

# zIIPing

BY WILLIAM FAVERO

**Z**IIPing and zAAPing away. While that's a cute opening for this article, it introduces a serious topic. IBM has shown a solid commitment to lowering the cost of running your business on System z. Your potential to increase your processing capacity without increasing your software cost can be significant with use of the System z9 Integrated Information Processor (zIIP), the System z Application Assist Processor (zAAP), and the Integrated Facility for Linux (IFL).

The main focus of this article is the zIIP specialty engine, but we'll briefly examine the zAAP and IFL specialty engines, too. (Remember, I'm a database guy, writing from a database guy's perspective. If you're a z/OS systems programmer, you should also visit IBM's System z Websites for hardware and operating system perspectives; go to [www.ibm.com/systems/z/hardware/](http://www.ibm.com/systems/z/hardware/) and [www.ibm.com/systems/z/os/](http://www.ibm.com/systems/z/os/)).

When something is stated well, there's not a lot of need to try to re-state it, so here's a quote from Jim Stallings, general manager, IBM System z, that I think sums up the importance of the zIIP announcement:

*"Data is at the core of today's most critical business issues, and the IBM System z9 mainframe equipped with a zIIP engine can expertly orchestrate information resources. When users centralize their data on the mainframe, they may decrease the risks associated with having multiple copies of data across diverse systems. Audit, compliance, control, and business recovery may be easier to manage when there's a single copy of the data.*

*The mainframe with a zIIP engine can increasingly play an essential role as a security-rich enterprise data hub."*

## zIIP: Why and How

Although an IBM press release from Jan. 24, 2006 provided a glimpse into the zIIP specialty engine, it wasn't really made public until late April. It started shipping in May, and, as of this writing, all but one of the DB2 and z/OS enabling Program Temporary Fixes (PTFs) were available in June. With zIIP, you can lower your total cost of operations and improve resource optimization while making the z9 your data server of choice. The zIIP will simplify any decision about placing or keeping your data on System z. Moving eligible distributed, selected utility, and Business Intelligence (BI) workloads to a zIIP will improve available capacity of existing general purpose engines, increasing throughput and possibly delaying your next processor upgrade.

Here are the prerequisites to take advantage of a zIIP engine:

- **Hardware:** You'll need a System z9 server or future follow-on model. The zIIP isn't available on any other processor in the z family, which includes the new z9 Business Class (BC) server or the z9 Enterprise Class (EC) server (the server formally known as z9 109). (To learn more about z9 systems, refer to IBM's Redbooks *IBM System z9 Business Class Technical Introduction*, SG24-7241 or *IBM System z9 Enterprise Class Technical Guide*, SG24-7124.)
- **Operating System:** The minimum requirement is z/OS V1.6 or above; no

prior OS will work. You'll also need Function Mode Identifier (FMID) and Authorized Program Analysis Report (APAR) support for z/OS V1.6 and V1.7. If you're not yet at z/OS V1.6, zIIP might be a great reason to start your OS migration plan. After all, z/OS V1.3 is already out of service, and V1.4 and V1.5 will go out of service in March 2007.

- **Database:** Only DB2 for z/OS Version 8 (with a few APARs) and future releases of DB2 support the workloads eligible for the zIIP. You can expect other products to look at ways to take advantage of zIIP in the future. DB2 for z/OS V8 can take advantage of the zIIP engine in Compatibility Mode (CM), Enable New Function Mode (ENFM), and New Function Mode (NFM). So, regardless of where your V8 upgrade is, you can take advantage of zIIP. You should be thinking about upgrading to V8 if that isn't already in your plans. You must be in DB2 for z/OS V8 NFM before you can upgrade to DB2 9, which has been announced and is in beta testing.
- **Budget:** A zIIP specialty engine is priced considerably less than a general-purpose engine for a z9. The base list price for a zIIP specialty engine on a z9 EC is \$125,000 (U.S.), which is consistent with zAAP pricing.

## zIIP Implementation

As mentioned earlier in this article, all zIIP maintenance, with the exception of APAR PK27578, closed with PTFs available in June 2006. Here are some of the APARs you should keep on eye on if you're planning a zIIP implementation:

# Along

- **PK18454:** This is the enablement APAR for DB2 for z/OS exploitation of the zIIP for DRDA threads. This APAR has been updated with detail that makes an excellent read.
- **PK19920 and PK20487:** DB2 for z/OS exploitation of the zIIP for the index maintenance function of the utilities (LOAD, REORG, REBUILD INDEX)
- **PK19921:** DB2 for z/OS exploitation of the zIIP for star join parallel threads
- **PK27578:** DB2 for z/OS exploitation of the zIIP for query parallelism
- **PK18215:** zIIP support for SDSF 1.5 and 1.7 users
- **PK25395:** OMEGAMON XE for DB2 PE (5655-P07) and OMEGAMON XE for DB2 PM (5655-P08) modification to support zIIP metrics
- **OA13499:** Provides RMF monitoring support for activity on zIIP specialty engine on the z9 EC (2094) and z9 BC (2096) servers. This APAR makes for another good read if you want more details on how zIIP metrics are gathered and how zIIP works.
- **OA16005:** Fixes for zIIP SMF service time reporting problems
- **OA15899:** Fixes for OMEGAMON II for MVS v550 (component of OMEGAMON XE for z/OS even though it's a stand-alone product)
- **OA15898:** Fixes for the OMEGAMON Base v550 (component of the above product)
- **OA15900:** Fixes for OMEGAMON XE on z/OS v310.

You can track DB2 APARs at [www.ibm.com/software/data/db2/zos/sup-](http://www.ibm.com/software/data/db2/zos/sup-)

port.html. More zIIP APARs might emerge as this article is published; the above list isn't meant to be all-inclusive.

For z/OS and z/OS.e V1.6 and V1.7, you'll need the following downloads to enable zIIP:

- FMID JBB77S9 supplies the necessary zIIP function support for z/OS 1.6.
- FMID JBB772S supplies the necessary zIIP function support for z/OS 1.7.

Ensure you pull the proper zIIP PSP bucket for the appropriate operating system release for any additional service that might become available.

## zIIP Eligible Workloads

So what makes this specialty engine so different? The zIIP really isn't any different from the other Processor Units (PUs) in a z9. It runs a full instruction set similar to all the other processors. It parks itself on a book alongside a standard Central Processor (CP) like all the other processors. In the case of zIIP, however, it's all in the name and which workloads the Workload Manager (WLM) sends its way.

zIIP, like all specialty engines, is intended to help you optimize your IT infrastructure, reduce your cost of operations and ownership, and increase your processing power. zIIP is designed so the enclave Service Request Block (SRB) is defined as the unit of work eligible to be dispatched onto the specialty engine. WLM has been enhanced so when DB2 configures an enclave, DB2 can tell WLM whether this SRB should be scheduled to the zIIP. Utilization is transparent to the application. The appli-

cation has no idea if part of its work is being run on the zIIP specialty engine or on a general-purpose processor. All this is accomplished while keeping overhead at a minimum. Regardless of where the workload originated, the portion of the work, for at least the Distributed Relational Database Architecture (DRDA) workload, that's eligible to run on the zIIP specialty engine vs. a general-purpose processor will typically be up to 40 percent. This enclave SRB interface isn't exclusive to IBM's use; Independent Software Vendors (ISVs) also can take advantage of it.

There are three types of workloads initially being enabled to the zIIP engine:

- DRDA work that comes into DB2 by use of TCP/IP
- Some very specific BI workloads that use star schemas
- Index workload from three DB2 utility processes.

When these workloads use the zIIP engine, they're running on an engine that doesn't affect the pricing of the software running on the z9. So, although you're adding processors and MIPS to assist with the work in your z9, you're not changing the model number or MSU rating of your z9. By not changing the model or increasing the number of MSUs you've installed, your IBM software costs (DB2, IMS, CICS, etc.) don't increase.

Moreover, because some specific work is being processed by the zIIP, capacity on the CPs is freed up to do other tasks. What's especially nice about how a specialty engine works is that it

requires no application changes, except activation of the engine itself, to take advantage of the specialty engine. DB2 for z/OS and the z/OS operating system, or more specifically WLM, are responsible for getting the right work to the zIIP engine. The zIIP also is always “on” once it’s installed. There’s no switch or parameter that allows DB2 to take advantage of zIIP.

### Distributed Workload (DRDA)

Let’s take a closer look at the tasks DB2 will want to identify as eligible for the zIIP, the tasks using an enclave SRB. This is the SQL portion of a distributed application that’s being processed by DB2 for z/OS. Coincidentally, network processing using DRDA across TCP/IP uses an enclave SRB (and has been using an enclave SRB for sometime), making it a perfect match. Moving a portion of this distributed work to a zIIP has great potential. Purchased Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM) applications (e.g., SAP, PeopleSoft, Siebel, Lawson, etc.) can use DB2 Connect, a product based on DRDA protocol, to access data on DB2 for z/OS via TCP/IP. Such work is eligible for processing on the zIIP specialty engine.

The list of applications can be extended to many others. An example might be WebSphere running under Linux in an IFL. When “talking” to the DB2 for z/OS in another LPAR, a HiperSocket is used. A HiperSocket is a TCP/IP connection, and this particular application uses DRDA protocol, so the work is still zIIP eligible. Another example might be WebSphere running on Windows or Unix. In fact, any application, even desktop applications running

on Microsoft that access DB2 through a network using DRDA protocol and TCP/IP, is eligible to use zIIP. All could potentially use the zIIP engine immediately because no changes are necessary to the applications.

### BI Workload

In the BI community, we often use a star schema, which is a single, large fact table with many smaller dimension tables. The process of joining these tables in an SQL statement is referred to as a star join. Star joins are part of star queries. Clear as mud, right? If not, take a look at the star schema diagram in Figure 1. All this is just a lead into our second eligible workload for zIIP: requests to DB V8 that use star schema parallel query. The SQL for a star join already uses an SRB for processing. DB2 V8 will be modified so the SRB will be an enclave, completing the eligibility of the BI function. Being able to run a portion of a star query, the child portion of the parallel query, on the zIIP engine will make it more cost-effective to run a BI workload in a mainframe environment. The longer the query runs, the greater the benefit of using the zIIP.

In addition to the parallel child tasks of a star join, the child portion of long-running parallel queries, including any single table access or multi-table join method that uses parallelism will be an eligible workload directed to the zIIP specialty engine based on an internal CPU threshold. When the threshold is exceeded, the parallel child tasks initiated by a local or remote SNA connection can be directed to a zIIP. However, for DRDA workload (remote TCP/IP connections), a portion of the main tasks can also be directed to a zIIP. For

either operation, long-running queries or star joins, parallelism does have to be enabled through the use of DSNZPARM CDESSRDEF, the BIND keyword DEGREE (ANY), or dynamically with a SET CURRENT DEGREE = ‘ANY’. Parallel query workload combined with the star schema zIIP eligible workload can only enhance the current list of reasons to keep your data warehouse data on System z.

We’re not done yet. There still could be additional CPU savings for this type of BI workload. Because most BI package queries already arrive via TCP/IP using DRDA, the main task may also be eligible to use zIIP. When those queries use parallel star joins, then the star join portion also could take advantage of the zIIP. This is an example of two different eligible workloads being simultaneously able to take advantage of zIIP engines for even greater cost savings. However, not all BI work is eligible. If the BI work is run locally and a network isn’t involved, it doesn’t qualify as DRDA network activity, although a parallel star query, if present, still may meet the criteria.

### Utility Index Maintenance Workload

The final eligible workload for the zIIP specialty engine we will discuss is DB2 utility work that performs index maintenance. More specifically, LOAD, REORG, and REBUILD INDEX can potentially have the zIIP engine perform the index maintenance portion of their processing. How much utility index work can be moved to the zIIP? That’s a classic DB2 answer: It depends! It can be as low as 10 percent or as high as 80 percent. Several variables come into play:

- How many indexes are defined on the table? If there are few indexes, the percent of eligible work will be more toward the 10 percent side; if there are a high number, the percentage will lean toward 80 percent.
- Having compression enabled also could lower the percentage of eligible work.

The IBM DB2 utilities run in TCB mode. Some outstanding modifications need to be applied to fully enable the utilities to use an enclave SRB.

There will always be instances where a workload isn’t appropriate for the zIIP engine. WebSphere, when running locally on z/OS, wouldn’t use the zIIP engine. However, the WebSphere workload still could use the zAAP engine for

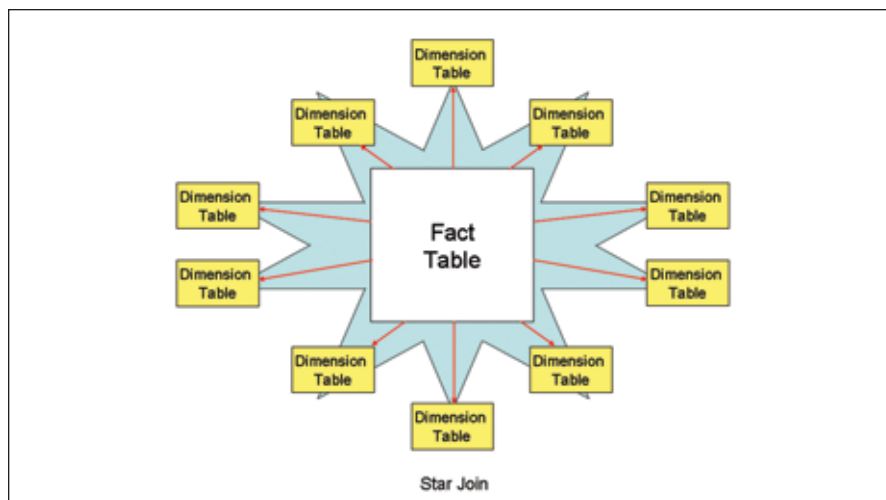


Figure 1: Star Schema Diagram

Java work from the JVM. DB2 batch workload running under TSO, CICS, and IMS still would run on traditional general-purpose processors (CPs) and wouldn't take advantage of zIIP specialty engine.

### How Many zIIPs Can You Have?

There's a limit to the number of zIIP engines you can order. The maximum number of zIIP engines must be less than or equal to the number of permanently purchased CPs. Stated another way, you can have one zIIP for every CP.

This maximum number is at the overall server level, not by LPAR. If the z9 is configured with more than one LPAR, one of the LPARs could be overloaded with zIIP engines. For example, you may have a situation where you have a BI-intensive workload running in one LPAR that might be able to take advantage of the zIIP engine and little eligible workload for the zIIP on the other LPARs. More of the zIIP engines could be enabled for the LPAR running the BI workload.

The zIIP and zAAP engines also can be configured to share a single CP. If you have eight CPs, you could have eight zIIP and eight zAAP engines. Your end configuration would be eight CPs, eight zIIP, and eight zAAP engines for a total of 24 processors, although only eight of those engines, the eight CPs, will have an effect on your monthly software license charges.

Consider another example: A fully loaded z9 (2094 model S54) can have up to 54 processors a customer can configure. That's four books with 16 processors per book with a total of 64. Of the 64 processors, there are eight System Assist Processors engines (two per book) and two processors marked as spares. The remaining 54 processors can be configured by the customer as CP, Internal Coupling Facilities (ICF), IFL, zAAP, and zIIP, in any combination, as long as the one-for-one ratio is maintained.

All PUs in a z9 server are physically identical. When the z9 is initialized, any PU can be characterized as a CP, IFL, ICF, zAAP, zIIP, or SAP. This concept gives the z9 its flexibility and high availability because any PU can be dynamically changed to avoid a server outage.

The only other "rule" to remember is that a zIIP can't be installed stand-alone in an LPAR. Any LPAR with a zIIP defined must have at least one CP enabled.

I tossed out quite a few acronyms in

# Stored Procedures

You've just read that stored procedures will not get to play with zIIP, at least not until DB2 9. And then, only native SQL stored procedures at that. Is this a negative thing? Does this leave you questioning your decision to implement stored procedures as part of your distributed application strategy? It shouldn't. All the benefits you have realized by using stored procedures do not change with the introduction of the zIIP specialty engine. zIIP is designed to lower your software stack cost. zIIP does nothing to improve a distributed application's performance, reduce its elapsed time, or number of instructions it executes. Stored procedures, on the other hand, will do all that and more. It's important to remember all the reasons stored procedures became so widely accepted in the first place:

- Stored procedures will reduce the CPU and elapsed time of a distributed task because multiple SQL statements are executed with a single stored procedure call rather than sending one SQL statement at a time across the network to DB2.
- Stored procedures communicate with DB2 via memory to memory operations locally rather than across a network with a remote application.
- Stored procedures are statically bound allowing a simple method for remote applications that are traditionally dynamic SQL to use static SQL.
- Application maintenance can be simpler and more straightforward because SQL can be modified in a single spot rather than having to be pushed out to numerous remote locations.

There are other reasons you might want to use stored procedures. This is just a brief recap of some of the more popular reasons for implementing stored procedures. And stored procedures are not completely void from using a zIIP specialty engine. It's estimated that as much as 10 percent of the processing of the SQL CALL statement, used to invoke a stored procedure, is done on a DDF enclave SRB and will be zIIP eligible. This includes processing input and output parameters, commit processing, and the processing of multiple result sets. If result set processing is a significant part of the CPU cost of a distributed application, enclave SRB cost will be higher, so there could be an even higher potential for substantial saving with a zIIP engine.

—W.F.

# Alphabet Soup

- *Central Processors (CPs)* are also referred to as general purpose processors. These are the processing units that do most of the work for the software you're running. In most cases, when you refer to where something is running, it's usually going to be on a CP.
- *System Assist Processor* is the first specialty engine. It manages the starting and ending of I/O operations for all LPARs and all attached I/O.
- *Internal Coupling Facilities (ICF)* was delivered in 1997 and provides additional processing capacity for the execution of the Coupling Facility Control Code (CFCC) in a Coupling Facility (CF) LPAR.
- *Integrated Facility for Linux (IFL)* provides additional capacity for Linux workloads running either natively or under z/VM. Linux is one of the next great pieces of software. It can be deployed anywhere, literally. Bring up Linux on Wintel and deploy a few applications. Maybe after a year or so, you might decide you no longer want that app on that platform. So you can get an IFL, install z/VM, bring up Linux and simply deploy the app in the System z. It's really that simple.
- *zAAP (System z9 Application Assist Processor)* made its first appearance in 2004. This specialty engine was originally called the zSeries Application Assist Processor but changed names when I wasn't paying attention—processes Java work from the Java Virtual Machine (JVM).

—W.F.

the last paragraph. For more information on what they mean, please see the accompanying sidebar.

## More on Qualifying Workloads

Any workload that arrives at DB2 for z/OS via DB2 Connect satisfies the DRDA portion of the requirement to be an eligible workload. Also, up to 40 percent of the DRDA workload coming in via TCP/IP is eligible for zIIP processing, making distributed activity as cost-effective as running local types of work. Because DRDA is an open and published architecture, some work that doesn't use DB2 Connect also is eligible. As long as the product uses DRDA protocol, works over TCP/IP, and the SQL arrives at DB2 as an enclave SRB, it's zIIP eligible.

Why enclaves? There are two types of work in z/OS: work that's dispatched either via a Task Control Block (TCB) or Supervisor Request Block (SRB). DB2 has made extensive use of SRBs since its beginnings. However, SRB work traditionally ran at the priority of the originating address space and couldn't be interrupted. This was an issue for distributed work. Distributed work (SQL) arrives via the Distributed Data Facility (DDF) address space under an SRB and would therefore run at the priority of DDF. Fortunately, z/OS came up with pre-emptable SRBs, specifically the enclave SRB. Because enclaves are interruptible, WLM can dispatch an enclave SRB at a different priority than the owning address space. This was important to DB2's distributed work's performance. Combined with the fact distributed work arrives via TCP/IP using DRDA protocol, this workload is easily identifiable to WLM, working in conjunction with the z/OS dispatcher, to be dispatched to a zIIP specialty engine.

Anything that runs via a TCB isn't eligible for the zIIP. Today, that includes all IMS, CICS, batch DB2, and stored procedure work. Even though the execution of a stored procedure may be requested by a distributed application, once they make it to DB2 for z/OS, they're dispatched to their own address space and run under a TCB. However, there are plans to make native SQL stored procedures, a feature being delivered in DB2 9, zIIP eligible. In addition, the single SQL CALL statement used to invoke a stored procedure, if it follows all of the rules for a distributed (DRDA) piece of work, may be zIIP eligible.

The lack of initial support for stored procedures in the zIIP shouldn't cause

you to question your decision to implement stored procedures. zIIP is designed to lower your software stack cost; it does nothing to improve a distributed application's performance, reduce its elapsed time, or number of instructions it executes. Stored procedures will do all that and more (see the accompanying sidebar titled "Stored Procedures").

Another workload type that doesn't qualify is WebSphere Application Server (WAS) running on the same z/OS as DB2 and accessing DB2 via a JDBC Type 2 driver. Although this is the recommended configuration to optimize performance, it isn't a zIIP-eligible configuration. If WAS accessed DB2 in a different LPAR using a JDBC Type 4 driver, therefore using DRDA and TCP/IP, it would then be an eligible workload.

## Am I a Candidate?

How do you know if your workload is a candidate for the zIIP? Although it's affordably priced, it's still a good piece of change to spend on something you don't even know if you can benefit from. Fear not! There's a process in place for you to determine the potential amount of DB2 zIIP eligibility you might achieve:

- Contact your local IBM office and inform them you want to determine your zIIP eligibility. After that, there are a couple different things that might occur.
- Start gathering up SMF 70 and 72 records for a slice of time that's representative of a good mixture of your workload. Analyzing this data will tell you if you have enough DRDA workload to do further analysis.
- Ensure you have well-defined WLM service classes and report classes for your DRDA workload. Adequate granularity also is essential to have as an accurate prediction of the amount of zIIP eligible work.

This process assumes all your DRDA work being tracked is TCP/IP and not SNA. This process can't differentiate between the two. If you have mixed workloads, the amount of predicted zIIP-eligible work could be inflated. The SMF records also can be extracted on a DB2 V7 or V8 subsystem, though SMF data from a V8 subsystem will yield a more accurate prediction.

Even with a significant amount of DRDA workload, you will still need to ensure that all of that workload will

actually qualify as zIIP eligible. If you make extensive use of stored procedures or User Defined Functions (UDFs), further analysis is necessary. Although stored procedures and UDFs often arrive as distributed work, they run in their own address space under a TCB, which isn't zIIP eligible. In that case, gathering DB2 accounting data via the SMF Type 101 records is necessary. You should use IBM's zIIP Estimator Data Collector, which can be loaded on a customer's machine to do some pre-analysis against SMF Type 101 records to minimize the volume. IBM will assist you in the SMF analysis and work you through the process.

### zIIP Metrics

We haven't discussed metrics, as there aren't yet a lot of measurements available for zIIP. Once you have your zIIP, you should be able to use the DB2 statistics and accounting trace records to determine how much workload was eligible and took advantage of the zIIP.

However, there have been no IFCIDs (Instrumentation Facility Component Identifiers) added to DB2 in support of

zIIP. All CPU time that's processed by a zIIP specialty engine won't be accumulated into the CPU fields of the current DB2 statistics and accounting records. This should result in an immediate reduction in the CPU consumption reported by existing chargeback tools that use the IFCIDs associated with the statistic and accounting traces.

The existing IFCIDs that currently participate in the statistics and accounting traces have had fields added to accommodate for zIIP CPU time and zIIP-eligible CPU time. In addition, RMF will provide usage information both before and after your zIIP purchase. How much time is spent in zIIP and how much time eligible zIIP workload was spent running in a CP will be added to the SMF Type 30 records, along with zIIP usage information planned for SMF Type 72 records.

### Conclusion

You want to put your data on the mainframe, where you have the highest availability, unmatched integrity, and uncompromising security. To save on software costs, let WLM move some of

your work over to a zIIP specialty engine, which will provide a possible reduction in MSUs and the price of your software stack. You can further reduce your software cost by consolidating your servers under Linux on z with the help of an IFL and z/VM. This can reduce your server administration and environmental cost. And just moving to z9 will get you more speed for your processing dollars. These are just some reasons the mainframe should be your enterprise data hub. And they said the mainframe was dead!

**Acknowledgement:** I'd like to thank IBM's DB2 for z/OS Product Marketing Manager, Terrie Jacopi, for her assistance. **Z**

### About the Author

In the past 29 years, Willie Favero has been a customer, worked for IBM, worked for a vendor, and is now an IBM employee again. He has always worked with databases and has more than 20 years of DB2 experience. A well-known, frequent speaker at conferences and author of numerous articles on DB2, he's currently with North American Lab Services for DB2 for z/OS, part of IBM's Software Group. Visit his DB2 blog at <http://blogs.ittoolbox.com/database/db2zos>. e-Mail: [wfavero@attlgobal.net](mailto:wfavero@attlgobal.net)

*"The articles in  
DB2 Update are  
**VERY  
INFORMATIVE  
AND  
RELEVANT.***

*The code examples  
allow their  
functionality to be  
quickly incorporated  
into any site code,  
thus saving lots of  
time and effort —  
this is a real bonus  
for any  
DBA/systems  
programmer."*

**Pav Kumar-Chatterjee**

**DB2**

**131**

**update**

**In this issue**

- 3 No more covering letters
- 7 Using DB2 710 Real Time Statistics (RTS) and DSNACCOR
- 25 Transfer data utility
- 38 The DB2 UDB db2relocatedb command
- 48 DB2 news

**For more information, visit [www.xephonusa.com](http://www.xephonusa.com)**