type of system, you will need to install a web server. This means that your system will be slightly slower. The web server will not, however, be used very often and this should not have any measurable impact on system performance on either Windows NT or on a more powerful UNIX system. Remember that these screens are not heavily used.

Timeouts: Long running operations like dbcc may time out your browser connection. The solution is to run your long-running job in the background, redirect its output to a file, and then to poll the file with your CGI form. Use the function **header(-refresh=>'10')** instead of **header( )** in your CGI.pm forms if you wish them to automatically refresh after 10 seconds. You can run jobs in the background using **system( )**, **open( )**, or **fork( ) / exec( )**.

Redirecting Standard Error: If you write separate CGI utilities and web screens, be sure that your CGI utilities do not print to standard error. Most web servers redirect standard error to their error logs, which is probably not what you want. Also be sure that output is not buffered. The command **$|=1;** in perl will unbuffer standard output. I often add a -h option to these command line scripts, which will cause them to output HTML for my web screens.

Webify.PM

I have written a free perl library to help create web based systems, which is distributed free at http://www.edbarlow.com. The Webify library provides a unified look and feel, navigation, database access, and system security. It does all the common tasks required to run a web screen, allowing each screen to be built easily. Each screen requires four functions, representing the differences between screens. All common functions (like the title or page background) are automatically performed by the library, which gets the data from a configuration file. Forms that use the Webify library have the following general flow (pseudo code):

```
form_initialize()
check security information and goto login_form() if > 2Hours since login
if( button was pushed )
  if( validate_button_action() )          # form validation
       $cmd = button_action_override()
       if( $cmd = "" ) get command from configuration file,
       run command and print output in special format
print page heading and navigation bar
print @err array ( error messages from system or form validation )
paint_form()                              # information specific to the form
print page end information
```

This flow is common to all forms. The four user-defined functions represent the minimum functionality required to customize for each screen. The most important, and the only one that is required to contain code, is **paint_form( )**, which draws the body of the screen. The other functions are optional and will initialize, run local processing, and check for data entry errors on your form.

Each form script will initialize a few variables (form title), define the functions **form_initialize( )**, **button_action_override( )**, **validate_button_action( )**, and **paint_form( )** and will call **new_form( )**. Webify.PM provides application security, a common look and feel, inter-screen navigation, and the API to external programs. Screens, from which you can do things like view log files and view nightly batch results, can be programmed in under 100 lines of code.

The code to the right is a Webify script that will manage our errorlogs as per our second (more complex) sample above. The main difference between the following code and the earlier code is that the Webify code is part of a system with navigation buttons on top, built in login security, and a common look and feel.

```
use CGI qw(-no_debug :standard);
use Webify;
use vars qw($debug %CONFIG @err );
$CONFIG{"title"}="Errorlog Viewer";

sub button_action_override {
  my($command)="filter_errorlog.pl -i\"".$CONFIG{"SYBASE"}."/install/errorlog\"";
  $command.= " -e " if param('errors') eq "yes";
  $command.= " -t " if param('today') eq "yes";
  return $command;
}
sub validate_button_action { return 1; }
sub paint_form {
  print
    "Show Errors Only", radio_group(-name=>'errors',-values=>['no','yes']),br,"\n",
    "Today's Messages Only", radio_group(-name=>'today',-values=>['no','yes']),br,"\n",
  submit_button('View Error Log');
}
sub form_initialize { }
new_form();
```

Typical Webify forms range from 200 to 400 lines of code. In under 200 lines of Webify code, you can write a screen that allows users to view all log files on a system (Sybase Server, backup server, open server, data transfer, web server, and system logs). Of course, if you wish to filter the log files, you will need a separate program. The interface between your screens and this separate filter program is just a line in a configuration file. Installation and setup of a system of this sort is not overly complicated, but does involve some work to define what your screens should look like and to setup your configuration files.

Conclusion

I have used Webify.pm to design two systems. One manages a fairly complex system that manages data transfers and open server. The system provides a unified user interface to operations that occur on multiple machines. The other is Webmonitor, an enterprise Sybase management system. Webmonitor provides a fairly nice interface to my free perl utility toolkit, a set of 50 simple Sybase administration scripts. I have also written a perl/CGI-based system that runs basic data collection (sar/iostat/vmstat) and converts it into thumbnail performance graphs to allow the administrator to view UNIX performance across the day.

This perl/CGI approach is a good way to replace the existing ASCII menu systems that many companies still use as an interface for their operations staff.

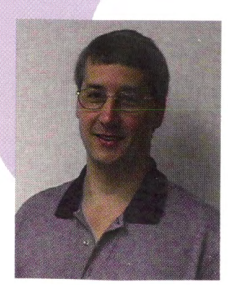
Some companies don't even have this level of automation, relying on operators to know UNIX commands in order to perform their jobs. With very little effort, a web-based system can be designed that removes the need for any ASCII menu systems. Additionally, this approach can give system monitoring and performance information to DBAs and simple screens with the output of sp_who and sp_lock can be written for end users. Since a simple screen can be usually written with 50 lines of code and complex screens with under 200 lines, perl/CGI provides a good solution to many DBA problems.

For additional information, check out the CGI.pm manual pages (available with perl). You can also find additional documentation, tools, and information on the Sybase shareware site at http://www.edbarlow.com. ■

Six Steps to Integrating Jaguar, PowerBuilder, and Power Dynamo

By Jay Hunt

Providing a roadmap for implementing EA Server using Jaguar and Power Dynamo



Jay Hunt is a Principal Architect with Automated Data Sciences. He can be reached at djayhunt@charter.net

Early this year, I began to take advantage of the power of EA Server and Jaguar CTS. Yet I found that no roadmap really existed for implementing EA Server to work in the HTML world. This article provides such a roadmap, focusing on the use of EA Studio 3.0 tools—including PowerBuilder 7.0, PowerDynamo 3.0, and Jaguar CTS 3.0—to help prepare users for EA Server development.

After the initial EA Studio 3.0 components have been installed, I recommend obtaining and installing the EA Studio Maintenance Release 3.0.1. At this time, this is only available on CD directly from Sybase. When applying a maintenance release, it is important to note that the products in EA Studio are somewhat integrated. Therefore, applying maintenance to one product without applying maintenance to another product could cause trouble.

Step 1: Our First Website

Our journey into the HTML/EA Server world begins with developing a simple website directly in PowerDynamo. Use the short cut that exists under the Sybase menu to start Sybase Central (Figure 1).

A PowerDynamo website is normally stored in an Adaptive Server Anywhere Database, so we will need to create one. This can be accomplished by clicking on the "Utilities" child of the "Adaptive Server Anywhere" treeview leaf. On the right side of the display, click on "Create Database". Then choose an appropriate name and storage location for your database. From this point, you can accept the defaults that are presented, and, when you reach the "Finish" dialog box, press the finish button.

We should now create an ODBC profile for this database. This can also be done under utilities of Adaptive Server Anywhere. Activate the ODBC Administrator; there will be several tabs, but for now leave the User DSN tab active and press Add. (Figure 2).

Select Adaptive Server Anywhere 6.0 and press Finish. An ODBC Configuration dialog box will appear. On the first tab, fill in the data-

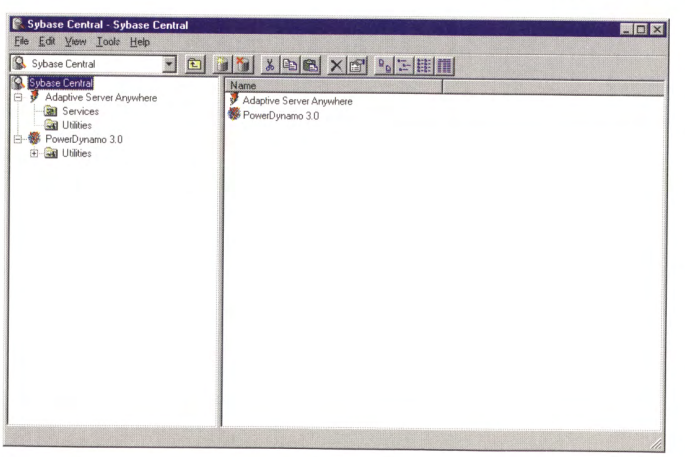


Figure 1

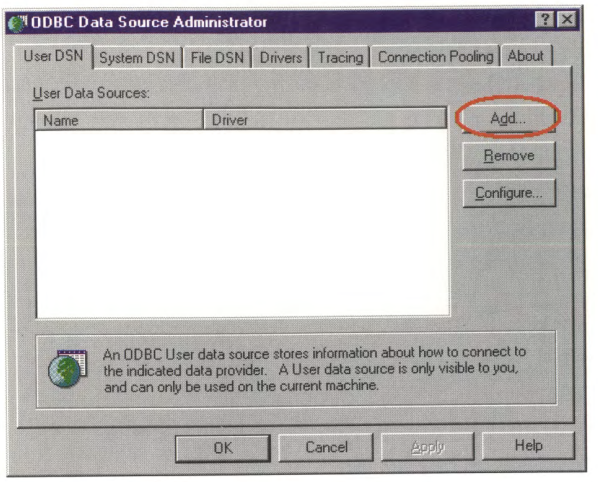


Figure 2

source name. Next on the Login tab, fill in *DBA* for User ID and *SQL* for Password. On the Database tab, you will find a browse button. Click it and navigate to the appropriate location of the database just created. Once the new database has been located, highlight it and click OPEN to return to the ODBC Configuration dialog. At this stage, enough information exists to create the ODBC data source, so click OK.

Step 2: Connection Profiles

We should now be back in Sybase Central. On the menu is the "Tools" option: Activate it and select "Connection Profiles..." A connection profile is a quick way to remember connection info for Sybase Central. Something similar to Figure 3 should be found. Click on "New".

A new dialog box appears. Type in a name (I will use *ISUG*) and make sure to change the dropdown to "PowerDynamo 3.0." Click on OK, and the Connection Profile dialog box appears. The connection type should be ODBC, while the ODBC datasource should match the name of the newly created ODBC datasource from earlier. Find it in

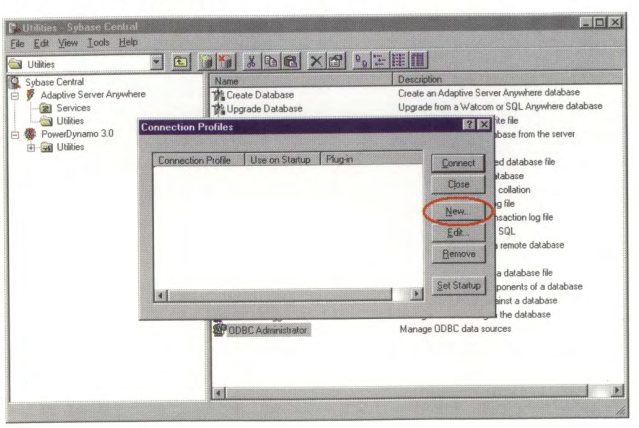


Figure 3

the drop down list and select it. Use DBA and SQL for User ID and Password respectively. Press the OK button and you return to Figure 4. I suggest pressing the Set Startup button while the new connection profile is highlighted. This will activate the selected profile automatically whenever Sybase Central is started.

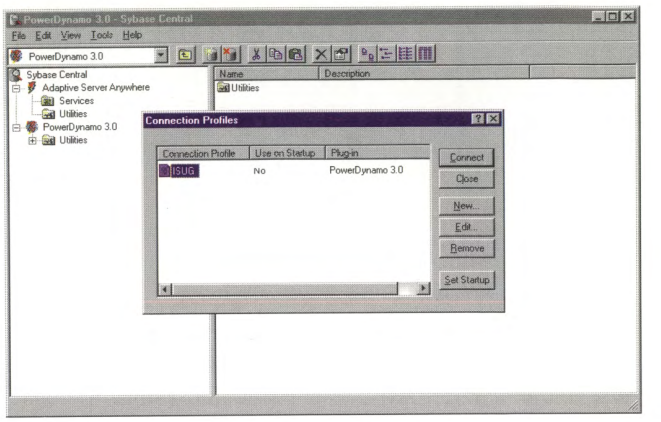


Figure 4

We are now ready to create our website, so press the Connect button. A message box will appear like the one in Figure 5; press Yes. Another dialog box appears, Figure 6. The Root folder name can be changed; but for now, let's leave it as *Site*. You also have the option of adding PowerDynamo HTML documentation and a sample application. It would be nice to review this information, but to keep things simple, let's not check these boxes.

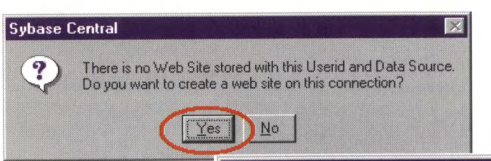


Figure 5

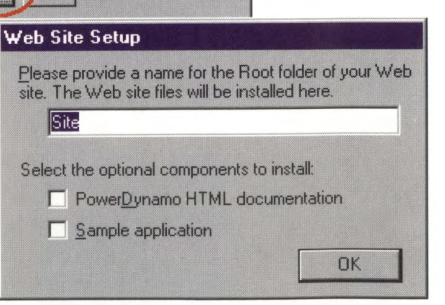


Figure 6

Files are automatically added and the website is created, but where is the data? In Sybase Central, we should see the *ISUG* connection profile (or whatever you called it) under the PowerDynamo leaf of the treeview. Expand the treeview so that it looks like Figure 7.

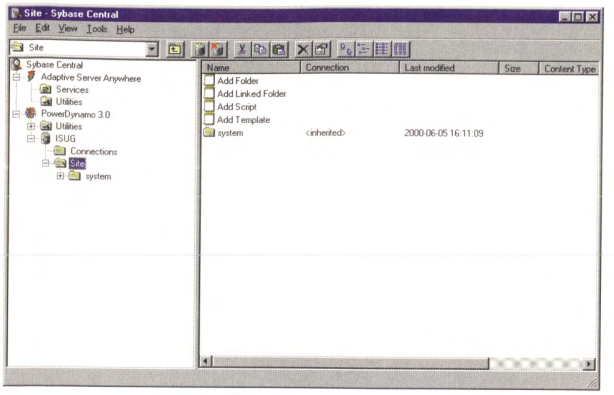


Figure 7

Step 3: Adding Web Pages

Our website is now set up and ready for use, but we don't have any pages to look at. To add a web page, double-click on the "Add Template" entry in the Name column on the right side of the display. You will be prompted for a name, so type **page1.htm** and press Next. If you feel the need, or if it is required by internal standards, type in a proper description. The next screen is the New Template Default Connection dialog box. This information is used if the web page is going to make database calls.

Leave the connection set at Inherited and press Next. Leave the Add SQL Query to New Template blank and press the Next button. All that is left is two more wizard screens. The first one is the document type. We are in the process of creating an HTML document, so leave the info as is and press the Next button. When the wizard screen appears, press the Finish button and our new page is created (Figure 8).

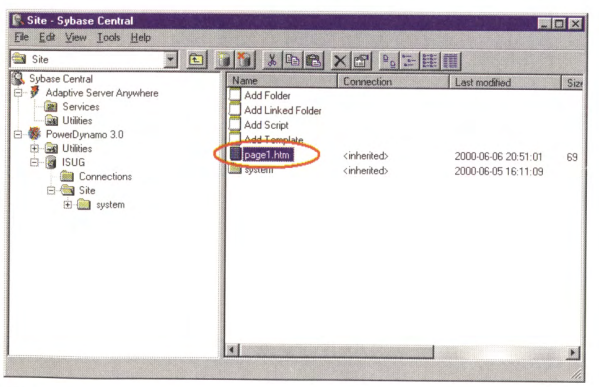


Figure 8

Double-click on Page1.htm and add the code from Figure 9.

Close the object by selecting File Close or clicking on the "X". We should be back in Sybase Central, where we can right click on Page1.htm and select Browse Output to see the result displayed in Figure 10.

Note that PowerDynamo is a very powerful tool that can be used by itself for database access. To find out more about PowerDynamo, I recommend reading previous *ISUG Technical Journal* articles by Thom Lamb or attending a PowerDynamo session at TechWave 2000 in Orlando. Thom and I will be presenting on Wednesday afternoon in IA337, "A Case Study on The Use of PowerDynamo to Build a Database-Driven Website."

```
<HTML>
<TITLE>page1.htm</TITLE>
<BODY>
<H1>ISUG Example</H1>
<P>This is my first PowerDynamo Page</P>
<!--Script
   d_date = new Date
   s_msg = "<P>Today is "
   s_msg+= "<strong>" + d_date.toLocaleString() + "</strong>"
   s_msg+= "</P>"

   document.writeln(s_msg)

-->
</BODY>
</HTML>
```

Figure 9

Step 4: PowerBuilder Prep Using Jaguar

We are now ready to move into the use of a Jaguar component created from a PowerBuilder NVO. This example will be very basic and serve no practical function except to demonstrate the process involved.

In PowerDynamo, we displayed the time using Dynascript. Let's repeat the process, but this time let's get the information from a Jaguar component. To do this, we will need to create a new application, build a Jaguar component, deploy it to Jaguar, create some Java stubs, and call the component from PowerDynamo. A lot of these steps can be found in the

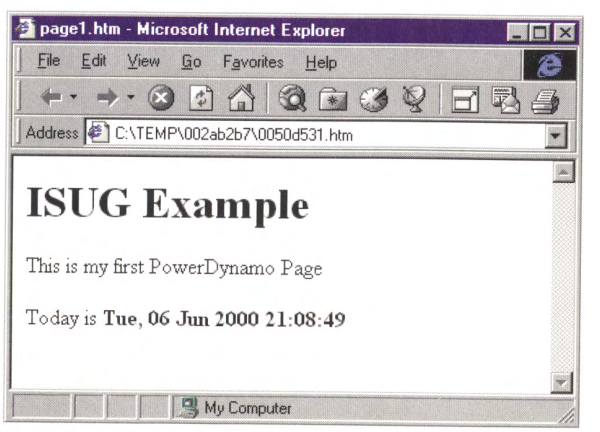


Figure 10

"Jaguar Component" wizard found on the New Tab for the Select Application Dialog box.

This dialog can be accessed by selecting File and then Select Application. Select the Jaguar Component icon and press OK. When the Information screens are displayed, press Next, until the Specify New Application and Library dialog prompts for the application name and location. I will use *ISUGEXAMPLE* for the application name.

Because we have just one pibble in our application, we can press the next button to continue on to the Specify New Component dialog. This is the area where we specify the name of the component we are to create—that is, the name of the NVO into which we will place code. Leave this as the default. The next screen, Specify Jaguar Component Name, may seem confusing because it presents the same NVO name that was entered in the previous screen. This is the name that we or others will refer to when we need to make a method call through Jaguar. Let's remove the "n_" so we should have something that looks like Figure 11.

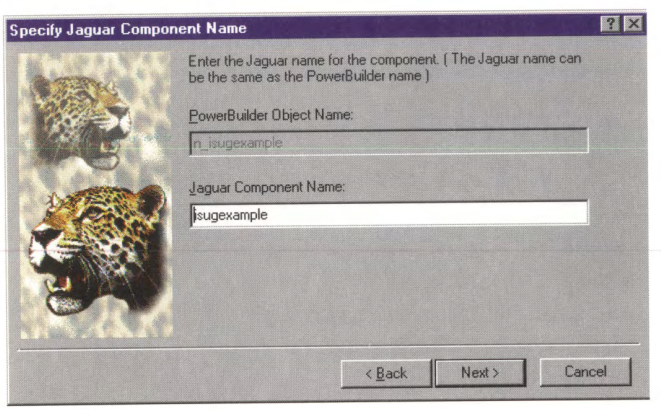


Figure 11

Jaguar Server Information

The next screen, Figure 12, allows us to specify information about the Jaguar server. By default, the Jaguar server is installed with a server name of localhost on port 9000. This can be changed and will be addressed in future articles. The default password is *jagadmin* and there is no password. Make the appropriate change and press Next, which opens the Specify Package Name. The package, which I have called *isugpackage*, is a logical collection of related components for my *ISUG* examples.

When the information is entered and the Next button is pressed, we are asked what type of component this will be. Leave the default as Standard Component, and press Next again. We are now at the Specify Instance Pooling Options wizard window. This example will support instance pooling, so all we have to do is click on the next button. The example

Figure 12

will not support transactions; we have no need to rollback or commit a batch of information, but we will check the Auto Demarcation/Deactivation box. By doing so, we will be activating the component with each method call we make.

Jaguar components have two new events, Activate and Deactivate; so if you have instance variables you plan to use in multiple method calls (yes, this can be done) and you have the Demarcation/Deactivation box checked, be careful not to re-initialize the variables in the activate event. A later article will cover the next five wizard pages, but for now just accept the defaults and complete the wizard. When we are done, there should be three objects (Figure 13).

Open the NVO *n_isugexample* and create a new function called **getdatetime** that returns a string. Add the code shown in Figure 14. We must now get the code/NVO into Jaguar. This is accomplished by opening and running the project p_isugexample. When this is done, we can close PowerBuilder and start the Jaguar Manager.

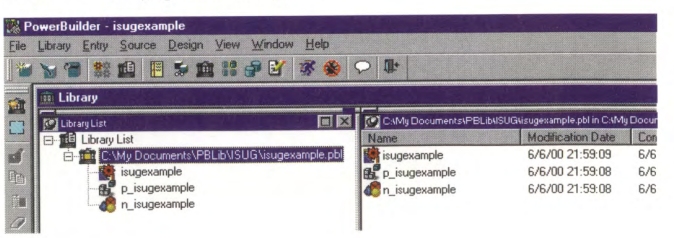


Figure 13

Step 5: Connecting to the Jaguar Server

If when Jaguar was installed, you did not check the box marked "Install Jaguar as an NT Service," you will need to click on the "Jaguar server" shortcut first. This will start the Jaguar server; otherwise, it should already be running. One way to check if the server is running is to open the Control Panel and double-click the Services icon.

It is now time to open the Jaguar Manager to attend to

```
String ls_datetime

ls_datetime = String(Today(), "ddd mmm dd yyyy")
ls_datetime+= " " + String(now(), "h:mm:ss am/pm")

Return ls_datetime
```

Figure 14

some details before we put everything together. We will need to connect to the Jaguar server, so click on Tools, Connect, Jaguar Manager. Fill in the user name as *jagadmin* and the Hostname as *localhost* (Figure 15). Open the installed packages folder under Servers, right-click on our package, isugpackage, and select Generate Stub/Skeleton (Figure 16).

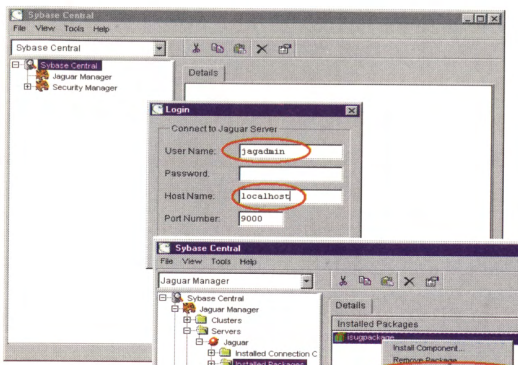


Figure 15

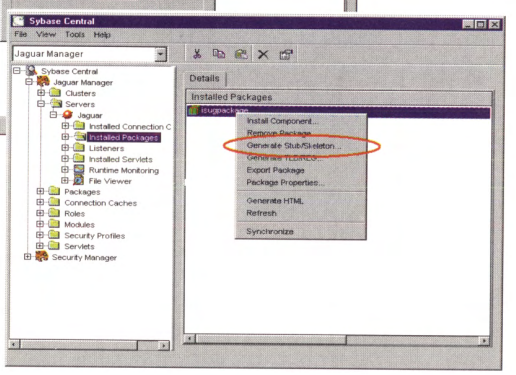


Figure 16

The communication between Jaguar and PowerDynamo is accomplished by making Java component calls. This process of generating skeletons and stubs creates Java source code that has the necessary information to communicate with the PowerBuilder NVO we deployed to Jaguar. The process of generating these Java stubs is only required when new methods are created on PowerBuilder NVO or the prototype of any method changes. Otherwise, this has to be done just once.

There are two boxes to check: Generate Stubs and Generate Java Stubs. Then, click on the Generate button to create the Java source code. At this point, it is necessary to open a command prompt to compile the new Java stubs. I recommend checking the classpath environment variable, because the default installation of EA Studio 3.0 sets this up wrong. The correct classpath statement should look like this:

```
c:\Program Files\Sybase\Shared\PowerBuilder\classes.zip;
c:\Program Files\Sybase\Shared\Sun\jdk118\lib\classes.zip;
c:\Program Files\Sybase\Jaguar CTS 3.0\html\classes;.
```

Under the default implementation of EA Studio, the Java stubs are created in C:\Program Files\Sybase\Jaguar CTS 3.0\html\classes\isugpackage.

Notice that this directory is the name of the package we are working with. In our example, there should be four Java files. We need to compile these files by typing **Javac *.java**. It has been my experience that if errors occur during the compile, the classpath is probably wrong.

Step 6: Assembling It All

It is now time to put everything together and add the Jaguar component call to the HTML document. Open the Sybase Central and PowerDynamo, and modify the code as shown in Figure 17. After we change the source code, we can close, save the page, and then browse the output.

```
<HTML>
<TITLE>page1.htm</TITLE>
<BODY>
<H1>ISUG Example</H1>
<P>This is my first PowerDynamo Page</P>
<!--Script

   s_comp = "isugpackage/isugexample";
   s_jag = "iiop://localhost:9000";
   s_uid = "jagadmin";
   s_pw = "";
   jagcomp = java.CreateComponent(s_comp, s_jag, s_uid, s_pw);

   s_msg = jagcomp.getdatetime()
   document.writeln(s_msg)

-->
</BODY>
</HTML>
```

Figure 17

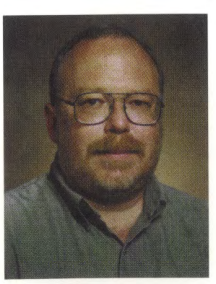
Conclusion

Although our first Jaguar / HTML communication example is not very powerful, we will build upon it in the next article and talk about the HTML Datawindow and building a HTML framework. With this knowledge we can prepare for moving forward with HTML development based upon our PowerBuilder experience and the Datawindow. ■

Where, Why, and How To Add or Modify Optimizer Statistics

By Eric Miner

Supporting better decision-making based on optimizer statistics



Eric Miner has been with Sybase since 1992, working with the optimizer team in engineering, as well as focusing on optimizer issues for technical and product support engineering. He can be reached at eric.miner@sybase.com.

In previous issues of this journal, we've discussed the new optimizer statistics introduced in ASE 11.9.2 and how to read, write, and simulate them using optdiag. This article discusses why, where, and how you might want to add or change optimizer statistics. These statistics are used by the optimizer to estimate the most efficient way to access the data required by a query. They are the only information the optimizer has about your tables, indexes, and data. This article will provide information you can use to help make decisions on how to take advantage of the powerful and flexible changes to optimizer statistics in ASE 11.9.2 and above.

ASE's Optimizer Statistics

As of ASE 11.9.2 the optimizer statistics are stored in two system tables. *Sysstatistics* stores the column level statistics that were previously stored in the single distribution page (a.k.a., the distribution statistics). The date and time of the last modification of the statistics are stored here. The statistics stored in sysstatistics can be written directly, as discussed below.

Systabstats contains the table/index level statistics such as row count, page count, index height, and the cluster ratios. These statistics are maintained dynamically; changes are first applied in memory and then flushed to systabstats. It's this regular overwriting of the table/index level statistics that makes it futile to attempt to modify them. In fact, optdiag will not allow you to input a file with modifications to the table/index statistics.

Before we continue, keep in mind that adding and/or modifying statistics is not absolutely necessary. However, it is highly recommended that you consider taking advantage of the flexibility and power of the new statistics and test their use at your site. You may find that adding or modifying the statistics dramatically increases the efficiency of your query plans. On the other hand, you may find for your datasets that keeping the statistics as they were works just fine.

Some General Terminology

Let's give some definitions before going on.

- Histogram: Used to represent the distribution of values in a column. Used by the optimizer when costing SARGs. Consists of steps/cells which represent either a range of values or a single value, weights which represent the percentage of the column occupied by the values of the cell and the boundary values which delineate the cell.
- Attribute of an index: Any column that is part of an index, single or multi-column.
- Column-level statistics: Statistics that describe the data in a column.
- Table/index level statistics: Statistics that describe a table and its indexes.

Natural, Default, Added, and Modified Statistics

Column-level statistics are created/updated based on the underlying data. This happens when you run any form of **update statistics**, and can be thought of as *natural* statistics.

The *default* statistics can be thought of as the result of running:

```
update statistics table_name [index_name]
```

This will create or update statistics on the leading column of the index/indexes. This was the only option in pre-ASE 11.9.2 versions. When you use **update statistics** to create/update statistics on any column other than the leading column of an index or indexes, the resulting statistics are called *added* or *additional* statistics.

Any statistics value that you directly change via optdiag is considered a *modified* statistic.

Why Add or Modify Statistics?

Adding or modifying statistics can result in more efficient query plans than in previous versions of ASE. This is primarily due to the fact that you can now provide the optimizer with a great deal more information about your indexes and data.

Adding Statistics: Why, Where, and How

Adding statistics to columns, indexed or non-indexed, is the most common way to take advantage of the new statistics. Let's look at adding statistics to inner columns (minor attributes) of composite indexes first.

With statistics only on the leading column (major attribute) of an index, the optimizer is limited to information for that column only, and must make assumptions about any other columns in the index. The additional statistics on inner columns gives the optimizer a complete view of the index, and it will not need to make any assumptions about how selective the column is. Let's take a look at the example of a simple test query:

```
select * from test
where col1 > 200
and col3 = .40
and col2 <= 300
```

In the first example, statistics are only on the leading column of the index (the default statistics). You can see that without statistics on col2 and col3, the optimizer has to use default selectivity values (formerly called *magic numbers*) to estimate selectivity for those columns. These values are assumptions about selectivity and are not likely to be accurate.

Traceon 302 output:

```
Estimated selectivity for col1,
  selectivity = 0.999597.

No statistics available for col2,
using the default range selectivity to estimate selectivity.

Estimated selectivity for col2,
  selectivity = 0.330000.

No statistics available for col3,
using the default equality selectivity to estimate selectivity.

Estimated selectivity for col3,
  selectivity = 0.100000.

costing 22243 pages, with an estimate of 16493 rows
Search argument selectivity is 0.032987.
```

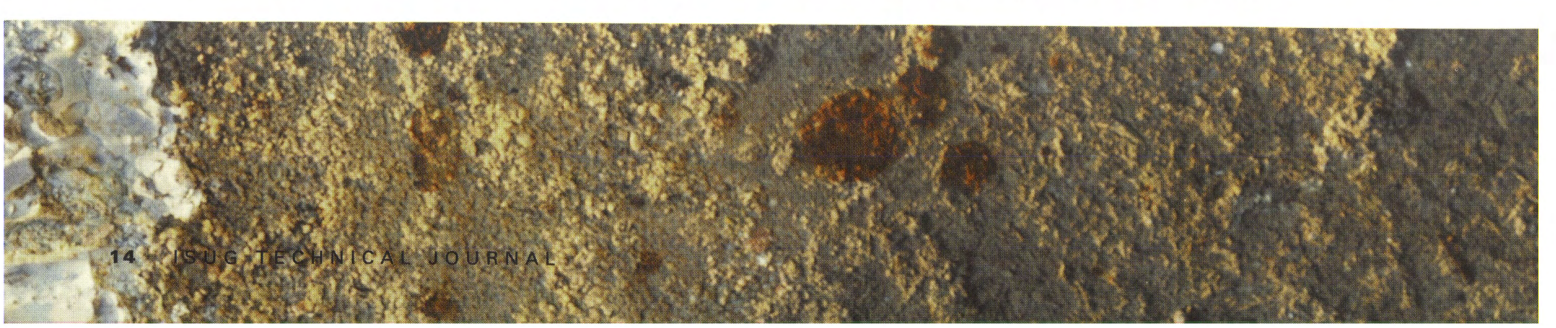
Now we add statistics to all columns of the index (col1, col2, and col3):

```
Estimated selectivity for col1,
  selectivity = 0.999597.

Estimated selectivity for col2,
  selectivity = 0.014925, upper limit = 0.052684.

Estimated selectivity for col3,
  selectivity = 0.020003, upper limit = 0.060138.

costing 5898 pages, with an estimate of 149 rows
Search argument selectivity is 0.000298.
```

Statistics io output:

```
Table: test scan count 1, logical reads: (regular=5895 apf=0
total=5895),
physical reads: (regular=143 apf=0 total=143), apf IOs used=0
Total writes for this command: 0

(147 rows affected)
```

After adding statistics to the inner columns of the index, the estimated cost of the query is far more accurate.



Now let's take a look at the effects of adding statistics to a non-indexed column that's participating in a simple join:

```
select * from test t,test2 t2
where t.col1 = t2.col2
and t.l_orderkey > 200
and t.col2 = 100
and t.col1 <= 300
```

Without statistics on test2.col1:

```
Traceon 302 output
Estimated selectivity for col1,
  selectivity = 0.100000.
```

With statistics on test2.col1:

```
Estimated selectivity for col1,
  selectivity = 0.000052, upper limit = 0.052684.
```

Statistics on the non-indexed column t2.col1 resulted in more accurate cost estimates. In the case of this simple join, the presence of statistics on the non-indexed column resulted in a different join order being used. Having statistics on columns that participate in joins is especially useful when you enable sort-merge joins in ASE 12.0 or above.

How To Add Statistics To Columns, Indexed or Non-Indexed

The fundamental changes to **update statistics** deal with adding statistics to a column in a number of ways. Also included in the new functionality is the ability to specify the number of steps to use for the column's histogram (*distribution statistics*). See Sybase documentation for a general introduction to the syntax and the basic changes to **update statistics**. However, below is some information on which form of **update statistics** to run in various situations.

To add statistics to columns, one column at a time, use the following syntax:

```
update statistics table_name (column_name)
```

You may have heard or read about **update index statistics**. This extension will create or update statistics for all columns of all indexes of a table or for a specified index:

```
update index statistics table_name [index_name]
```

Another option is **update all statistics**:

```
update all statistics table_name
```

This option will create or update statistics on all columns of a table. Be very cautious of this option for a couple of reasons: It can take a very long time to run, and in the vast majority of cases having statistics on all columns of a table is not necessary. Always test changes to the statistics completely before implementing them in production.

A Word On Maintaining Column-Level Statistics

As with everything in life, there's a trade-off with adding statistics. As we all know **update statistics** takes time to run, and the larger the table the longer it takes.

In pre-ASE 11.9.2 versions, **update statistics** had to read the leading column of an index and gather the statistics for that column only. Since the column was in sorted order, there was no need to do anything but the read (scan). In 11.9.2 and above, you can now add statistics to any column. If the column is not the leading column of an index, it needs to be sorted before the read in order to gather the statistics. This adds I/O and time to the process. It will also require space in tempdb to handle the worktable for the sort. If the column is in an index, the size of the worktable will be the size of the index leaf pages, plus or minus a few pages. If the column is not in an index, the worktable will be the size of the table.

In 11.9.2 or above, an additional scan will need to be done to gather the cluster ratio statistics. A table scan will need to be done if you specify only a table name or a table and column name in **update statistics**. If you specify an index, an index leaf scan will occur; except in the case of a clustered index on an All Pages Locked table, where a table scan will be done.

In most cases, the cost of maintenance will be outweighed by the more efficient plans the optimizer will generate. The added maintenance cost is another good reason to test the

effectiveness of adding statistics to columns. This added maintenance cost is also a very good reason to rethink running **update all statistics** or **update index statistics** without testing first.

Changing the Number of Requested Steps

The number of steps (cells) in the histogram of a column has a direct effect on the optimizer. If you create an index or statistics on a column that has no statistics, the default number of steps (20) is used. If statistics exist on the column, the number of requested steps used will be the last number of requested cells used, unless you specify a different value. You can also specify the number of steps to be used. There are a couple of reasons you may want to consider changing the number of steps for a column.

The first is to get more Frequency Count cells in the histogram. These are the most accurate type of cell, since they represent only one value. Frequency Count cells generally occur when a value or values occupies a large number of rows. A frequency count cell can be "pulled" from a Range Cell (a cell representing multiple values) if a value occupies more than 50% of a cell width. The width of a cell is the number of rows divided by the number of requested steps minus 1. You can use this formula to tell if a value that you know to be highly duplicated in your dataset will have a frequency count.

Another reason to increase the number of requested steps is to make the column's histogram more granular. A more granular histogram makes it easier for the optimizer to accurately estimate the cost of a SARG. This is especially true for range SARGs. For such SARGs, the optimizer estimates how close a SARG value falls to either boundary of a cell and uses this to estimate selectivity; the narrower the cells, the more accurate this is. Keep in mind that this applies only to Range Cells, since they represent more than one value and a SARG value's position within the cell must be estimated. When a SARG value falls into a frequency count cell, no estimations are needed since the cell represents only one value. Because the number of steps to use in a histogram is taken from the old distribution page during upgrade, do not delete statistics after upgrade completes.

Update statistics has been extended to allow you to specify the number of steps to use for building the histogram(s). For example:

```
update statistics table_name using 100 values
```

The above syntax will update statistics on the leading columns of all indexes in the table and will use 100 as the number of requested steps.

```
update statistics table_name (col1) using 100 values
```

The above syntax will create or update statistics on the specified column, col1, using 100 as the requested number of steps.

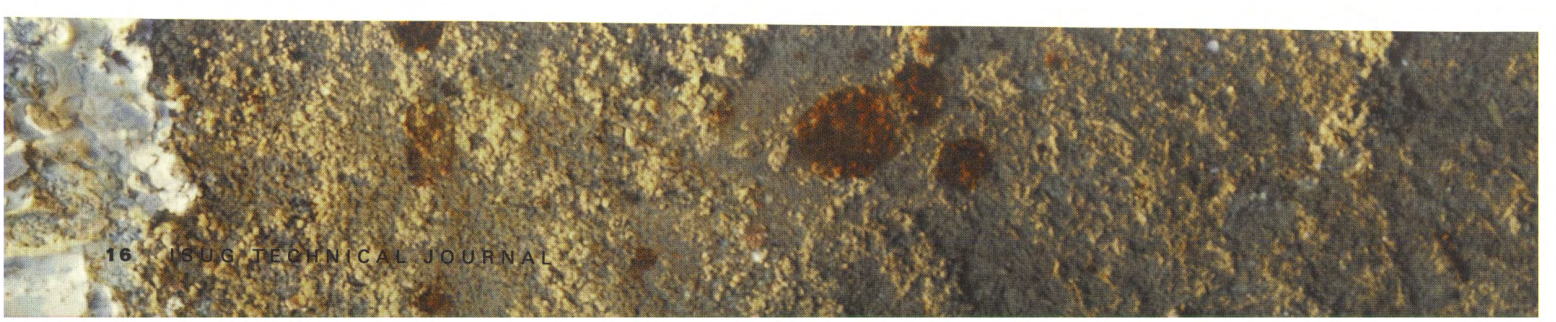
The number of steps can also be specified in create index with the same using X values syntax.

There is a slight trade-off when increasing the number of steps. Steps require memory taken from procedure cache whenever the optimizer needs to read them. The amount of memory is small, but the larger the datatype of the column and the more steps in the histogram, the more is needed. In the most extreme cases, this can have an adverse effect on parse and compile time. In most cases, the more efficient query plans that result from more steps far outweighs the cost of caching the steps.

A Word On Statistics and Upgrade

When upgrading to 11.9.2 or above from an earlier version, you'll need to take the statistics into consideration.

During upgrade, if a column is the leading column of an index, the old distribution page is read and its values are used to establish the new statistics. This essentially copies the old statistics values into the new values. This copy is not as accurate, as statistics obtained from the underlying data by ASE 11.9.2 or above. For example, the values of the steps in the distribution page are used to create the boundary values for the new histogram steps. However, the weights are estimated by using the number of values that fall between each step of the distribution page. This is not as accurate as getting the boundary values and weights from the data. Frequency count cells are created when a value appears in more than one step



in the distribution page. The number of requested steps to use in the new histogram is the number of steps in the old distribution page.

After upgrade completes, it is highly recommended that you run **update statistics** for all tables. This will gather the new, more accurate statistics. For this first post-upgrade run, use the syntax you used in previous versions:

```
update statistics table_name [index_name]
```

You may have heard that you ought to run **update index statistics**, or even the extreme **update all statistics** after upgrading. While it's recommended that you consider and test the effects of statistics on inner columns of composite indexes and/or on non-indexed columns, you ought to do so after upgrading the statistics to the new format. Hold off on adding statistics until you have a chance to test them.

There has been some folklore recently about deleting statistics after upgrade and then running **update statistics**. This is not necessary and can result in inefficient query plans. Since the number of requested steps are taken from the number of steps on the distribution page, the resulting histogram will closely match that of the distribution page. If you drop and recreate the statistics, the histogram will be built using the default of 20 steps. This is because the previous step count is lost when the statistics are deleted. You should avoid changing the granularity of the histogram until you've had a chance to test it against your queries.

The bottom line is that after upgrading to 11.9.2 or above from a previous ASE, do not delete or add statistics until you've tested their effects on your queries. But, do run **update statistics** as soon as possible after the upgrade completes.

Modifying Statistics: Why, Where and How?

You can now directly write the column-level statistics. You can use optdiag to do this since it's not advisable, nor supported by Sybase, to write directly to the system tables. There are a number of reasons you may want to write the statistics directly.

First, though, let's talk about how to write the statistics. As mentioned, use optdiag to do this by getting an optdiag output file. Use the **–o file_name** option at the command line. Once you have the file, you can begin to modify the statistics. You can use any text editor to do the job. Save a copy of the original optdiag output file in case you need to start over. Once the file is edited, you can read it back in via optdiag using the -i option. When the optimizer uses statistics that have been modified, traceon 302 will print this message:

```
Statistics for this column have been edited.
```

This will help you know when modified statistics are used by the optimizer.

A Note On Maintaining Modified Statistics

Let's take a look at what you'll need to do to maintain modified statistics should you decide to use them. The majority of the column-level statistics are not persistent; they will be overwritten the next time you run **update statistics**. The one exception is the default selectivity values. Keep this in mind if you decide to modify statistics, as it will add a step when updating the statistics.

In the most common scenario, you will be modifying only one or a few values. In these cases, you'll need to get a new optdiag output file after **update statistics** is run, re-edit it, and read it back in.

Modifying Statistics

There are a few different situations in which you may want to consider modifying the statistics.

Changing the Total Density Value to Deal with Data Skew

Data skew occurs when a few values each occupy many rows of a column, while many values occupy a few rows each. Data skew will have a direct effect on the Total Density value and thus on the optimizer's costing of joins. You'll see a few Frequency Count cells in the histogram surrounded by Range Cells. The effect this has on the Total Density value can result in an inaccurate number of rows being estimated for a join.

How do we modify Total Density to get around data skew? One approach is to set the Total Density value to match the Range Cell density value. This is a fine approach; however, you want to be cautious about setting the Total Density value

too low, since it may result in the optimizer being overly optimistic about the cost of a join on the column, which in turn results in an inaccurate number of rows being estimated. In most cases of data skew, the Range Cell Density is much lower than the Total Density. Optdiag output:

```
Range cell density: 0.0000502421670203
Total density:      0.2697381850000000
 Step  Weight                 Value

    1  0.00000000   <=        0
    2  0.05263000   <=        5222
    3  0.05264000   <=        10564
    4  0.05265000   <=        15779
    5  0.05263000   <=        20998
    6  0.04125000   <=        24999
    7  0.00000000   <         40000
    8  0.31933998   =         40000
    9  0.05263000   <=        96995
```

In the example above, the histogram is skewed on the value 40000 (a Frequency Count Cell) with almost 32% of the column occupied by that value. Notice that the Total Density is much larger than the Range Cell Density. Since the Total Density value is used to cost all joins that the column participates in, setting it too low may adversely effect other joins. Whenever you add **of modify** statistics, make sure to test the changes completely.

A new system stored procedure, *sp_modifystats*, can be used to set the Total Density value to equal the Range Cell Density value. This procedure is available in ASE 12.0 and 11.9.2.2. Again, be careful not to set the Total Density value too low. If the Range Cell Density is very low, don't use sp_modifystats, but rather change the Total Density directly via optdiag.

The Total Density value is also used as the default selectivity value for equi-SARGs when the value is unknown at runtime. If you have queries that have unknown eqi-SARG values (usually due to local variables) and the Total Density is not resulting in a good plan, try changing it. Again, keep in mind that any change to the Total Density will effect all joins on the column.

Changing the Default Selectivity Values

The default selectivity values are used to determine selectivity when the value of a SARG is unknown at runtime. Unknown values usually occur when a local variable is used in a query or a stored procedure (values are known if parameters are used in a stored procedure):

```
declare @1 int
select @1 = 100
select * from table where column = @1
```

Unknown values can also result, in some cases, when using functions or expressions in a SARG. In earlier versions of ASE, the magic numbers were hard-coded and could not be changed. In ASE 11.9.2 and above, the default selectivity values can be modified.

There are two default selectivity values. The "Range selectivity" value is used when the value of a range SARG (<,<=,>,>=) is unknown. By default this value is 0.33 (examples are from optdiag output):

```
Range selectivity:      default used (0.33)
```

The "In between selectivity" is used when the value of a between SARG is unknown. Its default value is 0.25:

```
In between selectivity:      default used (0.25)
```

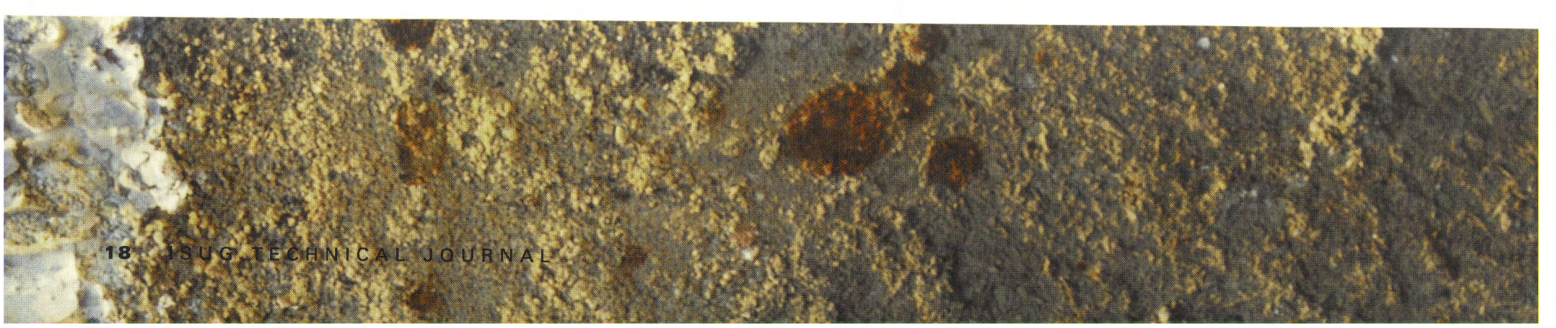
To modify either value make sure your changes are in the format below:

```
In between selectivity:      0.10
```

As mentioned earlier, the default selectivity value for an unknown equi-SARG value is the Total Density.

Modifying Statistics For Out of Range SARG Values

If a search argument value is larger than the largest boundary value in the histogram, or if it is smaller than the smallest boundary value in the histogram, it will be out of bounds.



This can occur if the search value is not in the column; or it is in the column but the statistics are not up to date and it is thus beyond the boundaries of the histogram. In either case, the optimizer must use special costing for out-of-bounds SARG values. The selectivity value used will be either 0.0 or 1.0, depending on the type of operator and where the value falls.

For example, if the value is less than the smallest value in the histogram and the operator is >=, the selectivity used will be 1.0. If the value is greater than the largest value in the histogram and the operator is >=, the selectivity used will be 0.0. If the value is outside either boundary, and an equi-SARG is being used, the selectivity value will be 0.0. In any case, the selectivity value is likely to not be accurate.

Traceon 302 will print the following message when there's an out-of-bounds SARG value (again, the selectivity value can be either 1 or 0):

```
Estimated selectivity for colA,
    selectivity = 0.000000, upper limit = 0.000000.
Lower bound search value 10000 is greater than the largest
    value in sysstatistics for this column.
```

There are two ways to get around out-of-bounds costing. The first is to add a dummy row to the table that contains a value well outside a boundary. This value will appear as the histogram value for the lowest or highest boundary of the histogram, the SARG value will now fall within the bounds of the histogram, and costing can be done. This approach is not always acceptable, though, since it requires the dummy data to be in the table.

Another approach is to use optdiag to change the value of the lowest or highest boundary value so that the search value always falls within bounds. In the first case, the dummy value is persistent; it will always be in the table. In the second case, the value is not persistent and an optdiag output file will need to be edited and read back in after each update statistics run.

Here's an example. Let's say it's July 15 and **update statistics** hasn't been run in a while. You want to see rows for July 12. The value July 12 will be out of range. If you were to use optdiag to change the last cell value (in this case, step 20) to say August 1, full costing could be done (example from optdiag output):

```
18  0.05301946  <=  "May 1 2000 12:00AM"
19  0.05290456  <=  "Jun 1 2000 12:00AM"
20  0.04818739  <=  "Jul 1 2000 12:00AM"
```

Modifying Histogram Cells

There are a few reasons why you might want to modify the cells of a histogram: If you want to add a dummy value for out-of-bounds SARG values as mentioned earlier; or if you want to add frequency count cells to the histogram. If you want to directly write statistics instead of running **update statistics**, or if you decide to modify the histogram's cells, there are a few rules to keep in mind:

- The step numbers must increase monotonically
- The weight of each cell should be between 0 and 1.0
- The sum of the cell weights must be close to 1.0 (0.99 to 1.01)

If you plan to modify or create a histogram for a column, make sure you understand the distribution of the values in the column. Care must be taken when doing this: Remember the optimizer is dependent on the histogram for costing SARGs. As with any change to the statistics, make sure to test thoroughly before implementing.

Conclusion

The changes to how optimizer statistics are stored and used provide you with powerful and flexible features and functionality. You can add column level statistics to inner columns of composite indexes, giving the optimizer much more accurate information about an index to use when costing a query. You can add statistics to columns that are non-indexed, thus giving the optimizer a clear view of the column when costing a join. You can specify the number of histogram steps to be used for the column—**update statistics** has been extended to allow you to do this. You can also modify all the column level statistics via optdiag, thus changing everything from the default selectivity values to the histogram.

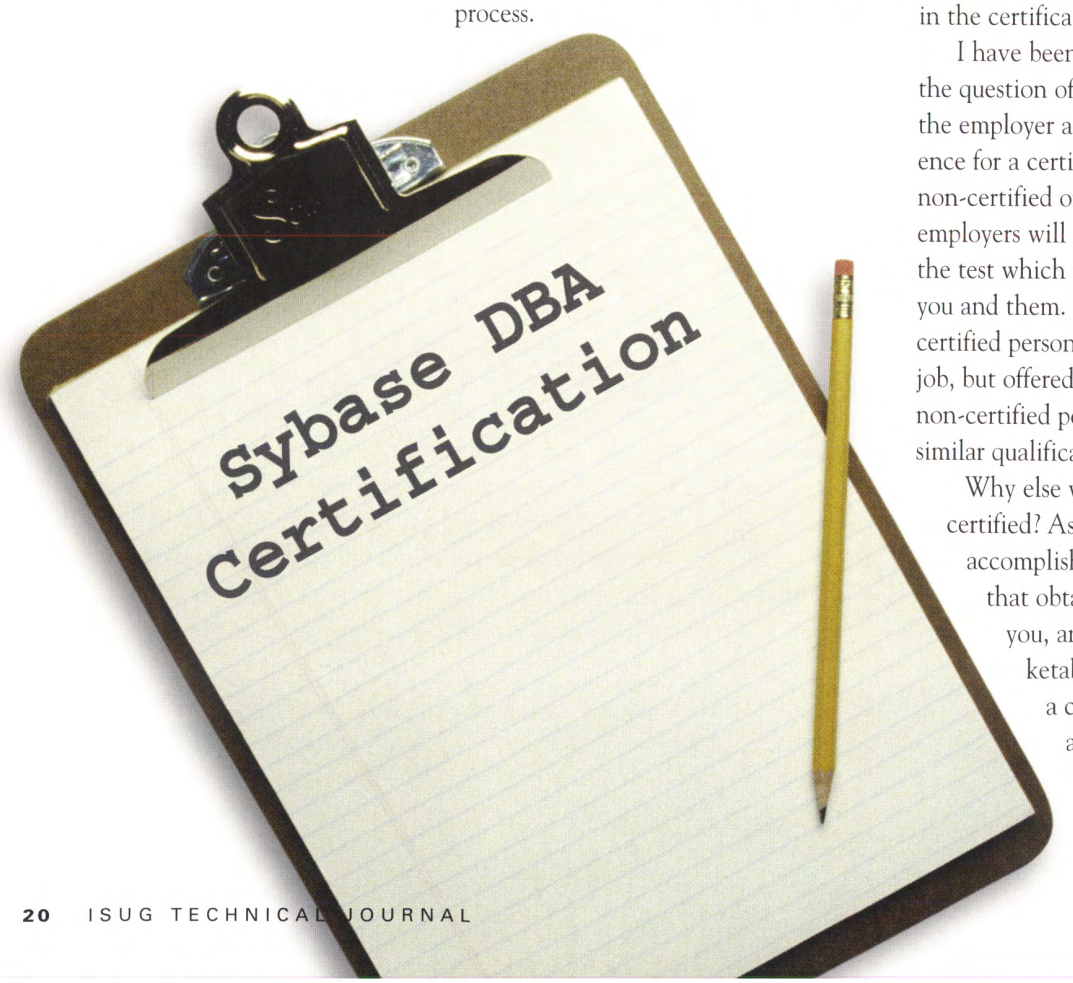
All this power and flexibility requires that you weigh the cost against the benefits and fully test any changes to or additions of statistics. We certainly recommend that you consider taking advantage of this new functionality. ■

Getting Your Sybase DBA Certification

By David Straiton

Should you or shouldn't you get your Sybase Certification?

Certification has been an issue among DBAs and consultants as long as I have been a practicing Sybase DBA. Many question the Sybase certification program: Does it add value to their resume and their marketability, is it worth taking? The next set of questions is usually how and what. How do I go about taking the certification test, how do I sign up for it and how do I study for it? They then wonder which certification to study for. This article is intended for those who have had similar questions and would like to know more about the process.



Why Get Certified?

When I teach Sybase classes, I am almost always asked the above questions. Let me first address the Why of the matter. As a Sybase professional you are specializing in a vendor-specific set of products. Having your certification certainly helps your marketability, and, more importantly, displays not just your aptitude, but your attitude. As a professional, a certification gives added weight and credentials to an area of expertise you already profess to be interested in. This shows your employer as well as yourself that you are proficient in the certification area.

I have been to many interviews where the question of certification came up, and the employer actually expressed a preference for a certified employee over a non-certified one. Additionally, many employers will pick up the cost for taking the test which provides value for both you and them. I have seen cases where a certified person was not only offered the job, but offered a higher salary than a non-certified person with otherwise similar qualifications.

Why else would you want to get certified? Aside from the sense of accomplishment and confidence that obtaining a certification gives you, and the increased marketability that can accompany a certification, this is a day and age of technical jobs with a need of people with technical skills. I have been in data

processing since 1980, and I can attest to the fact that it is hard to maintain your skill level in an era of rapidly changing technologies. When a major paradigm shift occurs such as the transition from legacy mainframe systems to a distributed client/server system, a whole new generation of skillshets is needed.

By being one of the first to get certified in a new technology, you maintain an edge and the opportunity for the preferred jobs that are available. It is not enough that a new technology comes along and that you get certified in it. You must stay current in that same technology as it matures.

Sybase first offered certification for its System 10 server, then continued to offer certifications as its technology changed to System 11, ASE 11.5, and now ASE 12.0. A technical person cannot rest on their laurels by obtaining a certification and then becoming complacent about it. It is also often hard to motivate yourself to train on technologies that your company either does not use yet, or will never use. A certification is a means of identifying an end goal, which has a measurable result, i.e.; a test score. This has given me sufficient motivation to pursue studying for and achieving my certifications. When I do not have a goal, I am often not motivated enough to pursue training and learning and one of Newton's laws takes over: "A body at rest tends to stay at rest."

Getting a Sybase Certification

Due to space limitations, the scope of this article is limited to the Sybase Certified ASE Associate and Sybase Certified Professional certifications. For the Sybase Certified ASE Associate certification, the suggested requirements for taking this test are the courses: Intro to SQL, Fastrack To Adaptive Server Enterprise, and System and Database Administration.

For the Sybase Certified Professional certification, the courses for the Associate certification and the Performance and Tuning for DBAs course is recommended. Additionally, some hands-on experience between taking the associates courses and the Performance and Tuning course for three to six months should take place. Get as much hands-on experience during the labs as possible, even to the point of extra labs or variations of the labs. My view in the value of an instructor-led class is to get as much information from the instructor as is possible, the most important of which is the instructor's actual experience as opposed to the class materials. (Try taking the quiz on page 22.)

The Value of the Command Line Approach

I think it is important to point out that the DBA who practices a more command line, hands-on approach is more likely to be successful. The DBA who relies on such GUI tools as Sybase Central, Desk Top DBA, and DB-Artisan will not have as firm a grasp of the syntax as is found on the certification test. A GUI tool often performs tasks for you, leaving you to wonder as to the methods it is using.

Practice is a key element of success as well. The tests cover a wide range of features and a DBA may not use all of these features, so a good approach to knowing this information is to actually put it in practice. Most companies have a test environment or a sandbox that the DBA may play in. If there are features that you are not clear on, I recommend that you actually implement these features where possible.

One of the most successful avenues of preparing for these examinations is to form a study group. If you are the only DBA in your company or the only one interested in getting certified, a good resource for finding others that would like to pursue a study group for certification can be found in your local Sybase User Group. A list of User Groups can be found at www.isug.com under the User Groups tab on the left.

Certification Resources

You can start with the Sybase URL http://slc.sybase.com/certification/index.html. This URL has sections such as Why Certification from Sybase, Frequently Asked Questions, List of Exams, What Exams Should I take for My Certification, Registering for an Exam/Exam details, sample exams, and more. There is also a section for getting a 25% discount on Sybase Certified Professional Exams at the 2000 Sybase TechWave conference.

A Study Guide exists for some of the certifications at http://slc.sybase.com/certification/cstudy.html for SQL Server 11 and Adaptive Server 11.5, and Adaptive Server Enterprise 12.0. These study guides have Frequently Asked Questions followed by a study guide loosely covering exam content.

The Sybase website http://support.sybase.com has the manuals to get information in more depth than the Sybase Class notebooks might provide. The manuals should be used to clear up any confusion about a particular topic or to just provide additional information. The Sybase Technical Library on CD is another excellent resource.

There are external websites dealing with various certifications including Sybase in case the resources listed here are not sufficient. ■

Test Yourself: The Sybase Certification Study Guide

Following is a sampling of questions from the Sybase Certification Study Guide, intended to help assess a DBA's readiness level to test for certification. The information came from Sybase's education materials and manuals. If there are any questions or general comments, send them to David Straiton at straiton@sybase.com.

1. **What stored procedure displays all active roles?**
   A) sp_showrole
   B) sp_helprole
   C) sp_activeroles

   Answer: **C**

2. **How do you change the execution class of a login?**
   A) sp_addexeclass classname, login
   B) sp_changexeclass classname, login
   C) sp_bindexeclass login,"LG",null,"exe1"

   Answer: **C**

3. **Adding a login and specifying the default database makes that login a user in the specified database.**
   A) True
   B) False

   Answer: **False**

4. **Which statement sets the servername inside an ASE server after installation:**
   A) sp_addserver servername,null, servername
   B) sp_addserver servername,local, servername
   C) add server servername,local

   Answer: **B**

5. **How do you change the future space allocations of a table or index:**
   A) alter table tablename with fillfactor = x
   B) alter table tablename with MAX_ROWS_PER_PAGE = x
   C) sp_chgattribute tablename, "max_rows_per_page", x

   Answer: **C**

6. **What is the stored procedure to help develop your DBCC Checkstorage abilities?**
   A) sp_plan_checkstorage
   B) sp_plan_dbccdb
   C) sp_help_checkstorage

   Answer: **B**

7. **Asynchronous prefetch can improve performance for:**
   (*Choose all the correct answers*)
   A) Sequential scans
   B) The Update Statistics Command
   C) Recovery
   D) Some DBCC Checks
   E) Point queries

   Answer: **A, B, C, D**

8. **Component Integration Services is built-in to ASE 11.5**
   A) True
   B) False

   Answer: **True**

9. **Select into is not allowed to transfer the contents of one table into a new table on any supported remote server.**
   A) True
   B) False

   Answer: **False**

10. **Which statement is not true about the dbcc checkstorage command:**
    A) Uses a named cache so that running it on the target database does not interfere with concurrent use of that database
    B) Scales linearly with the aggregate I/O throughput
    C) Does not lock tables or pages for extended periods
    D) Separates the functions of checking and reporting
    E) Provides a fix option that is transparent to the user

    Answer: **E**

11. **Each new threshold must be at least:**
    A) 2 times @@thresh_hysteresis value
    B) 3 times @@thresh_hysteresis value
    C) 64K
    D) @@thresh_hysteresis value

    Answer: **A**

12. **What is the maximum number of devices in a server?**
    A) 255
    B) 256
    C) 1024
    D) 2048

    Answer: **B**

13. **What command is used to clear a transaction log when it fills up?**
    A) dump database
    B) dump tran with truncate_only
    C) dump tran with free_space
    D) truncate syslogs

    Answer: **B**

14. **Which of the following is not a minimally logged operation?**
    A) select into
    B) Fast BCP
    C) Slow BCP
    D) truncate table
    E) Create Index

    Answer: **C**

15. **What is the default replacement policy of ASE 11.5?**
    A) strict MRU
    B) relaxed MRU
    C) strict LRU
    D) relaxed LRU

    Answer: **C**

16. **What Unix environment variable specifies the default Sybase server?**
    A) $SYBASE
    B) $DSQUERY
    C) $SERVER
    D) $DSLISTEN

    Answer: **B**

17. **In what system table would you find the application name of a process?**
    A) sysobjects
    B) sysprocedures
    C) sysprocesses
    D) syslogshold

    Answer: **C**

18. In the Unix environment, what command shows ASE and Backup servers currently running on the local machine?

A) sp_showserver
B) sp_serverinfo
C) showserver
D) list_servers

Answer: **C**

19. What information cannot be found in the Server Errorlog?

A) ASE Version
B) data and procedure cache size
C) default sort order
D) amount of swap space used

Answer: **D**

20. What is the value in the segmap column in the sysusages table when the data and logs are placed on the same device?

A) 3
B) 4
C) 7
D) 8

Answer: **C**

21. How many indexes can be associated with one table?

A) 16
B) 32
C) 200
D) 250

Answer: **D**

22. How many indexes associated with one table can be clustered?

A) 1
B) 32
C) 249
D) 250

Answer: **A**

23. Additional Network Memory is a configuration parameter that uses memory from:

A) Total Memory dedicated to Sybase Server
B) Operating System
C) Default data cache
D) Procedure cache

Answer: **B**

24. Does the Sybase server by default allow descending table scans?

A) Yes
B) No

Answer: **A**

25. Given the following scenario, what, if anything, gets rolled back?

Begin tran	**insert**
insert ...	**delete**
update ...	**FAILURE**
checkpoint ...	

A) 1st insert
B) 2nd insert and delete
C) Everything
D) Nothing

Answer: **B**

26. What system table contains the text of stored procedures kept

A) sysprocedures
B) syscomments
C) systext
D) sysmessages

Answer: **B**

27. Which one of the following is not a file descriptor in the metadata cache?

A) Number of open indexes
B) Number of open databases
C) Number of open Objects
D) Number of remote logins
E) spin lock ratio

Answer: **D**

28. Given the following table excerpts, construct the create/alter database statements master..sysdevices (excerpts). Assume database name is *contacts*.

low	*high*	*status*	*cntrltype*	*name*	*phyname*	*mirror-name*
67108865	67109887	2	0	contacts_dev	/usr/dev/contracts_dev.dat	null
50331648	50334207	2	0	data_dev	/usr/dev/data_dev.dat	null
33554432	33555455	2	0	log_dev	/usr/dev/log_dev.dat	null

master..sysusages

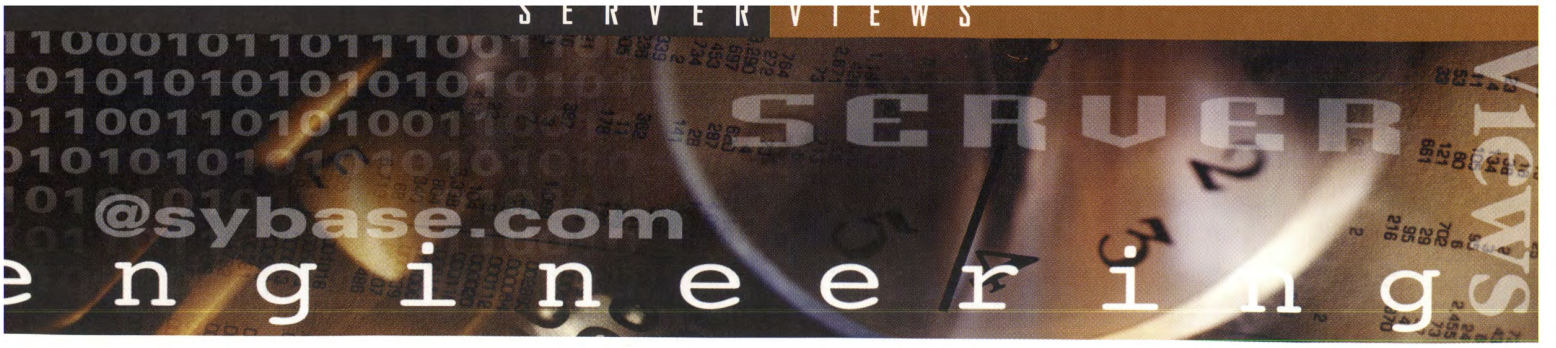
dbid	*segmap*	*lstart*	*size*	*vstart*	*pad*	*unreservedpgs*
1	7	0	1536	4	NULL	0
1	7	1536	1024	3588	NULL	0
1	7	2560	1024	4612	NULL	568
2	7	0	1024	2564	NULL	680
2	7	1024	512	8708	NULL	512
3	7	0	1024	1540	NULL	680
4	3	0	2560	50331648	NULL	1328
4	4	2560	1024	33554432	NULL	1024
4	8	3584	1024	67108864	NULL	1024
5	3	0	2560	5636	NULL	1392
5	4	2560	512	8196	NULL	400
5	7	3072	512	9220	NULL	0

syssegments (in the database)

segment	*name*	*status*
0	system	0
1	default	1
2	logsegment	0
3	contacts_seg	0

Answer:

```
create database contacts on data_dev=5
log on log_dev-2
alter database contacts on contacts_dev=2
use contacts
sp_addsegment contacts_seg, contacts, contacts_dev
sp_dropsegment system, contacts, contacts_dev
sp_dropsegment "default",contacts,contacts_dev
```



By Ian Smart

I was in our New York office the other week, and one of our top sales people commented with a beaming smile, "Hey, Ian, the phones are ringing again." He was referring to the fact that users are proactively coming to Sybase asking for our Electronic Portal solution—EP 1.0. The industry is buzzing with Sybase's name again. If ever there was a ringing endorsement that with EP Sybase has really set the direction of the market, this has to be an excellent indicator. It is indicative of the extremely "up" feeling that there is around Sybase at the moment, and I for one am glad to have friends coming to me wanting to know what Sybase is up to.

But, having achieved such a momentous milestone as EP 1.0, it would be easy for us at Sybase to take our feet off the gas and relax for a while, basking in this sense of achievement. The reality could not be further from the truth. We are now all focused on the next Sybase EP releases which we will be announcing at this year's TechWave conference in Orlando, Florida. Having gained the leadership position in the Enterprise Portal marketplace, there is no way that Sybase is going to let go of it. Rather, we are going to ensure that we consolidate on all the work and achievements that have so far been made and reinforce our position.

Sybase EP Planned Enhancements

The upcoming Sybase EP releases will focus on extending the functionality that was offered in EP 1.0 and ensuring that customer requirements are genuinely achievable in as rapid a timescale as web time demands. First of all, let us look at the overall functional architecture of the product (Figure 1).

As those of you familiar with the existing EP architecture will recognize, this is not a rethink. Rather, this builds upon the existing Sybase EP framework by adding new functionality in four major areas.

Enhancement of B2B and E-Commerce Functionality

We are now entering the third generation of B2B (business-to-business) solutions. Initially, suppliers created sites to automate order-entry and fulfillment. These were very specialized and required the user to have access to other material or knowledge to use the sites correctly. These were followed by companies such as Amazon.com, who provided buy-side sites to allow customers to make purchases via the Internet. These sites empowered the buyer and started a trend towards the consolidation of catalogs and e-procurement from the items offered.

In many peoples' minds, it was the creation of these second-generation sites that kicked off the e-commerce revolution. The third generation of B2B solutions will enable exchange networks of business communities. These communities may well already have an existing B2B marketplace that they use. The B2B server coming with future versions of Sybase EP can both be used to implement a B2B marketplace and to provide the B2B services that are required to put in place a common infrastructure to facilitate a full

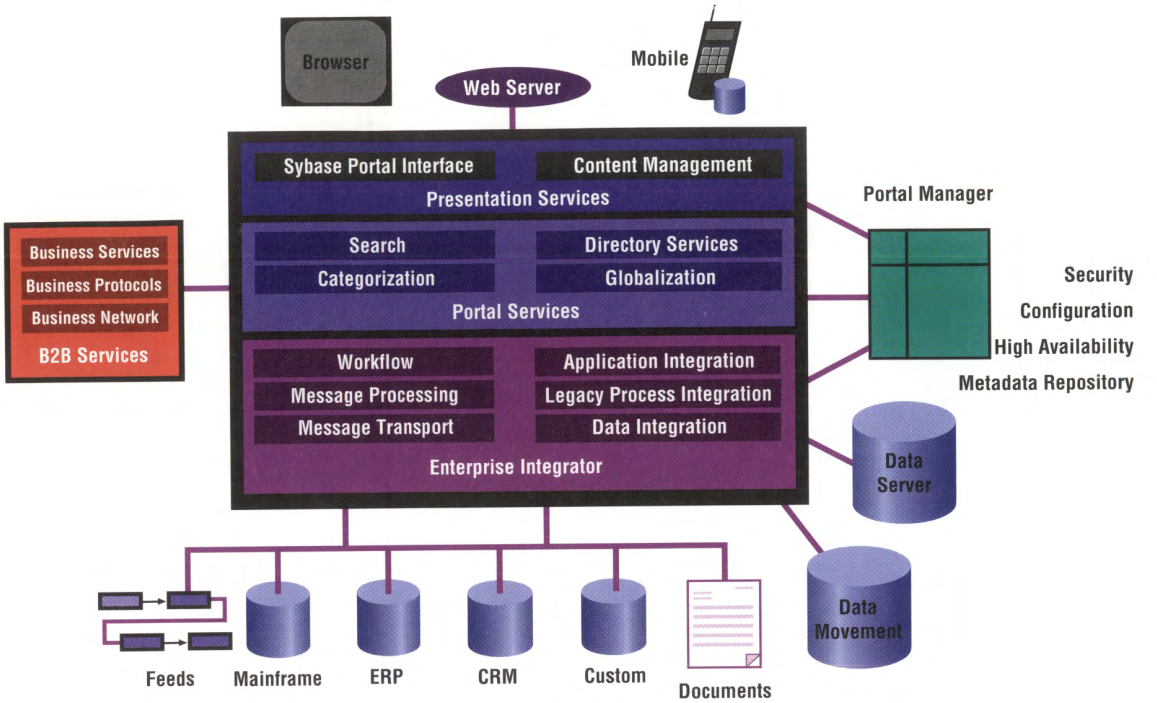


Figure 1 *Sybase Enterprise Portal–Functional Architecture*

trading network. Those of you who saw the SyTen Portal at the EP launch events earlier this year will have already seen one of the many potential uses that this sort of capability can provide. This application offered trading between telco vendors of spare cable bandwidth by linking their portals and providing facilities that you would expect in a virtual marketplace of this sort, e.g. targeted pricing, messaging, support for offer placement, and a secure environment for trade.

To enable the creation of these exchange networks, Sybase EP will, as you would expect, enhance our services to support the requirements of first and second-generation e-commerce sites. This will be through the provision of content services such as catalog management and support for suppliers through the provision of catalog descriptions, availability checking, purchase ordering, order status and invoicing, as well as shipping and payment. In addition, since globalization of the marketplace is a key driving factor of e-commerce, support is provided for different currencies, address formats, and languages.

But all of this functionality, critical though it is, is not enough for third-generation B2B solutions. To enable exchange networks to be put in place, EP will support, through a B2B server, asynchronous messaging between portals using a "publish and subscribe" mechanism. These will use recognized business protocols such as cXML and xCBL, as well as XML, to ensure that the data is transmitted in a commonly understood self-describing format. As the third generation of e-commerce solutions start to be put in place, the B2B server is seen as meeting key customer requirements to allow customers to either be able to host their own exchange network, or allow them to participate in a secure manner in existing marketplaces, or both.

Consolidation of Integration Infrastructure

The enterprise integrator provides a single engine to run the EP customer's business logic. Its metadata driven architecture enables the existing integration functionality, and application integrators to be consolidated into a single messaging infrastructure and enable more rapid integration to be performed. The use of a single metadata repository, with a supporting metadata manager, makes it easier to describe the messages and events that need to be integrated within the portal and then to map the workflow between these different application and business processes.

This use of a single source for the descriptions of all the processes and data to be integrated will enable faster and easier production of true business integration. A further enhancement is the ability to allow the different adapters that are being used to exist upon multiple portals, thus allowing for integration to not require the use of a single portal, or the migration of existing integration capabilities if this is not in the user's best interest. To strengthen the existing loosely

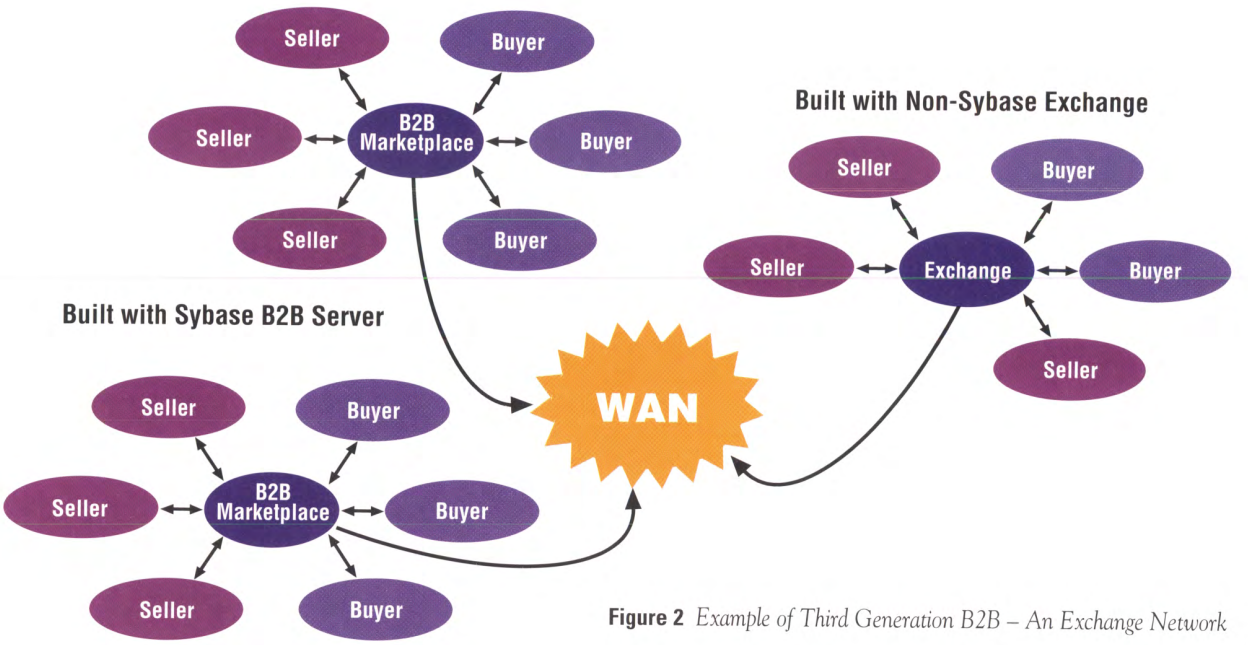


Figure 2 *Example of Third Generation B2B – An Exchange Network*

coupled messaging infrastructure, JMS support is now included. In addition, for those cases that need a more tightly coupled and synchronous mechanism, J2EE connectors have been added.

Building upon the existing Event Broker functionality within Sybase EP 1.0, the Message Broker provides a high-end transform that is capable of handling XML, Structured Data, JDBC results, JMS message transformations, and basically anything that can be understood as a message. In addition, it will support the context sensitive rule based dispatching of messages to CORBA/EJB invocations, JMS Queues, SEEB JMS Queues, and other out-bound protocols.

Enhanced Presentation Management

Through the provision of portelles, Sybase EP will be providing a set of common presentation services that are required to deliver Enterprise Portals. These include the services that customers are fast learning to expect from sites and include searching, channels, document publishing, dictionaries, diary, calendar, etc. In addition, customers can create their own portelles for functionality that is specific to them. All of these portelles can then be used in conjunction with a provided homepage design interface. This allows users to personalize their page by the custom arrangement of content and the placement of the portelles within that, providing a content-rich experience in a very short period of time. The ability to include users' own portelles coupled with the ability to modify colors, images, fonts, etc., provides an extremely rapid way for them to deliver personalized content to the web.

Enhancement of the Infrastructure Functionality

The final area to mention is the hardest to do justice to in the space available in this article. The portal already offers considerable functionality to put in place a customer's required infrastructure. In particular, components are provided in the areas of High Availability, Security, Search, Systems Management, Globalization, and Business Intelligence. Future Sybase EP features enhancements to all of these areas to raise the bar on what Enterprise Portals can provide.

Security – While security is an essential part of any portal, it can also be a drain upon performance. Currently, EP provides the user with a single sign on mechanism that, once the user has been validated, uses Secure Business Objects that are associated with each of the components to manage the security. Future releases of EP will enhance the existing servlet architecture that provides for greater robustness, flexibility and ease of use, while reducing the performance impact upon the user. This is coupled with a simplified version of the Access Control Database (ACDB) that now provides standardized use of roles across the portal. Much of the ACDB data is cached within the portal itself to aid ease of use while reducing the potential for performance degradation. To ensure that each portal user only has access to data that they have been authorized to see, Fine Grain Access Control has been implemented. Finally, the LDAP Object manager provider module enables access to the ACDB within an LDAP server.

Systems Management – Building upon the existing Sybase Central-based portal administration tool, future releases of EP will continue to use EJB's running in the portal that mediate with component agents in order to administer the different areas of the portal. However, the agents have had capabilities for event management and monitoring added, along with support for inter-component dependencies. An SDK for these agents has been produced and this will enable the inclusion of non-Sybase components within this single management interface. As in EP 1.0, all of the configuration data for the agents will be stored within a single configuration repository.

Globalization – As was mentioned earlier, the global marketplace is a major driving factor in the desire to do business across the Internet. But, this is more of an issue in those markets where English is not the primary language (or even a good second choice). To enhance the global capabilities of the portal support is now provided for the French, Chinese, and Japanese languages, along with the inclusion of a translation manager and cross-language searching. In addition, to assist with global e-commerce, a multi-currency mode is supported along with validation for all the major global address formats.

Alliances

The final piece has less to do with the technology of the Enterprise Portal itself, but everything to do with how long it will take customers to bring new areas of business functionality to market. To support the future releases of EP, Sybase has added a large number of new partners to our e-Portal Alliance program. These vendors have committed to delivering their solutions using Sybase EP. Since these are still being finalized at time of writing I have not included a list, but suffice it to say that they include e-commerce, e-CRM, e-analysis, e-corporate business productivity, e-content, e-information and e-workflow companies.

What is the Industry Looking For?

When we talk to the industry in general about their e-Business platform goals, all of them are ultimately trying to achieve the following:

- Rapid Deployment - Administrators are placing a high value on increasing flexibility so that they can design and implement new systems rapidly and respond promptly to change. These new models serve as the means to get closer to their customers, and as an alternative sales channel to bring in more revenue.
- Providing a 24x7 Platform - Worldwide financial systems are linking together to support e-business globally, providing the capability to access systems on a 24x7x365 schedule and perform real-time transactions.
- Highly Scalable and Reliable Systems - Information systems must be highly dependable and very scalable in order to support and sustain rapidly increasing e-business traffic. A recent Jupiter Communications study figures that online holiday sales grew from $3 billion in 1998 to $7 billion in 1999, more than a two-fold increase in just 12 months.
- Unlimited Data Access - New methods of connecting to information systems are making it possible for individuals to access information at all times and from any location. No longer are they confined to hardwired personal computers. Now there are next-generation wireless options and occasionally connected mobile devices such as PDAs, pagers, and smart phones.

The rate of rapid acceptance of this technology can best be highlighted by the struggle I had back in March to buy a WAP phone here in the UK. Compare this to last week, when I went into a mobile phone shop to discover two whole racks of WAP technology, some at a fraction of what I paid only three months ago.

Conclusion

Based on our excellent experiences with users who are already working with EP 1.0, the new functionality will assist with reducing deployment time. In addition, the new portelles, coupled with the other areas of new functionality within the portal, will provide a mechanism to deliver some of the most functionally rich sites for doing business on the Internet. I hope that this brief overview has given you a flavor of what is contained within upcoming releases planned for Sybase EP and shown you that we are not resting on our laurels. We are looking forward to seeing what we can assist our users to achieve once they start using this new release of the software. ■

Ian Smart is Senior Technical Evangelist for Sybase's Enterprise Systems Group. He has addressed U.S. and European ISUG audiences on many occasions, as well as speaking at numerous user group and masterclass events in the UK. If you have a question you would like answered or a comment about an article, Ian can be reached at ian.smart@sybase.com.

In the Palm of Your Hand II

By Thomas Lamb

Data, data everywhere—including the palm of my hand! In this issue, we will continue our exploration of moving a PowerBuilder application to the mobile environment. The objective of this issue's column is to extract data from a PowerBuilder application deployed to a Palm-based flatfile database manager.

Remember that not all of the code is included. All of the PowerBuilder objects discussed in this article can be found on my website at www.lamb.tj. For the other software mentioned, I have provided links where available. For more information on Palm Computing, check out their web site at www.palm.com.

Application Development Environment

Let's review the platform(s) on which this work will be taking place. The information below describes the hardware and software used during the development and testing of this article. This information is provided for your reference, as no warranty is offered or implied that the code in this column will work on other platforms. If you find that it does not work, let me know by sending an e-mail to t.lamb@bigfoot.com.

I would like to personally thank the folks at Land-J Software, developers of jFile, for their help and support. For more information visit their website at www.land-j.com.

Palm Databases

As I mentioned in the last issue, the journey to this series of articles, has been one of excitement wrought with the frustration of having to go back in time to my days (albeit few) of coding and developing in C. The major obstacle I had to overcome in this endeavor was dealing with nulls; more specifically, the padding of information with nulls. After much trial and error, the most expedient mechanism I found to resolve this dilemma was to create a Binary Registry Entry filled with nulls. Once this has been created on the desired PC, the objects can read it into a blob storage area, having a bank of null characters at its disposal.

Of course, if life was only so simple. In addition to the use of null character, Palm Device datetime fields are calculated as an unsigned long representing the number of seconds since the first moment on January 1, 1904. I have no idea where this date comes from, as I've looked through the Palm programming references at my disposal, and can find no definitive answer. It is also important to note that most of the numeric data is stored as either unsigned integers or unsigned longs. While this might sound relatively trivial at the onset, the C language wades in once again to muck things up. Since most of the functions that I'll be dealing with use a blob as a temporary storage area (this is done to preserve the null characters), PowerBuilder provides no functions for the direct conversion of these numeric datatypes to Blob compatible types.

Hardware	*Software*	*Operating System*
ACER Desktop AMD-K6 160MB	PowerBuilder 7.0	Windows 98
DELL Latitude CPi D266XT 128MB	PowerBuilder 7.0	Windows NT 4.0 Workstation
Palm V	jFile v4.1	Palm OS v3.3

nv_palmdb

Much of this object's methods are concerned with the conversion of standard PowerBuilder datatypes to, and from, Blobs. Rather than go into details of the code in these methods I have listed them below:

Method	*Argument*	*Returns*	*Notes*
of_blobstr	a_s_string, a_i_len	Blob	Returns a null-padded string of length a_i_len as a blob
of_blobnull	a_s_string	Blob	Returns a null-terminated string as a blob
of_blobint	a_i_value	Blob	Returns a null-padded two-byte blob representing the unsigned integer
of_bloblong	a_l_value	Blob	Returns a null-padded four-byte blob representing the unsigned long
Of_datetime	A_dt_value	Blob	Returns a null-padded four-byte blob representing the unsigned long value for the no. of seconds a_dt_value occurs after Jan. 1, 1904.
of_blobint	A_blb	Unsigned integer	Returns an unsigned integer corresponding to the first two bytes of a_blb
of_bloblong	A_blb	Unsignedlong	Returns an unsigned long corresponding to the first four bytes of a_blb
Of_datetime	A_l_secs	Datetime	Returns a datetime value representing the number of seconds that have transpired since Jan 1, 1904.

Since the Palm Devices typically use a Motorola 68000 family processor, all of the numeric routines above are written to arrange the data with the Most Significant Byte first. Since most of us using PowerBuilder typically run on an Intel-based processor, it is important to note this difference in the byte-ordering. Additionally, we don't want to store the String() value of the number, but the actual bytes that represent the number in the correct order for the Palm Device.

Major Sections of the Palm PDB file

The Palm Database (or .pdb file) consists of five sections of information: 1) the header, 2) the record list, 3) the application information (optional), 4) the sort information (optional), and 5) the data. The header section supplies the basics of the file format: the file name (on the Palm Device), various datetime stamps, version numbers, file attributes, creator and type information, etc. The record list provides the offset and attributes for every record within the data section. The application information and sort information areas are application-specific. The final area, data, is also optional, but is rarely empty.

Palm Database header format

The header (78 bytes in length) is made of the following fields:

Field	*Bytes*	*Type*	*Notes*
Name	32	Null-terminated string	This is the name of the database on the Palm Device.
Attributes	2	Numeric	2 - Read-only 4 - Dirty AppInfoArea 8 - Backup this database 16 - Install newer over older copy 32 - Reset Palm Device after install 64 - Make database unbeamable
Version	2	Numeric	Defined by the application
Creation Date	4	Numeric	Datetime database was created. Cannot be zero. See of_datetime function above.
Modification Date	4	Numeric	Datetime database was modified. Cannot be zero. See of_datetime function above.
Last Backup Date	4	Numeric	This should be set to zero.
Modification Number	4	Numeric	This should be set to zero.
AppInfoArea	4	Numeric	The offset within the PDB file that the AppInfoArea is located. If there is none, set to zero.
SortInfoArea	4	Numeric	The offset within the PDB file that the SortInfoArea is located. If there is none, set to zero. I can find very little information on this area at this time, so I recommend not using it.
Database Type	4	String	Set this to the desired value. Generally it should match the Database Type used by the corresponding application.
Creator ID	4	String	Set this to the desired value. Generally it should match the Creator ID used by the corresponding application. For unique Create ID's be certain to register them on the Palm website.
Unique ID Seed	4	Numeric	This should be set to zero.
Next Record List ID	4	Numeric	This should be set to zero.
Number of Records	2	Numeric	This contains the actual number of records.

You'll note that a Palm Device database is limited by two factors: 1) it can contain no more than 65,535 records, and 2) the amount of memory available on the Palm Device itself.

The Record List

The record list is made up of "n" structures, where "n" represents the number of records defined in the afore-mentioned header area.

Field	*Bytes*	*Type*	*Notes*
Record Data Offset	4	Numeric	The byte offset in the PDB file at which the record is located.
Attributes	1	Numeric	16 - Secret Record bit 32 - Record in use (busy bit) 64 - Dirty record bit 128 - Delete record on next HotSync The least significant four bits are used to represent the category values.
Unique ID	3	Numeric	This should be set to zero

Putting the File Together

Area	*Notes*
Database Header	The complete header record (all 78 bytes)
Record List	This must contain at least one entry
Filler	Two bytes of null characters
App Info Area	If needed
Sort Info Area	If needed
Data	All records in sequential order

When calculating the offsets for the record list you will need to know:

1) The length of the header	78 bytes
2) The length of the record list	8 bytes * number of records
3) The two-bytes of filler	2 bytes
4) The length of the AppInfoArea	user-defined
5) The length of the SortInfoArea	user-defined

Using a structure to manage the header information, creating a new database would look like this:

```
datetime ldt = DateTime( Today(), Now() )

i_s_dbname = a_s_dbname
i_s_dbpath = ''
i_dbheader.dbname = i_s_dbname
//  8 = Backup this database, 16 = Install newer over existing copy
i_dbheader.dbflags = 8
i_dbheader.version = 1
i_dbheader.creation_time = ldt
i_dbheader.modification_time = i_dbheader.creation_time
//  i_dt_palmdate is the base Palm Device date
i_dbheader.backup_time = i_dt_palmdate
i_dbheader.modification_number = 0
//  Default value, assuming at least one record
i_dbheader.app_info_offset = 88
i_dbheader.sort_info_offset = 0
i_dbheader.dbtype = ''
i_dbheader.dbcreator = ''
i_dbheader.uniqueid = 0
i_dbheader.next_rec_id = 0
i_dbheader.rec_count = 0
RETURN 1
```

Writing the header information to a file using the blob functions referenced above would look this:

```
blob lb_data
long li, ll

// ll= FileLenth( i_s_dbpath + i_s_dbname )
li = FileOpen( i_s_dbpath + i_s_dbname + '.pdb', StreamMode!,
Write!, LockWrite!, Replace! )

//      All datetime values in the Palm OS are specified as the
//      ULongs which store the number of seconds elapsed since
//      January 1, 1904.

i_b_dbdata = of_blobstr( i_dbheader.dbname, 32)
i_b_dbdata += of_blobint( i_dbheader.dbflags )
i_b_dbdata += of_blobint( i_dbheader.version )
i_b_dbdata += of_datetime( i_dbheader.creation_time )
i_b_dbdata += of_datetime( i_dbheader.modification_time )
i_b_dbdata += of_datetime( i_dbheader.backup_time )
i_b_dbdata += of_bloblong( i_dbheader.modification_number )
i_b_dbdata += of_bloblong( this.of_appinfoarea() )
i_b_dbdata += of_bloblong( i_dbheader.sort_info_offset )
i_b_dbdata += blob( i_dbheader.dbtype )
i_b_dbdata += blob( i_dbheader.dbcreator )
i_b_dbdata += of_bloblong( i_dbheader.uniqueid )
i_b_dbdata += of_bloblong( i_dbheader.next_rec_id )
i_b_dbdata += of_blobint( i_dbheader.rec_count )
i_b_dbdata += a_blob

FileWrite( li, i_b_dbdata )
FileClose( li )
RETURN 1
```

The above function allows for a blob to passed containing the rest of the database information. I will refine this approach in the future, but for now it works.

nv_jFiledb

jFile is one of several "flat-file" database managers available for Palm Devices. The main reason I chose to utilize them over any other was the willingness of the developer to support me during the development of this article. jFile makes extensive use of the Palm databases AppInfoArea. The AppInfoArea is used to store the definition of the columns within the database including name, type, sorting, searching, and popup list box definitions. Note: Popup list boxes on the Palm Device are similar to PowerBuilder's DropDownData-Windows or DropDownListBoxes.

AppInfoArea

The major difference between custom Palm Device databases and jFile databases is that all data stored in a record is stored as null-terminated string. Let's begin by examining the AppInfoArea of a jFile database.

Field	*Bytes*	*Type*	*Notes*
Name[]	21	Null-terminated string	The name of the column, this is an array of 50 values, no more and no less!
Type[]	2	Numeric	An array of 50 values containing one of the following codes: 1 - String (default) 2 - Boolean (Checkbox) 4 - Date 8 - Integer 16 - Float 32 - Time 64 - Popup list 65 - Auto Record Creation Date 66 - Auto Record Creation Time 67 - Auto Integer 71 - Auto Record Modification Date 72 - Auto Record Modification Time 73 - Multi select Popup list
Field Count	2	Numeric	The number of fields used in the database
Version	2	Numeric	The version of this database, this should be set to 576 for version 1.0d of jFile
Column Widths[]	2	Numeric	This is an array of 50 values. The width of the columns in Palm Device units. A typical Palm Device is 160 units wide. jFile supports automatic horizontal scrolling of database information.
Sort Field[]	2	Numeric	This is an array of 3 values. The no. of the column(s) used for sorting. For descending sort order subtract the column number from 65536.
Field Field	2	Numeric	The number of the column used in the find.
Filter Field	2	Numeric	The number of the column used for filtering.
Find String	16	String	The string used in the last find
Filter String	16	String	The string used in the standard filter command
Flags	2	Numeric	1 - Close the database on application exit 2 - View only database 4 - Not in use 8 - Structure Locked 16 - Private DB 32 - Filtering (internal use only)
First Column	2	Numeric	The number of the first column to show beyond the first column in the database. This should be zero by default
Extra Data[	4	Numeric	This is an array of 50 values. The data represents the current increment number for auto increment integer fields
Extra Data2[	4	Numeric	
Popuplist	??	Null-terminated strings	Storage area for popup list values. See explanation below.
Filler	13	Null-terminated string	'EndJFileData'

Popup Lists are an indeterminate chunk of memory in the above definition. If there are no popup lists in the database, then this area will contain 4 null characters. Each list is a continuous sequence of null terminated string that hold the contents of the items in the list. Each list is prefixed with the null-terminated string "popupX," where X is a, b, c, … z, A, B, C, … X for fields 1, 2, 3, … 26, 27, 28, 29, … 50. The entire popup list is terminated by a single null character as well.

Example: popupb\0First 2\0Second2\0popupd\0First 4\0Second 4\0Third 4\0\0

This example creates two popup lists, one for the second field (b) and another for the fourth field (d). The first popup list contains two items: First 2, and Second 2. The second popup list contains three items: First 4, Second 4, Third 4. Here is an example of how the AppInfoArea would be built:

```
lj = i_jfheader.numfields
FOR li = 1 to lj
        l_blob += of_blobstr( left(i_jfheader.field_names[li],20), 21 )
NEXT
FOR li = lj+1 to i_l_max_fields
        l_blob += of_blobstr( '', 21 )
NEXT
FOR li = 1 to lj
        l_blob += of_blobint( i_jfheader.field_types[li] )
NEXT
FOR li = lj+1 to i_l_max_fields
        l_blob += of_blobint( 1 )
NEXT
l_blob += of_blobint( i_jfheader.numfields )
l_blob += of_blobint( i_l_current_version )
FOR li = 1 to lj
        l_blob += of_blobint( i_jfheader.col_widths[li] )
NEXT
FOR li = li to i_l_max_fields
        l_blob += of_blobint( 80 )
NEXT
l_blob += of_blobint( i_jfheader.datawidth )
lj = MIN( UPPERBOUND( i_jfheader.field_sort ), 3 )
FOR li = 1 to lj
        l_blob += of_blobint( i_jfheader.field_sort[li] )
NEXT
FOR li = li to 3
        l_blob += of_blobint( 0 )
NEXT
l_blob += of_blobint( i_jfheader.field_find )
l_blob += of_blobint( i_jfheader.field_filter )
l_blob += of_blobstr( i_jfheader.find_str, 16 )
l_blob += of_blobstr( i_jfheader.filter_str, 16 )
l_blob += of_blobint( i_jfheader.flags )
l_blob += of_blobint( i_jfheader.col_first )
FOR li = 1 to lj
        l_blob += of_bloblong( i_jfheader.field_extra[li] )
NEXT
FOR li = li to i_l_max_fields
        l_blob += of_bloblong( 0 )
NEXT
FOR li = 1 to lj
        l_blob += of_bloblong( i_jfheader.field_extra2[li] )
NEXT
FOR li = li to i_l_max_fields
        l_blob += of_bloblong( 0 )
NEXT
lj = UPPERBOUND( i_jfheader.popuplists )
IF lj > 0 THEN
        FOR li = 1 to lj
                l_blob += of_blobnull( 'popup'+char( &
                        ii+i_jfheader.popuplists[li].field ) )
                jj = upperbound( i_jfheader.popuplists[li].slist )
                FOR ji = 1 to jj
                        l_blob += of_blobnull( i_jfheader.popuplists[li].slist[ji] )
                        NEXT
                        l_blob += of_blobstr( '', 1 )
                NEXT
        ELSE
                l_blob += of_blobstr( '', 4 )
        END IF
        l_blob += of_blobnull( 'EndJFileData' )
```

Data Area

The records themselves are also stored in a unique format. The beginning of the record contains an array of 2 byte integers containing the length of the data, including the null-terminator, of the used fields. Note that this is only the fields used, and not an array of 50 values as we saw in the AppInfoArea. This array is then followed by a series of null-terminated strings containing the actual data in the field.

Sample Data Area: Containing Seven Fields

The first 14 bytes in the record (in hexadecimal):

```
00 06 00 02 00 0B 00 09 00 07 00 02 00 04
```

This is immediately followed by the actual data:

```
Data1\01\003/22/2000\01 1:18 pm\0Popup1\01\01.0\0
```

Here is an example for the building of the data area. Note this also include the building of the record list, since the record lengths are calculated at the same time.

```
lj = UPPERBOUND( i_jfheader.recs )
FOR li = 1 to lj
        l_bloblist += of_bloblong( lr ) + of_bloblong( 0 )
        // l_bloblist stores the Palm Device record list
        jj = UPPERBOUND( i_jfheader.recs[li].data )
        // each record should actually contain the same number of
        // columns
        l_blobrec = BLOB('')
        FOR ji = 1 to jj

                CHOOSE CASE
ClassName(i_jfheader.recs[li].data[ji] )
```

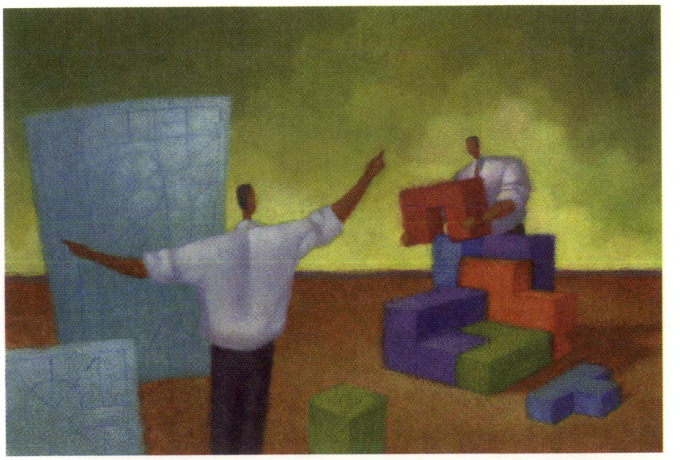
```
    CASE "date"
      l_blobrec += of_blobint( INT( &
      len( STRING( i_jfheader.recs[li].data[ji], i_s_date_fmt ) ) + 1 ) )
    CASE "time"
      l_blobrec += of_blobint( &
      INT( len( STRING( i_jfheader.recs [li].data[ji], i_s_time_fmt ) ) + 1 ) )
    CASE ELSE
      l_blobrec += of_blobint( INT( &
      len( STRING( i_jfheader.recs [li].data[ji] ) ) + 1 ) )
    END CHOOSE
  NEXT
  FOR ji = 1 to jj
    CHOOSE CASE ClassName(i_jfheader.recs[li].data[ji] )
      CASE "date"
        l_blobrec += of_blobnull( &
          STRING( i_jfheader.recs [li].data[ji],i_s_date_fmt ) )
      CASE "time"
        l_blobrec += of_blobnull( &
          STRING( i_jfheader.recs [li].data[ji],i_s_time_fmt ) )
      CASE ELSE
        l_blobrec += of_blobnull( &
          STRING( i_jfheader.recs [li].data[ji] ) )
      END CHOOSE
    NEXT
    lr += Len( l_blobrec )
    l_blob += l_blobrec
  NEXT
```

Putting It All Together

Creating a jFile Database

Now that we've got the basic concepts behind us, let's look at an example of how we can create a Palm Device jFile compatible database:

```
// create a jFile Palm Database
lnv_palmdb = CREATE nv_jFiledb
// Name it Test, and give it seven columns
lnv_palmdb.of_dbnew( 'Test', &
    { 'Field1', 'Field2', 'Field3', 'Field4', 'Field5', 'Field6', 'Field7' }, &
    { lnv_palmdb.i_l_ft_string, lnv_palmdb.i_l_ft_boolean,
lnv_palmdb.i_l_ft_date, lnv_palmdb.i_l_ft_time,
lnv_palmdb.i_l_ft_popup, lnv_palmdb.i_l_ft_int,
lnv_palmdb.i_l_ft_float } )
// Define the popup list for the fifth column
lnv_palmdb.of_popup( 5, { 'Popup1' } )
// Put some data into records
FOR ii = 1 to 7
    data[1] = 'Data' + STRING( ii )
```



```
    data[2] = 1 * ii
    data[3] = Today()
    data[4] = Now()
    data[5] = 'Popup1'
    data[6] = 1 * ii
    data[7] = 1.0 * ii
    lnv_palmdb.of_record( data )
NEXT
// Write the database out
lnv_palmdb.of_dbwrite( blob('') )
```

Installing the Database on the Palm

Once the database has been created, you can push it to the Install Directory. Here is an example:

```
// This will install the file for the last user to Hot Sync on this PC
IF FileExists( 'Test.PDB' ) THEN
    IF lnv_palmdb.of_isvalid() THEN
        lnv_palmdb.of_readuserdata()
        lnv_palmdb.of_installuser( lnv_palmdata.of_
            getlastusernumber(), 'Test.PDB' )
        MessageBox( 'jFile Create', 'Your jFile Test DB has been
            installed' )
    END IF
END IF
DESTROY lnv_palmdb
```

Coming Up Next Time

In the next issue, I'll be taking this concept further. I'll be using OrbWorks PocketC with CControls to actually build an application running on the Palm, using a downloaded database extracted from a PowerBuilder Datawindow!! Keep checking my website (www.lamb.tj), as I will be posting the objects there as I work on them. ■

It's Time to Join ISUG

By Joe Burger, ISUG Membership Director

As I registered for TechWave 2000, my adrenaline increased as I recalled the excitement of last year's conference. Disney World, talking with the movers and shakers of Sybase, seeing old conference buddies, experiencing the best and newest technology products, the Event, relaxing at the Hospitality Suites, meeting people from around the world, and of course, the Exhibit Hall, with all the great new products and the giveaways!

Imagine this: a giveaway that you can get without even attending the conference, which you are guaranteed to win. It's not a stress ball, or a T-shirt, or even boxer shorts (don't laugh, I have three pairs). Even better, it doesn't sit on your desk, collecting dust, but gives you benefits throughout the year. Yes, it's an ISUG membership.

Getting that Return On Your Investment

Hard to justify to your boss, you say? Too busy to sign up? In five minutes you can become a member. It's as easy as www.isug.com. And how hard is free to justify? How about getting 10% off the PowerBuilder 7 or ASE 12 or EAStudio class you are planning to attend? That $75 investment could result in savings of double or even triple the return. The last class I attended saved my company $220! Believe me, it is much easier to get an item approved when you are doing your best to contain costs.

The More The Merrier

Recently, a co-worker needed a course for our cross-training strategy. The savings provided additional justification to attend the class and he got a free ISUG membership along with the valuable education he needed. A person I met last year at the conference was having trouble getting his company to pay for the membership fee. "No problem," I said. "Just show your boss the $75 discount for ISUG members on the conference registration form. That means your membership is free."

Just The Beginning of the Benefits

Now that you are a member, it's time to make the most of it. Order the SQL Anywhere Studio (worth $399). Take that certification exam with a savings of 20%. Add your voice by voting on product enhancements. Get involved in your Local User Group for leadership, networking and to pass on your knowledge. Cruise through the *ISUG Technical Journal* for articles on product tips and information.

Joining at TechWave

Become an ISUG member at TechWave and you will get the conference CD, containing those presentations you wish you had time to attend. That alone is easily worth the $75 and you will get the member discount at next year's conference. Be proud, you are now part of a group of over 3,000 professionals from over 40 countries! ■

New Programs, Expanding Membership During 2000

By Kathy Ridley, ISUG Vice President

If you have ever served on a board of directors, you know that it has its challenges. We on the ISUG board are working to bring our members new benefits and exciting programs.

A New CD and Board Appointments

Early in the year, the Board met at the Emeryville, CA, Sybase office. During this meeting we decided to produce ISUG's first Techwave Conference CD, loaded with all the conference proceedings and an electronic version of the prior year's *ISUG Technical Journal*. This CD has proved so popular that ISUG will produce it for future TechWave conferences.

During that meeting, we also appointed our members-at-large. These are special individual who have contributed to Sybase for many years and ISUG uses them as advisors and for special committees. This year Peter Thawley was appointed: He has spoken for users' groups for many years. Jeff Roberts, the PBUG users group president in Atlanta, has now taken on the role of Electronic Media Director, following Michael Peppler's resignation from that position.

New Programs Implemented in 2000

The board's second-quarter meeting took place in May at Boulder, CO, in the Sybase office. During this meeting we approved a new Partner Program that allows ISUG to serve as a vehicle for vendors and service providers to participate in user group events and to mail product information to our worldwide membership, plus have the ability to conduct user surveys, web site profiles, journal articles, and web links.

Another new program is the ISUG Advocate program, in which ISUG teams with Sybase to discuss and review Sybase's products as well as become customer references. ISUG will be sponsoring a European road show later in the year. We have also been asked to attend key Asian user group events in 2000.

At this meeting, ISUG finalized preparation for the TechWave 2000 conference in Orlando, Florida. We completed paper selection activities with the help of few key volunteer members, made arrangements for the Special Interest Group (SIG) meetings, selected our booth, decided on conference giveaways, and arranged for the User Group Presidents lunch. We are looking forward to seeing our members there.

ISUG also reviewed its enhancements process. Sybase has implemented 50% of our existing request and we are hoping to get more feedback from users and Sybase product marketing. Remember, members can always submit an enhancements request online at www.isug.com.

A Growing ISUG Membership

Over the past year, ISUG membership continued to expand. We had 800 members join while registering for TechWave 1999 (this conference included a free ISUG membership). This year, we have implemented a new membership application on the web with secured transactions for credit card processing. Very soon, ISUG will become a portal on MySybase, as we continue to achieve our goal to act as a conduit of information between our members and Sybase.

Stop by and visit the ISUG booth in Orlando, Florida during the North America conference! ■

INTERNATIONAL

Sybase User Group

ISUG Membership Application

ISUG Benefits

- Free Copy of SQL Anywhere Studio
- EA server with PowerJ evaluation CD
- ASE 12.0 Technical Documentation CD
- A subscription to the quarterly ISUG Technical Journal
- Access to the online ISUG membership directory
- Access to the online ISUG Enhancement Request process
- Technical Training at local user groups
- TechWave Conference proceedings CD

Education Discounts

- Save 10% on Sybase education classes
- Save 20% on Certification Exams from Sylvan Prometric
- $75 discount on registration for TechWave and other ISUG-affiliated conferences

Note: Benefits are subject to change. Please refer to the ISUG website (www.isug.com) for current list.

please check one:

❑ individual membership*†	US	$ 75.00
❑ corporate membership*† (10 people)	US	$500.00
❑ government subscription*	US	$ 75.00

* Requires Sybase, Inc. Site ID or CBSS number.

† Members joining from Germany, the United Kingdom, Belgium, Luxembourg, the Netherlands, or France must join through their local user group. See LUG Directory for contact information.

Sybase Site ID or CBSS number (includes application & middleware product Site ID or CBSS number)

number _____

Your Site ID (CBSS number), found on your invoice or packing slip, is a 5-digit number followed by "- # -#".

please fill in the following contact information completely:**

name _____

title _____

phone _____

fax _____

email _____

company/organization _____

department _____

address _____

mailstop _____

city _____ state _____

postal code _____ country _____

If this is a corporate membership, please attach a separate sheet with the above information for nine additional persons.

Sybase products•

version platform

1. _____
2. _____
3. _____

•**Note:** *This information will be kept confidential.*

Are you a member of a Local User Group (LUG)? ❑ Yes ❑ No

Name of group _____

❑ Please send me information on the LUG in my area.

Are you a member of a Special Interest Group (SIG)? ❑ Yes ❑ No

name of group _____

I am most interested in the following SIGs:

- ❑ Adaptive Server
- ❑ Application Tools
- ❑ Architecture & Design
- ❑ Query & Reporting Tools
- ❑ Data Warehousing
- ❑ NT Server
- ❑ Systems Management
- ❑ VLDB
- ❑ WWW/Internet
- ❑ Middleware
- ❑ PowerBuilder

payment instructions

Please send a check made payable to "International Sybase User Group." Outside North America, please send a check for the currency equivalent.

Note: *All checks must be drawn from a US bank.*

To pay by credit card, please fill in the following information:

❑ VISA ❑ MASTERCARD

card number _____ exp. date ____

cardholder signature _____

Return this form with check or credit card information by enclosing in an envelope and applying stamp.

membership directory release form

Contact information will be distributed in the membership directory to ISUG members only. Product information will not be released. Please sign this form if you want your contact information published in the ISUG directory.

signature _____ date _____

non-disclosure agreement

ISUG members agree not to use any Sybase confidential information for any purpose except their business relationship with Sybase. Members also agree that they will not disclose confidential information to any person other than their company's employees who are directly involved in the use of Sybase products.

signature _____ date _____

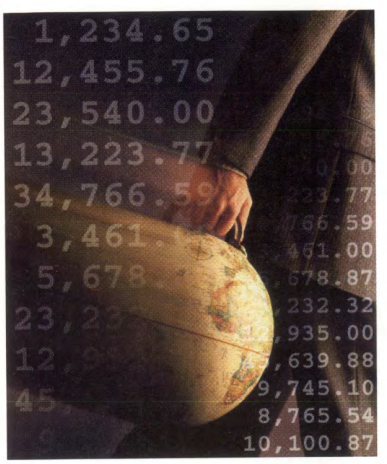
Mail form to...

International Sybase User Group
6475 Christie Avenue
Emeryville, CA 94608
Fax: 510-922-0882
E-mail: isug@sybase.com

Note: Members joining from Germany, the United Kingdom, Belgium, Luxembourg, the Netherlands, or France must join through their local user group. See LUG Directory for contact information.

Calendar of Events

User Group Meetings and Events



Atlanta
User Meeting: 3rd Thurs. each month

Baltimore Area PBUG
User Meeting: 1st Wed. of even months

Birmingham PBUG
User Meeting: 2nd Thurs. each month

Central Ohio PBUG
User Meeting: 1st Tues. of even months

Charlotte PBUG
User Meeting: 1st Thurs. each month

ChicagoLand Sybase Tools User Group
User Meeting: 1st Tues. each month

TechWave North America/Europe 2000 Conference
Orlando, Florida
July 30-August 2, 2000

Colorado PBUG
User Meeting: Fourth Wed. bi-monthly

Des Moines PBUG
User Meeting: 1st Tues. of odd months

DFW PBUG
User Meeting: 2nd Tues. of each month

East Michigan PBUG
User Meeting: 2nd Thurs. of odd months

Houston
User Meeting: 3rd Thurs. of even months

Los Angeles
User Meeting: 1st Tues. of each month

Louisiana PBUG
User Meeting: 2nd Wed. bi-monthly

Music City Powersoft UG
User Meeting: 3rd Tues. each month

New Jersey PBUG
User Meeting: 3rd Thurs. of each quarter

North West PBUG
User Meeting: 2nd Tuesday of odd months

Northwest Arkansas Area PBUG
User Meeting: Last Mon. of each month

Oklahoma City
User Meeting: 1st Wed. each month

Orange County
User Meeting: 3rd Thurs. of odd months

Orlando PBUG
User Meeting: 3rd Wed. of each month

Ottawa PBUG
User Meeting: 2nd Wed. each month

PBUG of Northeast Ohio
User Meeting: 2nd or 3rd Wed. of odd months

Philadelphia Sybase IAD User Group
User Meeting: 2nd Wed. each month

Portland, Oregon PBUG
User Meeting: July 12, Sept. 6

Riverside Area PBUG
User Meeting: 3rd Wed. of each month

Rocky Mtn/Denver/Salt Lake City
User Meeting: 2nd Tues. each month

SNUG
User Meeting: 1st Wed. of odd months

St. Louis
User Meeting: 3rd Tues. of odd months

INTERNATIONAL

Sybase User Group

Let us know what your user group is doing! Contact Sybase User Group Marketing with the dates and locations of your group's meetings and special events. They can be reached at: Phone 510-922-7525, Fax 510-922-0882 or Email isug@sybase.com

Board of Directors

President

Thomas J. Lamb
Automated Data Sciences
Phone: 815-621-5262
t.lamb@bigfoot.com

Vice President

Kathy Ridley
CGI
Houston, TX
Phone: 713-868-5537, x160
kridley@drthou.com

Secretary

Karen Pursch
Sybase
Emeryville, CA
Phone: 720-820-0188
karenk@sybase.com

Treasurer

Luc Van der Veurst
Academic Hospital, VUB
Brussels, Belgium
Phone: 32-2-477-6980
lucv@az.vub.ac.be

Conference Director

Cynthia Gill
CGI
Long Beach, CA
Phone: 562-421-2070 x217
cynthia_gill@drthou.com

ISUG Technical Journal Director

Teresa A. Larson
VetCentric.com
Phone: 410-571-6790
tlarson@vetcentric.com

Enhancements Co-Directors

Jibu Abraham
Jones Cyber Solutions
Phone: 303-784-3612
jibu_abraham@hotmail.com

Jaideep Chawla
Pricewaterhouse Coopers
Phone: 703-645-5882
Jaideep.Chawla@us.pwcglobal.com

Membership Director

Joe Burger
The Longaberger Company
Newark, OH
Phone: 740-322-5091
jburger@longaberger.com

Australasia RUG Director

Anthony Mandic
Mandic Consulting Pty, Ltd.
Sydney, Australia
amandic@start.com.au

Electronic Media Director

Jeff Roberts
CCS Consulting
Phone: 770-582-0360
Fax: 770-582-0590
jroberts@consultCCS.com

Members-at-Large

Linda Morison
Automated Data Sciences
Phone: 202-205-6717
Fax: 202-205-7064
Lmorison@erols.com

Peter F. Thawley
InterTrust
Phone: 408-855-0100
Fax: 408-222-6144
Peter@intertrust.com

Partner Membership Director

Frank Monteverdi
CGI
Phone: 713-868-5537x149
Fax: 713-868-4014
Frank_monteverdi@drthou.com

European RUG Director

Dorus Kruse
Belastingdienst
Automatiseringscentrum
The Netherlands
Phone: 31 55 528 7945
doruskruse@hotmail.com

North American RUG Director

Cindy Bean
BMC Software, Inc.
Phone: 713-918-1841
Fax: 713-918-1173
cynthia_bean@bmc.com

SIG Co-Directors

David Johnstone
Bank of Scotland
Phone: 44-0131-442-7957
david_johnstone@bankofscotland.co.uk

Jay Hunt
Automated Data Sciences
Phone: 618-567-1177
Fax: 618-692-1188
djayhunt@charter.net

Sybase Contact:

Yared Benyam
User Group Marketing
Emeryville, CA
Phone: 510-922-7525
Fax: 510-922-0882
isug@sybase.com

Sybase User Group Directory

European Region

Austria
Dr. Wolfgang Lipa
ZAMG
Phone: 43-0222-36-44-53 x 2603
Fax: 43-0222-369-1233

Belgium "Blues"
Luc Van der Veurst
Academic Hospital VUB
Phone: 32-2-477-6980
lucv@az.vub.ac.be

Denmark
Lars Henrichsen
COWI Consult A/S
Phone: 45-45-972068
or 45-45-972211
Fax: 45-45-972212
lh@cowi.dk

Estonia
Gustav Seier
Tallinn, Estonia
Phone: 372 6655467
gustav.seier@eyp.ec

Finland
Kristiina Salminen
Sanoma Corporation
Phone: 358 9 122 87 3184
kristiina.salminen@sanoma.fi

France "Fibonacci"
M. Gerard Lledo
C.E.A.
CENTRE ETUDES SACLAY
Phone: 33-1-6908-9616
Fax: 33-1-6908-9608

Germany

Sybase User Group
Jens Timmermann
TOPOLOGIX GmbH
Phone: 49-40-352-253
Fax: 49-40-340-340
jens.timmermann@topologix.com

PowerBuilder User Group
Ludwin Feiten
Power People
Vorstand@pbugg.de
www.pbugg.de

Ireland
David Quirke
Phone: 00 353 1 677 6777
Fax: 00 353 1 6776614
dquirke@sybase.ie

The Netherlands
Jolanda Zwaan
Phone: 036 5226249
Fax: 036 5226249
dsug@tref.nl

Norway
Jan Kare Fjeldstad
Aker Elektro
Phone: 47-53-49-23-55
jkf@ael.aker.no

Spain
Manuel Blanco Ulled
Unipapel, S.A.
Phone: 91-806-96-10
Fax: 91-803-52-22
mblanco@unipapel.com

Switzerland
Dr. Roberto Buzzi
Zurich Versicherungs
Phone: 41-1-205-2121
Fax: 41-1-205-3375
chzurkrb@ibmmail.com

Turkey
Levent Sensezgin
Sybase Turkey
Phone: 90-212-284-8339
Fax: 90-212-284-8342
levent.sensezgin@sybase.com.tr

United Kingdom
Khosro Parnian
Sensitek, Ltd.
Phone: +44-0171-325-0374
parnian_khosro@jpmorgan.com

UK PowerBuilder User Group
Anne Bocock
PB Associates
Phone: 020-84213533
Fax: 020-84201420
anne@pbug.co.uk

International

Argentina
Sergio Di Cuffa
Sybase Argentina
Phone: 54-1-393-0421
Fax: 54-1-326-7039

Australia
Igor Geninson
IBS Technology
Phone: 61-02-388-7675
Fax: 61-02-388-8067

Brazil
Luiz Guilherme Mendonca
LOCUS Ltd.
Phone: 55-21-553-3086
Fax: 55-21-551-8461
lguilherme@locus.com.br

Chile
Jean Pierre Lefranc
Sybase Chile
Phone: 56-2-3306700
Fax: 56-2-3306800

New Zealand
Shayne Duncan
Sybase NZ, Ltd.
Phone: 64-4-473-3661
Fax: 64-4-499-9068
shayne@sybase.com

Saudi Arabia
A. Omran
Al-Omran
Phone: 96-61-4662373
Fax: 96-61-4662502

South Africa
Deirdre Martin
c367455@sn2.edsa.co.za

Thailand
Chakkrapong Tongsak
Bangkok University
Fax: 65-273-9159
ct@lily.bu.ac.th

United Arab Emirates
George Khouri
Sybase Products - Middle East
Abu Dhabi
Phone: 97-1-2-325-911
Fax: 97-1-2-340-850

Sybase Local User Groups of North America

Northeast Region

Regional Contact
Jan Barcelou
Barings Asset Management
Phone: 617-946-5325
Fax: 617-946-5410
Jan.barcelou@baring-asset.com

Albany, New York PowerBuilder User Group
Greg Fisher
gfisher@dttus.com
www.anypbug.org

Boston (SNEKUS–Sybase North East Kingdom Users Society)
Bruce Driver, Consultant
Phone: 508-443-3310
bdriver@btrgroup.com or
bdriver@csi.com
www.snekus.org

Harrisburg, PA, PowerBuilder User Group
Dennis Rehm
dennisrehm@appliedcomputing.net

New Jersey
Bob Munson
Phone: 973-367-3634
robert.munson@prudential.com

New Jersey PowerBuilder User Group
Boris Gasin
Phone: 973-402-5600
bgasin@dynamictechgroup.com
www.njpbug.org

Ontario Sybase User Group
Chris Baker, Visual Systems Development Group
Phone: 416-591-0005 x245
OSUG@interlog.com
www.interlog.com/~osug

Ottawa
Tony Antonallo
Phone: 613-798-7507

Ottawa PowerBuilder User Group
Carole Hargrave
Saphire Consulting Services, Inc.
Phone: 613-292-2018
opbug@opbug.com
www.opbug.com

Philadelphia
Sybase, Inc.
Phone: 610-260-4300
Fax: 610-260-4399

Philadelphia Sybase IAD User Group
Chuck Miller
cmiller@appltech.net
pbug.fcg.com/PBUG/index.html

Toronto PowerBuilder User Group
Paul Bis
Tpbug@interlog.com
www.interlog.com/~tpbug

Southeast Region

Atlanta PowerBuilder User Group
Jeff Roberts
CCS Consulting
Phone: 770-582-0360
Fax: 770-582-0590
jroberts@consultccs.com
www.apbug.org

Baltimore Area PowerBuilder User Group
Debbie Arczynski
softquip@home.com
www.geocities.com/bapbug

Baltimore/Washington, D.C.
Keith Altman
Dcasug@dgsys.com
www2.dgsys.com/~dcasug

Birmingham PowerBuilder User Group
Dennis Marcum
Dennis_marcum@protective.com
www.highland-consulting.com/bpbug/index.html

Boca Raton
John Glover
Data Breeze
Phone: 954-427-4416 x309
Fax: 954-427-0280
jdg@phamis.com

Charlotte PowerBuilder User Group
Alan Augustine
aaugustine@ttpartners.com
www.charpbug.org

Louisiana PowerBuilder User Group
Joe Sherrill
joesherrill@msn.com
www.notrs.com/lapbug

Miami
Ray Liera
University of Miami
Phone: 305-284-4207
Fax: 305-284-4753
Rleira@miami.edu

Music City Powersoft User Group
Jeff Gibson
jgibson@frontrunner.com
www.frontrunner.com/mcpug

Nashville
Jimmy Hogan
Phone: 615-641-5550
Fax: 615-841-5556

North Carolina
Clay Bullard
SMCI
Phone: 704-375-5788
Fax: 704-375-5699
smcic@ix.netcom.com

Northwest Arkansas Sybase User Group
Dan Luttrell
vicsinc@mindspring.com
www.cyberspace.org/~nwapbug

Orlando PowerBuilder User Group
Bryan Enochs
AT&T
Phone: 407-858-8320
Fax: 407-858-8352
benochs@ao.net
www.wedowebs.com/opbug

Piedmont Triad PowerBuilder User Group
James Stoertz
jstoertz@fitech-gso.com

Tampa Bay Area
Jeffrey Garbus
Phone: 813-949-7016
Jeffg@soaringeagleltd.com

Washington Area Sybase Enterprise Application Tools User Group (WA/SEATUG)
Theo Rushin, Jr.
National Institutes of Health
Phone: 301-402-4609
Fax: 301-480-6105
rushint@mail.nih.gov

North Central Region

Regional Contact
Gaynel Walden
AT&T
Phone: 816-995-3723
gaynel@att.com

Central Ohio PowerBuilder User Group
Barry McDonald
bmcdonald@CBSInc.com
www.cmnpbug.com

Chicago Sybase User Group
John Da Silva
Power 2000
Phone: 630-369-7175
Fax: 630-369-8042
jdasilva@power2000.com
www.csug.com

ChicagoLand Sybase Tools User Group
Michael Baraz
Phone: 630-235-4529
mbaraz@tradesolutions.com
www.cpbug.org

Cincinnati/Dayton
Scott Stegman
Phone: 513-241-5949
Fax: 513-241-6731
sstegman@clientserver.com

Des Moines PowerBuilder User Group
Mary Eagan
meagan@ifmc.org

East Michigan PowerBuilder User Group
Benny Cheung
benny@mpbug.org
www.mpbug.org

Indianapolis
Julie Clark
Phone: 317-501-7559
jclark@moser-inc.com
www.cs.bsu.edu/homepages/sam/isug

Michigan PowerBuilder User Group
Jagdish Karira
Phone: 248-524-3280
jagdish@techie.com

Minneapolis, MN
Maendra Saraswate
Hunter Software
Phone: 612-943-3986
Fax: 612-941-0933
msarasw@primenet.com

PowerBuilder User Group of Northeast Ohio
Todd Senauskas
tsenauskas@datavantage.com
www.internet-club.com/usa/neopbug

St. Louis Sybase Internet Tools User Group
Dave Blankenship
president@stlpbug.org
www.stlpbug.org

Twin Cities Sybase Tools User Group
Bob Sharp
Phone: 612-483-1506
www.tcpbug.com

Winnipeg
Jim Novisat
Great–West Life Assurance Co.
Phone: 204-946-7748
Fax: 204-946-4567
jzn@gwl.ca

Wisconsin Sybase User Group
Michelle Murphy
Wisconsin Electric Power Co.
Phone: 414-221-2068
Fax: 414-221-4744
michelle.murphy@wepco.com

Bill Mitchell
Northwestern Mutual Life
Phone: 414-299-4022
Fax: 414-299-1686
Billmitchell@northwesternmutual.com
www.reveregroup.com/wisug

SW-Mountain Region

Regional Contact
Cindy Bean
BMC Software, Inc.
Phone: 713-918-1841
Fax: 713-918-1173
cynthia_bean@bmc.com

Austin
Rob Gustavson
BCS Systems, Inc.
Phone: 512-921-0074
Rgustavson@rfdinc.com

Colorado PowerBuilder User Group
Mark Juodawlkis
mjuodawlkis@gr.com
www.co-pbug.org

Colorado Springs PowerBuilder User Group
Judith McEntee
jmcentee@gr.com
www.camelotconsulting.com/cspbug

Dallas Sybase User Group
Phil Adams
FireSteed Software, Inc.
Phone: 817-296-6238
Fax: 817-282-8980
pcadams@firesteedsoftware.com
www.firesteedsoftware.com/tsug.html

Dallas/Ft. Worth PowerBuilder User Group
Marcie Jones
marciejones@yahoo.com
www.dfwpbug.com

Houston (Sybase Users' Group of Texas)
Tony Broussard
TradeCapture.com
Phone: 713-752-4063
admin@sugtx.org
www.sugtx.org

Inter-Mountain/Utah
Dianne Garcia
Discover Brokerage Direct
Phone: 801-902-4222
Garcia@discoverbrokerage.com

Kansas City Area Sybase User Group
George Meiers
Hallmark Cards, Inc.
Phone: 816-545-6395
Gmeier1@hallmark.com

Oklahoma City PowerBuilder User Group
Dianna Demotto
okcpbug@usa.net
okcpbug.iwarp.com

Rocky Mountain/Denver/ Salt Lake City
Doug Thomas
Time Warner Telecom
Phone: 303-566-1432
Fax: 303-566-1400
dougtho@geocities.com
rmsug.intranets.com

San Antonio Sybase User Group
Bill Purnell
purnell@texas.net

San Antonio PowerBuilder User Group
Richard Carrier
rcarrier@tesoropetroleum.com
www.pfccheatsheet.com/pbugsa.htl

Tulsa PowerBuilder User Group
Mike Deasy
mdeasy@energy.twc.com
www.geocities.com/tulsa_pbug

Wichita Kansas PowerBuilder User Group
Rodney Hughes
Phone: 316-517-1847
rjhughes@cessna.textron.com

Far West Region

Regional Contact
Cynthia Gill, CGI
Phone: 562-421-2070 x217
cynthia.gill@cgiusa.com

Arizona Sybase/Powersoft User Group
Reny Abrego
Phone: 602-402-6491
reninaz@aol.com
azspug.webjump.com

Hawaii
Sterling Yee
Hawaiian Electronic Industries
Phone: 808-532-5870
Fax: 808-532-5828
syee@hei.com

Los Angeles
Cory Isaacson, Compuflex
Phone: 818-772-7990
Fax: 818-772-7999

Los Angles PowerBuilder User Group
Brian J. Smith
Phone: 310-221-0798
75530.332@compuserve.com

Orange County PowerBuilder User Group
Victor A. Reinhart
victora.reinhart@phs.com
www.ocpbug.com

Riverside Area Poweruilder User Group
Joe Fontaine
Riverside Cty. Dept. Information Technology
Phone: 909-955-3692
Fax: 909-955-9390
jfontain@pe.net
www.pe.net/!jfontain/rapbug.html

Northwest Region

Regional Contact
Matt Townsend, Ayupp Inc.
Phone: 707-829-1173
Matt_townsend@earthlink.net

Boise
Michelle Featherston
Micron Semiconductor
Phone: 208-368-4498
Fax: 208-368-1043
mfeatherston@micron.com

Calgary/Edmonton
David Owen
dowen@trimac.com

Edmonton PowerBuilder User Group
Theodore J. Allen
73614.3174@compuserve.com
www.freenet.edmonton.ab.ca/~vschmid/pbug.htm

Northwest PowerBuilder User Group (Seattle)
Anirban Choudhury
anirban@iname.com
www.cascadia-sw.com/nwpbug

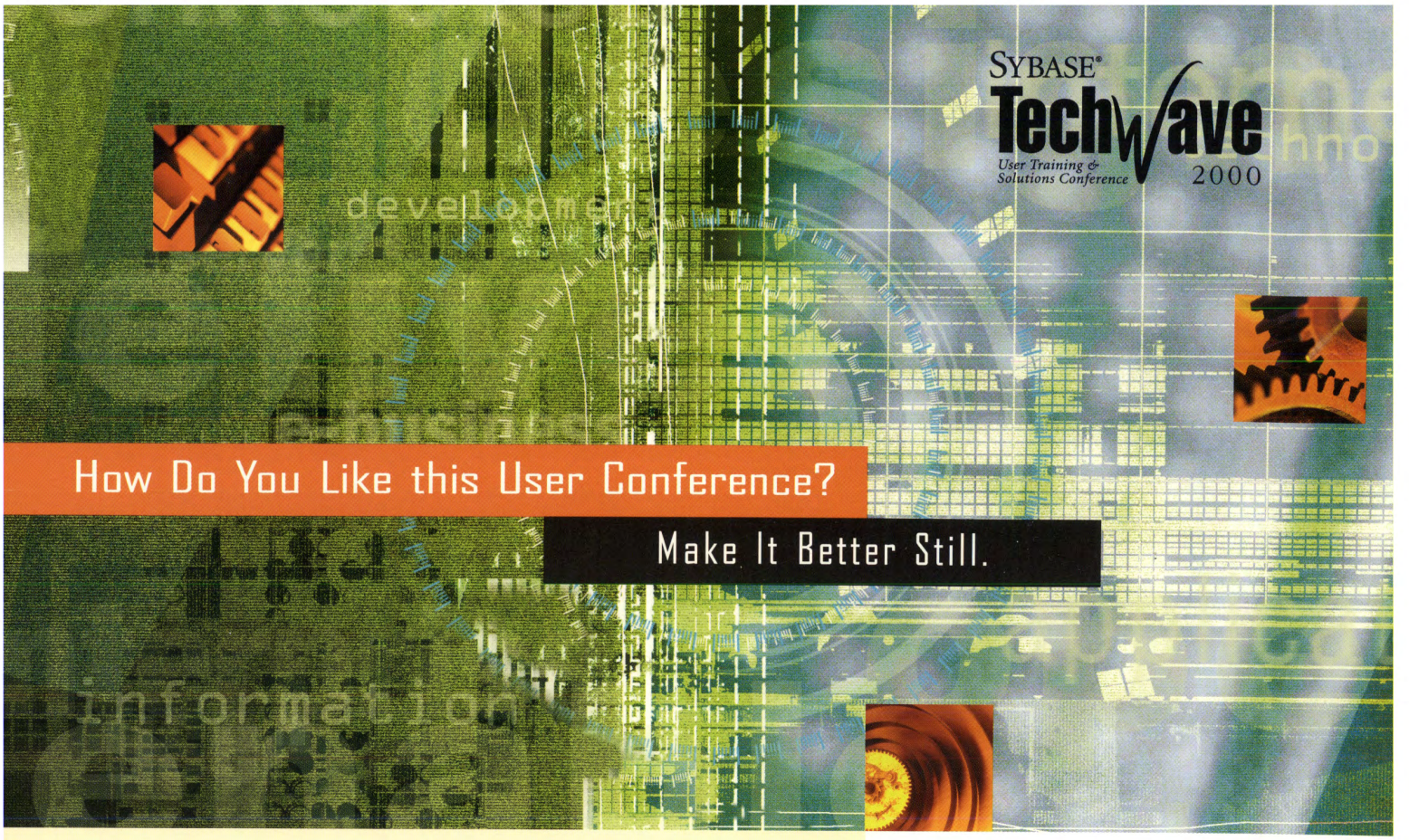
Portland, Oregon Sybase Developers User Group
Brad Ashton
bashton@centric-corp.com
www.teleport.com/~wagnerc/pbug

Pacific Northwest/Seattle (SNUG)
Chris Young
Cascade Software, Inc.
Phone: 206-224-3725
Fax: 206-224-6205
cyoung@cascadia-sw.com
www.cascadia-sw.com/snug

San Francisco PowerBuilder User Group
Amine Khechfe
Phone: 650-321-2640 x104
a@psi-systems.com
www.sfpbug.org/

Vancouver PowerBuilder User Group
Dennis Lee
Phone: 604-893-7040
dlee@vusual.com

For more information on PowerBuilder Local User Groups, please contact Micki Williams at 978-287-2303 (phone) or mwilliam@sybase.com.



Whether or not you're attending the Sybase TechWave 2000 User Training and Solutions Conference, you can benefit from joining the International Sybase User Group. DBAs, system administrators, and developers receive great discounts and plenty of free tools to help in your work every day!

More Information and Training for Less

- ❑ $75 discount on Registration at TechWave and Other ISUG-Affiliated Conferences
- ❑ Save 10% on Sybase Education Classes
- ❑ Save 20% on Sybase Certification Exams

Free Tools and User Benefits

- ❑ Free Copy of SQL Anywhere Studio
- ❑ EA Server with PowerJ evaluation CD
- ❑ ASE 12.0 Technical Documentation CD
- ❑ A subscription to the quarterly ISUG Technical Journal
- ❑ TechWave Conference proceedings CD

Better Access and Impact on Sybase Products

- ❑ Voting Power on Sybase Product Enhancements
- ❑ Access to the Online ISUG Membership Directory
- ❑ Technical Training at Local User Groups

INTERNATIONAL

Sybase User Group

For more information,
check out the ISUG website at

www.isug.com

or fill out the application form
in this journal!