122

AIX update

December 2005

In this issue

© Xephon Inc 2005

A itc PUBLICATION

Published by

Xephon Inc
PO Box 550547
Dallas, Texas 75355
USA

Phone: 214-340-5690
Fax: 214-341-7081

Editor

Trevor Eddolls
E-mail: trevore@xephon.com

Publisher

Colin Smith
E-mail: info@xephon.com

Subscriptions and back-issues

A year's subscription to *AIX Update*, comprising twelve monthly issues, costs $275.00 in the USA and Canada; £180.00 in the UK; £186.00 in Europe; £192.00 in Australasia and Japan; and £190.50 elsewhere. In all cases the price includes postage. Individual issues, starting with the November 2001 issue, are available separately to subscribers for $24.00 (£16.00) each including postage.

AIX Update on-line

Code from *AIX Update*, and complete issues in Acrobat PDF format, can be downloaded from our Web site at http://www.xephon.com/aix; you will need to supply a word from the printed issue.

Disclaimer

Readers are cautioned that, although the information in this journal is presented in good faith, neither Xephon nor the organizations or individuals that supplied information in this journal give any warranty or make any representations as to the accuracy of the material it contains. Neither Xephon nor the contributing organizations or individuals accept any liability of any kind howsoever arising out of the use of such material. Readers should satisfy themselves as to the correctness and relevance to their circumstances of all advice, information, code, JCL, scripts, and other contents of this journal before making any use of it.

Contributions

When Xephon is given copyright, articles published in *AIX Update* are paid for at the rate of $160 (£100 outside North America) per 1000 words and $80 (£50) per 100 lines of code for the first 200 lines of original material. The remaining code is paid for at the rate of $32 (£20) per 100 lines. To find out more about contributing an article, without any obligation, please download a copy of our *Notes for Contributors* from www.xephon.com/nfc.

© Xephon Inc 2005. All rights reserved. None of the text in this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior permission of the copyright owner. Subscribers are free to copy any code reproduced in this publication for use in their own installations, but may not sell such code or incorporate it in any commercial product. No part of this publication may be used for any form of advertising, sales promotion, or publicity without the written permission of the publisher.

Printed in England.

AIX error notification

Although AIX logs an error whenever any abnormal software or hardware event occurs (such as a physical volume becoming inaccessible, a segmentation fault in a program, or filesystem full, etc), there are circumstances in which an error can go undetected – especially if the problem is not serious enough to interrupt. In such cases, automatic notification can alert the administrator to the problem.

The following procedure describes how to set up automatic notification. The technique this procedure adopts is to use the errnotify ODM method, which gets called when a new error entry is created. This technique or procedure is not something new. It is mentioned in plenty of IBM documentation. A simple search on Google will return many hits showing various examples of how to do it. This article is no different: it is just another procedure that is compiled and simplified, and can be implemented as is, with a little modification, of course, to suit your needs. You can both track and be notified about all errors, or track those significant and serious enough for your attention. For example you may want to be notified about only hardware disk errors, or only filesystem full events, etc. By modifying the following procedure, you can track other errors that are significant to you.

The procedure will be explained through various examples. This procedure will show two examples. Example 1 will show you how to create two errornotify (errnotify) events for hardware and software errors and how to report each one individually. Example 2 will show you how to track all errors and report them through a script.

We need to create an errnotify ODM method that includes the following options:

- en_pid – Process ID (PID) for use in identifying the error notification object.

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

- en_name – uniquely identifies the object.
- en_persistenceflg – 0 non-persistent (removed at boot time); 1 persistent (persists through boot).
- en_label – label associated with a particular error identifier (errpt –t).
- en_crcid – error identifier associated with a particular error.
- en_class – class of the error log entries to match (H, S, O, or U).
- en_type – severity of error log entries to match (TEMP, PERM, INFO, UNKN, PEND, etc).
- en_alertflg – identifies whether the error is alertable (TRUE, FALSE).
- en_resource – name of the failing resource. For hardware error class, it is the device name.
- en_rtype – type of failing resource. For hardware error class, it is the device type.
- en_rclass – class of the failing resource. For hardware error class, it is the device class.
- en_method – a user-programmable action, such as a shell script or command string, to be run when an error matching the selection criteria of this error notification object is logged.

In Example 1, we will create two errornotify ODM methods for software and hardware. We can add these methods to a file and update the ODM using this file:

```
# cat errnotify_odm
errnotify:
  en_pid = 0
  en_name = "syslog"
  en_persistenceflg = 1
  en_label = ""
  en_crcid = 0
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
  en_class = "H"
  en_type = ""
  en_alertflg = ""
  en_resource = ""
  en_rtype = ""
  en_rclass = ""
  en_method = "/usr/bin/errpt -l $1 -a | mail -s 'NEW hardware error'
HW_helpdesk@yourdomain.com"
errnotify:
  en_pid = Ø
  en_name = "syslog"
  en_persistenceflg = 1
  en_label = ""
  en_crcid = Ø
  en_class = "S"
  en_type = ""
  en_alertflg = ""
  en_resource = ""
  en_rtype = ""
  en_rclass = ""
  en_method = "/usr/bin/errpt -l $1 -a | mail -s 'NEW software error'
SW_helpdesk@yourdomain.com "
```

Note in the example that the detailed error was sent via e-mail to the correct correspondent in the *en_method* stanza.

In Example 2, we will create one errornotify ODM method to track all errors. We can add these methods to a file and update the ODM using this file:

```
# cat errnotify_odm
errnotify:
  en_pid = Ø
  en_name = "syslog"
  en_persistenceflg = 1
  en_label = ""
  en_crcid = Ø
  en_class = ""
  en_type = ""
  en_alertflg = ""
  en_resource = ""
  en_rtype = ""
  en_rclass = ""
  en_method = "/location_path/notifyme $1"
```

Note in this example that *en_method* is set to run a program or script called */location_path/notifyme*. In this program you can specify what you want to do with this error, like e-mail,

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

page, ignore, etc.

Check whether you already have any errornotify objects in the ODM and delete them if they exist:

```
# odmget -q"en_name=syslog" errnotify
# odmdelete -o errnotify -q"en_name=syslog"
```

Update the ODM by adding the newly-created errornotify objects using the file errnotify_odm that we created earlier:

```
# odmadd errnotify_odm
# odmget -q"en_name=syslog" errnotify
```

The error notification is now ready. However, in the case of your *en_method* for the errornotify object being set to use a custom program or script to handle the new error entries, like the program 'notifyme' in Example 2, this program needs to be created. An example of such program would be the following:

```
# cat /location_path/notifyme

##############################################################################
#!/bin/ksh
# This script will evaluate the new error log entries and will
#  act depending on the error.

email_HW='HW_helpdesk@yourdomain.com'
email_SW='SW_helpdesk@yourdomain.com'
email_admin='sysadmin@yourdomain.com'
emergency_pager='pager@yourdomain.com'

subject="New Error Log Entry $(date)"

err_class="$( errpt -l $1 -a | grep "^Class:" | awk '{print $2}')"

# If it is Hardware send to Hardware support
if [ "${err_class}" = "H" ]
then
errpt -l $1 -a | mail -s "${subject}" $email_HW

# If it is a permanent disk error page emergency
if [[ "$( errpt -l $1 -a | grep "^Type:" | awk '{print $2}')" = "PERM"
and "$( errpt -l $1 -a | grep "^Resource Class:" | awk '{print $2}')" =
"disk" ]]
then
label='errpt -l $1 -a | grep '^LABEL:' | awk '{print $2}''
errpt -l $1 -a | grep -v '\—.*' | sed \
               -e "s/  */ /g" \
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
                    -e "s/   //g"  \
                -e "/^$/d"     \
                   | mail -s "Disk Error ${label}" ${emergency_pager}
      fi
fi

# If it is Software send to Software support
if [[ "${err_class}" = "S" ]]
then
errpt -l $1 -a | mail -s "${subject}" $email_SW
fi

# If it is Other or undetermined errors send to System Admin
if [[ "${err_class}" = "O" or "${err_class}" = "U"]
then
errpt -l $1 -a | mail -s "${subject}" $email_admin
fi

exit
#########################################################################
```

The above script is just a sample. You can modify it in a way that suits your needs, or you can create your own script.

Since we are on the subject of error notification, in a forthcoming edition of *AIX Update* there will be an article about forwarding error logs from multiple AIX servers to a log on a central server. This is another variation of the same error notification subject of this article.

Basim Chafik
Senior Systems Analyst
IBM Certified Advanced Technical Expert (CATE)
Plexus (Division of BancTec) (Canada)
© Xephon 2005

KSH scripting: tally all values inside a loop

To add all values of an incrementing or decrementing variable inside a loop, the following Korn shell script can be used:

```
x=0; for i in 1 2 3 4 5 6 7 8 9 10; do x=$(($x+$i)); done; print $x
```

Robert Kaiser
System Analyst
Bayerischer Rundfunk (Germany)
© Xephon 2005

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

AIX and DB2 tuning for DB2 performance – a case study: part 2

This month we conclude the article looking at a case study for tuning DB2 under AIX.

I/O PERFORMANCE

Now we discuss various reports used as input for tuning the I/O performance problem.

NODE0

tty:	tin	tout	Avg-cpu:	user	sys	idle	iowait
	1.1	1038.6		2.3	7.2	24.6	65.9
Disks:	tm_act	Kbps	tps	Kb_read	Kb_wrtn		
hdisk1	79.7	826.4	206.3	8208	56		
hdisk2	0	0	0	0	0		
hdisk0	85.8	869.6	217.1	8640	56		
hdisk4	0	0	0	0	0		
hdisk3	0	0.4	0.1	0	4		
cd0	0	0	0	0	0		
tty:	tin	tout	avg-cpu:	user	sys	idle	iowait
	0.3	827.1		1.3	3.2	0.2	95.3
Disks:	tm_act	Kbps	tps	Kb_read	Kb_wrtn		
hdisk1	100	1000.4	250.1	10000	4		
hdisk2	0	0	0	0	0		
hdisk0	100	1033.2	258.3	10328	4		
hdisk4	0	0	0	0	0		
hdisk3	0.1	2	0.4	0	20		
cd0	0	0	0	0	0		

Figure 1: Iostat output for node 0

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

NODE1

tty:	tin	tout	avg-cpu:	user	sys	idle	iowait
	0	0		18.1	15.9	6	60
Disks:	tm_act	Kbps	tps	Kb_read	Kb_wrtn		
hdisk2	55.3	472.4	111.1	1476	3248		
hdisk0	93.9	864	199.7	2292	6348		
hdisk1	93.4	831.2	190.8	2092	6220		
hdisk4	41	1538.4	151.1	12092	3292		
hdisk3	0	0	0	0	0		
cd0	0	0	0	0	0		
tty:	tin	tout	avg-cpu:	user	sys	idle	iowait
	0	0		35.9	23.9	2.1	38.1
Disks:	tm_act	Kbps	tps	Kb_read	Kb_wrtn		
hdisk2	45.9	387.6	96.5	3140	736		
hdisk0	92.5	704.8	174.4	5104	1944		
hdisk1	89.5	665.6	164.6	4760	1896		
hdisk4	51.1	2550.8	222.1	23436	2072		
hdisk3	0	0	0	0	0		
cd0	0	0	0	0	0		

Figure 2: Iostat output for node 1

Data collection

Figures 1 and 2 show the output from the **iostat** command for this system:

```
# iostat 1 10
```

Only two samples have been included for each node displaying the problem.

Data analysis

The **iostat** output investigation shows:

- A very high % *tm_act* value for hdisk0 and hdisk1, and very low % *tm_act* value for hdisk2, hdisk3, and hdisk4.

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

The % *tm_act* is the percentage of time the disk is busy. In order to deliver good performance, a system should be recording an average disk busy of less than 40%.

- A high % *iowait* value.

  The % *iowait* is the percentage of time the CPU is idle while waiting on local I/O.

Generally % *iowait* values over 25% indicate that the disk subsystem may not be balanced properly, or it may be the result of a disk-intensive workload.

Inference:

- Paging space can be another reason for a high % *tm* observation. In order to deliver good performance, a system should be recording an average disk busy of less than 40%.
- Generally % *iowait* values over 25% indicate that the disk subsystem may not be balanced properly, or it may be the result of a disk-intensive workload.

Recommendations:

- Move some data from busy drives (hdisk0 and hdisk1) to idle drives (hdisk2, hdisk3, and hdisk4 have low disk activity), which will give improved performance.
- Check paging activity. Only hdisk0 is used as the paging space for NODE0. High paging activity is another reason for high disk activity. Spread the paging over multiple drives if possible, thus sharing the paging space load across multiple drives. AIX uses virtual memory to address more memory than is physically available in the system. The Virtual Memory Manager (VMM) handles the management of memory pages in RAM or on disk. Virtual memory segments are partitioned in units called pages. A paging space is a type of logical volume with allocated disk space storing information that is resident in virtual memory but is not currently being accessed. This logical

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

volume has an attribute type equal to paging, and is usually simply referred to as paging space or swap space. When the amount of free RAM in the system is low, programs or data that have not been used recently are moved from memory to paging space to release memory for other activities.

Page Space	Physical Volume	Volume Group	Size	%Used	Active	Auto	Type
hd6	Hdisk0	rootvg	100MB	98	yes	yes	lv

Figure 3: Page space information

The page space details are output from the command:

```
# lsps -a
```

Example output is shown in Figure 3.

- Increase *NUM_IOSERVERS* because *Time waited for prefetch (ms)* is in seconds. Set the optimal prefetch size,

ENVIRONMENT DETAILS	
Configuration Parameter	
NUM_IOSERVERS	5
NUM_IOCLEANERS	4
Registry Variables	
DB2_PARALLEL_IO	Not Set
Database Snapshot	
Time waited for prefetch (ms)	3241525
Bufferpool Snapshot	
Asynchronous pool data page writes	57976
Asynchronous pool index page writes	2161
Buffer pool data writes	105445
Buffer pool index writes	2197

Figure 4: Environment details

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

PREFETCHSIZE = (Extent size * N) – see Figure 4.

- Increase *NUM_IOCLEANERS* because the Asynchronous Write Percentage (AWP) is less than 90%.

  AWP = ((Asynchronous pool data page writes + Asynchronous pool index page writes) * 100) / ("Buffer pool data writes" + "Buffer pool index writes")

  (57976 + 2161)/(105445+2197)*100

  60137*100/107642 = 55.86%

- Perform a **runstats** to ensure that the optimizer is aware of indexes.
- Increase the bufferpool to ensure that data is retrieved more often from memory than disk
- Increase the SORTHEAP parameters to ensure that sort uses memory rather than disk.
- Do a reorg of tables that are heavily used. This will defragment the data allowing data to be retrieved with fewer disk scans.
- Define only one container when using a RAID device. The reason being, when we define multiple containers, the data is striped, not only across the DB2 containers, but effectively also at the hardware level by the RAID device. This gives us striping on top of striping, which is not desirable.
- When using striping, the following are some general performance considerations:
  - DB2_PARALLEL_IO=*
  - DB2_STRIPED_CONTAINERS=ON
  - Tablespace Extent size = RAID stripe size
  - PREFETCHSIZE = (Extent size * N) for a N+P disk array.

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

Notes

The **iostat** command reports CPU statistics and input/output statistics for tty devices, disks, and CD-ROMs. It is used for monitoring system input/output device loading by observing the time for which the physical disks are active in relation to their average transfer rates. The **iostat** command generates reports that can be used to change system configuration to better balance the input/output load between physical disks.

The **iostat** command generates two types of report – the tty and CPU utilization report, and the disk utilization report.

The first report generated by the **iostat** command is the tty and CPU utilization report. For multiprocessor systems, the CPU values are global averages among all processors. Also, the I/O wait state is defined system-wide and not per processor. The report has the format shown below:

- *Tin* – shows the total number of characters read by the system for all ttys.
- *Tout* – shows the total number of characters written by the system to all ttys.
- *% user* – shows the percentage of CPU utilization that occurred while executing at the user level (application).
- *% sys* – shows the percentage of CPU utilization that occurred while executing at the system level (kernel).
- *% idle* – shows the percentage of time that the CPU or CPUs were idle and the system did not have an outstanding disk I/O request.
- *% iowait* – shows the percentage of time that the CPU or CPUs were idle during which the system had an outstanding disk I/O request. This value may be slightly inflated if several processors are idling at the same time (an unusual occurrence).

The second report generated by the **iostat** command is the disk utilization report. The disk report provides statistics on a

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

per physical disk basis. The report has a format similar to the following:

- % *tm_act* – indicates the percentage of time the physical disk was active (bandwidth utilization for the drive).
- *Kbps* – indicates the amount of data transferred (read or written) to the drive in KB per second.
- *tps* – indicates the number of transfers per second that were issued to the physical disk. A transfer is an I/O request to the physical disk. Multiple logical requests can be combined into a single I/O request to the disk. A transfer is of indeterminate size.
- Kb_read – the total number of KB read.
- Kb_wrtn – the total number of KB written.

MEMORY PERFORMANCE

We could see very high memory utilization and high paging activity. The paging performance was investigated and the symptoms of high memory utilization could be explained.

Data collection

The system outputs are gathered by the **vmstat** and **vmtune** commands. Figures 5 shows the output from the system using the **vmstat** command:

```
# vmstat 2 40
```

The output of the **vmtune** command:

```
# vmtune
```

looks like Figure 6.

Inference:

- We can see that in the **vmstat** output *wa* increases the moment there is some paging (*pi* and *po*). Until then it is low. (*pi* are pages paged in from paging space, *po* are

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

Node0																					
r	b	Avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	Cpu	sr/fr			
																	Util				
1	2	327770	864	0	0	0	0	0	0	578	1369	916	1	1	98	0	2				
0	2	327770	860	0	0	0	0	0	0	570	1235	872	0	1	98	1	1				
0	2	327770	867	0	0	0	0	0	0	391	672	402	0	0	99	0	0				
1	2	327770	865	0	0	0	0	0	0	395	683	414	0	0	99	0	0				
0	2	327770	864	0	0	0	0	0	0	390	656	401	0	0	99	0	0				
0	2	327983	617	0	0	0	0	0	0	398	1048	418	0	1	98	1	1				
0	2	327983	615	0	0	0	0	0	0	392	740	406	0	0	99	0	0				
0	2	327983	613	0	0	0	0	0	0	394	746	413	0	0	99	0	0				
0	2	327775	844	0	0	0	0	0	0	484	1043	655	0	1	99	0	1				
1	**11**	317430	3061	0	**507**	**0**	58	70	0	1388	1469	2464	1	4	16	**79**	5	1.207			
0	**2**	317554	2720	0	**6**	**0**	0	0	0	1254	5456	3117	3	4	92	**1**	7				
1	**5**	321886	121	0	**102**	**293**	702	1131	0	2321	8838	4938	24	11	15	**49**	35	1.611			
2	**6**	334243	159	0	**276**	**145**	1715	2441	0	2211	9398	3972	33	23	3	**40**	56	1.423			
5	**7**	348549	123	0	**275**	**94**	2214	2617	0	2197	15290	3655	47	39	1	**13**	**86**	1.182			
5	**3**	353523	126	0	**130**	**6**	992	1071	0	2448	12624	3509	57	36	1	**6**	**93**	1.08			
6	**2**	350458	901	0	**41**	**2**	109	131	0	2377	12328	3088	52	44	2	**2**	**96**	1.202			
3	**2**	349312	827	0	**48**	**0**	55	90	0	1823	9442	3255	40	34	19	**8**	74	1.636			
2	2	349799	690	0	6	6	259	311	0	1608	6777	2949	19	24	48	8	43	1.201			
2	2	350490	688	0	1	0	89	93	0	1408	5276	2398	19	14	66	1	33	1.045			
3	2	350214	881	0	11	2	126	145	0	1376	6510	2661	33	24	38	6	57	1.151			

Figure 5: ***vmstat*** *output (continues)*

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

r		b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	Cpu	sr/fr
																		Util	
5		2	305441	25259	0	8	0	0	0	0	1705	8726	3233	55	19	23	2	74	
4		2	308158	15566	0	6	0	0	0	0	1743	9332	3228	52	21	24	3	73	
7		2	308445	314	0	3	42	1104	6396	0	1917	7722	2990	64	23	8	5	87	5.793
4		2	309043	127	0	4	0	69	1411	0	1877	9022	3047	56	22	21	2	78	20.45
6		2	316577	2303	0	3	80	700	5340	0	1892	8817	2959	50	27	22	1	77	7.629
4		2	318292	156	0	23	6	39	82	0	2012	9087	3104	58	22	17	3	80	2.103
3		1	314109	5288	0	5	16	84	205	0	1773	8224	3258	47	20	31	2	67	2.44
2		2	304230	11897	0	8	0	0	0	0	2095	8397	3845	54	12	29	5	66	
1		2	300810	15759	0	1	0	0	0	0	1597	5389	3031	30	6	61	3	36	
7		**2**	310222	120	0	**13**	**74**	661	1884	0	1812	6949	3062	82	16	1	**1**	**98**	2.85
6		**3**	311029	1093	0	**44**	**216**	3018	15139	0	2267	6671	3668	66	27	0	**6**	**93**	5.016
3		**5**	315582	114	0	**94**	**269**	2297	3564	0	2643	6316	4423	31	17	5	**47**	48	1.552
3		**3**	312711	4255	0	**143**	**50**	932	1599	0	2410	8786	3975	46	19	15	**20**	65	1.716
1		**2**	305949	9582	0	**16**	**0**	0	0	0	1675	7234	3467	22	22	51	**5**	44	
4		**3**	307254	1095	0	**99**	**13**	172	307	0	2324	9287	3991	41	26	11	**21**	67	1.785
2		**2**	309506	125	0	**74**	**19**	562	1039	0	2413	7600	4464	36	12	21	**31**	48	1.849
3		**2**	308894	123	0	**11**	**14**	178	326	0	2064	7084	3621	49	10	31	**10**	59	1.831
2		**2**	306841	1650	0	**23**	**18**	154	306	0	1836	8543	3297	57	18	17	**8**	75	1.987
3		**2**	306052	710	0	**32**	**15**	251	465	0	2220	9111	3668	53	18	19	**11**	71	1.853
3		**2**	305093	2198	0	**17**	**15**	328	468	0	2265	9343	3594	61	19	14	**6**	80	1.427

Figure 5: ***vmstat*** *output (continues)*

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

NODE1																				
r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	CPU Util	sr/fr
0	2	363997	221	0	0	0	6	11	0	403	948	429	0	1	99	0	1	1.83
0	2	363997	208	0	0	0	0	0	0	395	937	407	0	1	98	0	1	
1	2	363997	206	0	0	0	0	0	0	404	928	427	0	1	98	0	1	
0	2	364210	159	0	0	0	11	16	0	398	1379	413	0	2	98	0	2	1.455
1	**8**	362684	1803	0	**177**	**1**	282	436	0	920	2634	1563	2	5	69	**25**	7	1.546
1	**15**	348056	4347	0	**490**	**0**	0	0	0	1277	1726	2225	1	3	4	**91**	4	
1	**3**	339725	3267	0	**138**	**0**	0	0	0	2110	11660	5349	16	10	52	**22**	26	
2	**2**	341689	120	0	**93**	**0**	229	394	0	2189	13346	4917	47	12	21	**21**	59	1.721
3	**3**	340397	659	0	**119**	**4**	655	1133	0	2206	8852	3992	47	16	13	**24**	63	1.73
4	**3**	338664	282	0	**134**	**8**	665	1487	0	1972	11762	3470	49	36	2	**12**	**85**	2.236
5	**2**	338649	416	0	**24**	**1**	200	462	0	2769	12744	3932	48	30	18	4	78	2.31
4	2	336051	3099	0	16	2	101	245	0	2394	12918	3514	45	37	15	2	82	2.426
2	2	335606	2517	0	5	0	0	0	0	1793	8354	3333	41	18	39	2	59	
2	2	336111	507	0	6	3	167	325	0	1647	6111	2990	26	15	50	10	41	1.946
1	2	337869	298	0	3	9	169	311	0	1519	5939	2578	35	14	48	3	49	1.84
2	2	336619	122	0	3	0	0	0	0	1575	6665	3031	32	18	44	6	50	
2	2	336082	1427	0	1	2	105	705	0	1457	6358	2802	15	15	68	2	30	6.714
2	2	337273	126	0	8	4	111	4599	0	1570	7356	2969	34	20	41	4	54	41.43
2	2	336971	1620	0	3	32	202	2930	0	1832	8303	3319	27	20	51	2	47	14.51
1	2	337104	1021	0	4	0	0	0	0	1443	6151	2967	18	15	64	2	33	

Figure 5: ***vmstat*** *output (continues)*

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

r	b	avm	fre	re	pi	po	fr	sr	cy	in	sy	cs	us	sy	id	wa	CPU	sr/fr
																	Util	
9	2	378917	2162	0	18	172	490	732	0	2574	10034	3429	73	25	1	1	98	1.494
7	3	379264	134	0	87	16	361	3620	0	2069	8513	3233	75	25	0	1	100	10.03
6	2	369267	10374	0	109	4	261	1647	0	2376	11519	3839	76	24	0	0	100	6.31
5	2	361812	16311	0	19	0	0	0	0	2113	8482	3006	86	14	0	0	100	
5	2	365919	5277	0	9	0	0	0	0	1698	11444	2894	85	10	3	2	95	
8	2	372321	121	0	17	41	435	4178	0	1819	7898	3250	86	14	0	0	100	9.605
3	3	372040	116	0	23	46	985	2800	0	2070	6780	3629	48	11	22	19	59	2.843
2	7	377072	126	0	40	263	1376	3152	0	2917	9853	4893	39	15	2	45	54	2.291
3	4	370520	7582	0	68	63	274	540	0	2499	9627	3803	53	16	11	19	69	1.971
4	3	365697	6188	0	53	0	0	0	0	1963	9370	3820	54	27	4	15	81	
4	2	365723	1116	0	77	0	0	0	0	2142	10672	3519	65	23	4	8	88	
3	3	367585	219	0	15	94	324	1030	0	2186	7350	3583	44	12	25	20	56	3.179
4	2	366760	62	0	63	38	210	636	0	2293	9361	3785	60	14	15	12	74	3.029
6	2	365197	3075	0	59	31	295	4837	0	2068	9934	3567	71	20	3	6	91	16.4
3	2	366024	968	0	11	0	35	282	0	2260	9364	3612	80	12	6	1	92	8.057
4	2	368869	675	0	22	29	470	3797	0	2177	8811	3584	74	19	5	3	93	8.079
3	2	367265	1837	0	5	1	11	25	0	1742	7950	3179	66	15	18	1	81	2.273
3	2	363423	2982	0	13	0	0	0	0	2062	9280	3490	63	19	12	6	82	
3	3	364456	1270	0	17	35	354	1339	0	2124	8841	3782	59	18	14	10	77	3.782
3	2	365300	242	0	5	15	178	631	0	2110	8162	3243	43	13	41	3	56	3.545

Figure 5: ***vmstat*** *output (continued)*

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

minperm	maxperm	minpgahead	maxpgahead	minfree	maxfree	pd_npages
23343	146208	2	8	115	154	524288

Figure 6: ***vmtune*** *output*

pages paged out to paging space.)

- Notice the high I/O wait in the output and also the number of threads on the blocked queue. Other I/O activity might cause an I/O wait, but in this particular case, the I/O wait is most likely caused by the paging in and out from paging space.
- The *pi* column varied from 0 to the highest level of 490 pages paged in from paging space for NODE1 and highest level of 507 for NODE0. Although a *pi* level of no more than 5 is considered acceptable, a level higher than 5 is not necessarily an indication of a performance problem, due to the fact that for every page paged in, there must have been a page that was paged out.

Recommendations:

- We notice very good buffer pool hit ratios, but lots of paging. Initially we also noticed large sort spills. Thus we can conclude that insufficient sort space is causing the paging. Increase the SORTHEAP parameter.

  The output from the bufferpool snapshot:

  ```
  # get snapshot for all bufferpools
  ```

  The bufferpool hit ratio calculated using formula:

  (1 - ((data physical reads + index physical reads)/(data logical reads + index logical reads)) * 100.

  The output is shown in Figure 7.

- Lower bufferpool memory and check hit ratios. If hit ratios are still high, reducing the bufferpool will have the biggest

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

	Bufferpool	Buffer pool	Buffer pool	Buffer pool	Buffer pool	Buffer Hit
	name	data logical	data physical	index logical	index physical	Ratio
		reads	reads	reads	reads	
NODE0	BMDEFAULTBP	1329	4	1342	1	99.8128
	SANBP	470540	18818	0	0	96.00077
	SANBP_CM	1919772	263025	3505941	19572	94.79152
	SANBP_FX	273197	45703	68118	603	**86.43306**
	SANBP_FS	74	2	5	3	93.67089
	SANBP_TM	5383864	95694	3283904	4207	98.84744
	SANBP_LT	259887	2005	37971	18	99.32082
	SANBP_CC	2333	0	567	2	99.93103
	SANBP_BO	64	0	0	0	100
	SANBP_PIN	9522	0	7722	0	100
NODE1	IBMDEFAULTBP	0	0	0	0	NULL
	SANBP	857916	8385	0	0	99.02263
	SANBP_CM	622534	121718	1204006	8199	92.88726
	SANBP_FX	96864	45168	84339	1217	**74.40164**
	SANBP_FS	0	0	36	0	100
	SANBP_TM	1202830	9233	392310	637	99.38125
	SANBP_LT	47234	2862	1326	1	94.1042
	SANBP_CC	297	0	65	0	100
	SANBP_BO	0	0	0	0	NULL
	SANBP_PIN	0	0	0	0	NULL

Figure 7: Bufferpool snapshot output

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

ENVIRONMENT DETAILS	
Configuration Parameter	
LOGBUFSZ	8
LOGFILSIZ	250
LOGPRIMARY	3
LOGSECOND	2
Database Snapshot	
Log pages read	1
Log pages written	26055
Log pages read/Log pages written	3.84E-05

Figure 8: Environment details

effect on reducing memory.

- The ratio of *log pages read* to *log pages written* is low, hence we can reduce the size to lower the number and check whether the problem persists. If yes, then we can try a lower log buffer size – see Figure 8.

The ratio between *log pages read* and *log pages written* should be as small as possible. An ideal value would be zero *log pages read* to a good number of *log pages written*.

Notes

The **vmtune** command gives and changes operational parameters for the Virtual Memory Manager and other AIX components.

The Virtual Memory Manager (VMM) maintains a list of free real-memory page frames. These page frames are available to hold virtual memory pages needed to satisfy a page fault. When the number of pages on the free list falls below that specified by the *minfree* parameter, the VMM begins to steal pages to add to the free list. The VMM continues to steal pages until the free list has at least the number of pages specified by the *maxfree* parameter.

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

If the number of file pages (permanent pages) in memory is less than the number specified by the *minperm* parameter, the VMM steals frames from either computational or file pages. If the number of file pages is greater than the number specified by the *maxperm* parameter, the VMM steals frames only from file pages.

Important column headings from **vmtune** output and their description are:

- *minfree* specifies the minimum number of frames on the free list. This number can range from 8 to 204,800.
- *maxfree* specifies the number of frames on the free list at which page stealing is to stop. This number can range from 16 to 204,800, but must be greater than the number specified by the *minfree* parameter by at least the value of *maxpgahead*.
- *minperm* specifies the point below which file pages are protected from the repage algorithm. This value is a percentage of the total real-memory page frames in the system. The specified value must be greater than or equal to 5.
- *maxperm* specifies the point above which the page-stealing algorithm steals only file pages. This value is expressed as a percentage of the total real-memory page frames in the system. The specified value must be greater than or equal to 5.
- *minpgahead* specifies the number of pages with which sequential read-ahead starts. This value can range from 0 to 4096. It should be a power of 2.
- *maxpgahead* specifies the maximum number of pages to be read ahead. This value can range from 0 to 4096. It should be a power of 2 and should be greater than or equal to *minpgahead*.
- *pd_npages* specifies the number of pages that should be deleted in one chunk from RAM when a file is deleted. The

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

default value is the largest possible file size divided by the page size (currently 4096). If the largest possible file size is 2GB, then *pd_npages* is by default 524,288. Tuning this option is really only useful for real-time applications.

BRINGING IT ALL TOGETHER

Tuning a DB2 UDB system, or any complex RDBMS, for optimum performance can be a lengthy process. This article and the previous one ('AIX and DB2 tuning essentials and best practices for DB2 performance', see *AIX Update* issues 119 and 120, September and October 2005) give you the basics of everything you need to know in order to get the best DB2 performance.

We have seen how, when a system encounters a performance bottleneck, to use monitor switches and various AIX tools to determine the problem domain. We have seen how to tune various DB2 UDB registry variables, DB2 Database Manager instance configuration parameters, and database configuration parameters that can have the biggest impact on performance. We have discussed best practices for creating tables, indexes and bufferpools, and looked at how bufferpools are being used to determine whether additional bufferpools or different bufferpool sizes would help. Choosing the proper tablespace type, extent size, and prefetch size – as well as keeping system catalog statistics up to date – round out the basics of performance tuning.

T S Laxminarayan (ts_laxminarayan@yahoo.com)
System Programmer (India)
© Xephon 2005

Cron time

One of the main tasks for a systems administrator is to schedule tasks. Whether it be overnight batch schedules or simple back-ups, there are many ways to achieve a relatively

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

simple task. Over the years, products such as Tivoli Workload Scheduler (TWS/Maestro) and Control-M have come along and, for a price, you can centralize and organize the control of all your scheduled tasks. These products have nice graphical front-ends and aids to help you achieve your aims. However, the basic cron/crontab has remained pretty much the same. Whilst I do not intend to reprint the manual on using **cron** and **at**, the following program will, hopefully, help in making sure that what is in your systems **crontab**/**at** list is really what you expected.

Every so often, I seem to come across a problem that emanates from a **cron** entry. Either a job just doesn't seem to run or the job runs but doesn't seem to have done anything. Maybe there was a typo and **cron** tried to execute a command that didn't exist. The possibilities are (almost) endless. (In *AIX Update* issue 69, July 2001, as part of the 'Tidy up before you go!' program, there is logic that will check the program entries in **cron** for validity.)

From AIX 5.1 a log file was introduced to log executions of programs from within crontabs (*/var/adm/cron/log*). A patch (v5.1 : IY46416) was then produced to enable administrators to keep track of which return code related to which program. However, **cron** still has some basic limitations, such as a planning feature and ease of use. It is these functions that are addressed here.

The following program reads the crontabs on a system and produces a chronological list of programs to be run. The program takes many parameters to enable you to work out what will happen in the future, be it tomorrow, next week, or next year. With no parameters, the current day is used and an asterisk indicates the next program to be run. A further option allows any file to be parsed rather than the current ones in the crontab directory. The parameters are:

- **-a xxx**, where *xxx* is one of Mon, Tue, Wed, Thu, Fri, Sat, or Sun; it produces a list of programs to be run on that day of the week.

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

- **-d nn**, where *nn* is a decimal number between 1 and 31 inclusive; it produces a list of programs to be run on that day of the month.
- **-f filename**, where *filename* is the full path of a file you wish to have parsed as though it were a crontab.
- **-m nn**, where *nn* is a decimal number between 1 and 12 and produces a list of programs to be run on the current day in the *nn*th month (can be used in conjunction with **-d**).
- **-y [cc]nn**, where *nn* is a two-digit year and *cc* indicates a different century.

One last note before some sample output. The program will also check any **at** jobs that are in the system. However, whilst the crontab part of the program will work on Sun, HP, and Tru64 Unix, because of AIX's handling of **at** (see *AIX Update* issue 85, November 2002, 'Understanding the **at** command'), it cannot detect **at** jobs on other flavours of Unix.

Example crontab (from a Sun machine I know):

```
#ident      "@(#)root      1.2Ø      Ø1/11/Ø6 SMI"
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#
1Ø 3 * * * /usr/sbin/logadm
15 3 * * Ø /usr/lib/fs/nfs/nfsfind
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1
3Ø 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/
gsscred_clean
#1Ø 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
2 2 * * Ø /sunday_task
2 3 * * 1 /monday_task
2 4 * * 2 /tuesday_task
2 5 * * 3 /wednesday_task
2 6 * * 4 /thursday_task
2 7,8,9 * * 5 /friday_task
2 12-15,18 * * 6 /saturday_task
19 23 * * * /usr/local/test.pl
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

On a Tuesday it produces:

```
Date to check is TUE 23 AUG 2005 21:13:00
( 2:01) [    root]              [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c
> /dev/null 2>&1
( 3:10) [    root]
/usr/sbin/logadm
( 3:30) [    root]        [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/
gss/gsscred_clean
( 4:02) [    root]
/tuesday_task
(23:19)*[    root]
/usr/local/test.pl
```

CRON_TIME.PL

```
#!/usr/bin/perl -w
$|=1;

#use strict;
use Getopt::Std;
use Time::Local;

use vars qw/ $opt_a $opt_d $opt_y $opt_m $opt_f /;
my $fortoday=0;
my $file;
my $is_today="n";
my $tempa;
my $tempb;
my $tempc;
my $usern;
my %users;
my  $crondir="/var/spool/cron/crontabs";
my @mins;
my @hrs;
my @days;
my @months;
my @wkdays;
my $mins;
my $hrs;
my $dom;
my $moy;
my $dow;
my @cmmd;
my $ccmd;
my %hash;
my
@month_name=("JAN","FEB","MAR","APR","MAY","JUN","JUL","AUG","SEP","OCT","NOV","DEC");
my  $anchortime=time();
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
my $timetocheck=$anchortime;
my  ($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,$currwday,
$curryday,$curridst)=localtime($anchortime);
my  ($origsec,$origmin,$orighr,$origmday,$origmon,$origyear,$origwday,
$origyday,$origidst)=localtime($anchortime);
my  @weekdays=("SUN","MON","TUE","WED","THU","FRI","SAT");
getopts('d:y:m:a:f:');
if ($opt_y) {
  if ( $opt_y =~ /^[0-9][0-9]$/ ) {
    $opt_y+=100;
  } elsif ( $opt_y =~ /^[0-9]{4}$/ ) {
    $opt_y-=1900;
  } else {
    print "Error : Invalid year input...must be either 2 or 4 digits\n";
    exit 1;
  }
 $timetocheck=timelocal($currsec,$currmin,$currhr,$currmday,$currmon,$opt_y);
}
if ($opt_m) {
  if ( $opt_m !~ /^[0-1]?[0-9]$/ ) {
    print "Error : Invalid month input...must be between 1 and 12\n";
    exit 1;
  } elsif ( $opt_m > 12 || $opt_m < 1 ) {
    print "Error : Invalid month input...must be between 1 and 12\n";
    exit 1;
  } else {
    $opt_m--;
  }
   ($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,
                  $currwday,$curryday,$curridst)=localtime($timetocheck);
 $timetocheck=timelocal($currsec,$currmin,$currhr,$currmday,$opt_m,$curryear);
   ($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,
                  $currwday,$curryday,$curridst)=localtime($timetocheck);
}
if ($opt_d) {
  if ($opt_d !~ /^[0-3]?[0-9]$/) {
    print "Error : Date (-d) must be numeric 1-31\n";
    exit 1;
  } elsif ( ( $curryear%4 eq 0 ) && $currmon eq 1 && $opt_d > 29 ) {
    print "Error : Invalid date ($opt_d) specified for month
$month_name[$currmon]\n";
    exit 1;
  } elsif ( $opt_d < 1 || $opt_d > 31 || (($currmon eq 3 || $currmon eq
5 || $currmon eq 8 || $currmon eq 10 ) && $opt_d > 30) || ($currmon eq 1
&& $opt_d > 28 && $curryear%4 ne 0) ) {
    print "Error : Invalid date ($opt_d) specified for month
$month_name[$currmon]\n";
    exit 1;
  } else {
     ($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
                  $currwday,$curryday,$curridst)=localtime($timetocheck);
  $timetocheck=timelocal($currsec,$currmin,$currhr,$opt_d,$currmon,$curryear);
     ($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,
                  $currwday,$curryday,$curridst)=localtime($timetocheck);
  }
}
if ($opt_a) {
  if ( $timetocheck != $anchortime ) {
    print "Warning : -a flag specified with other date flags, value will
be ignored\n";
  } else {
    $tempb=0;
    $opt_a=uc(substr($opt_a,0,3));
    foreach $tempa (@weekdays) {
      if ( $tempa =~ /^$opt_a/ ) {
        $tempc=$tempb;
      }
      $tempb++;
    }
    if ( $tempc < $currwday ) {
      $timetocheck+=((7+$tempc)-$currwday)*60*60*24;
    } elsif ( $tempc > $currwday ) {
      $timetocheck+=($tempc-$currwday)*60*60*24;
    }
  }
}
($currsec,$currmin,$currhr,$currmday,$currmon,$curryear,
                  $currwday,$curryday,$curridst)=localtime($timetocheck);
printf "Date to check is %3s %2d %3s %4d
%2d:%02d:%02d\n",$weekdays[$currwday],$currmday,$month_name[$currmon],$curryear+1900,$currhr,$currmin,$currsec;

sub get_when {
  my @entries;
  my @tempentries;
  my $entry=shift;
  my $units=shift;
  my $temp1;
  my $i;

  sub ranges {

    my $argd=shift;
    my $argtype=shift;
    my $outarray;
    my $toplim;
    my $lowlim=0;
    my @rentries;
    my $ix;
    my ($fromh,$toh)=split(/-/,$argd);
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
  if ( $fromh > $toh ) {
    if ( $argtype eq "w" ) {
      $toplim=6;
    } elsif ( $argtype eq "h" ) {
      $toplim=23;
    } elsif ( $argtype eq "m" ) {
      $toplim=59;
    } elsif ( $argtype eq "o" ) {
      $lowlim=1;
      $toplim=12;
    } elsif ( $argtype eq "d" ) {
      $lowlim=1;
      $toplim=31;
    }
    for ($ix = $fromh; $ix <= $toplim; $ix++ ) {
      push(@rentries,$ix);
    }
    for ($ix = $lowlim; $ix <= $toh; $ix++ ) {
      push(@rentries,$ix);
    }
  } else {
    for ($ix = $fromh; $ix <= $toh; $ix++ ) {
      push(@rentries,$ix);
    }
  }
  return @rentries;
}

if ( $entry =~ /^[0-9\-]+,[0-9]+/ ) {
  @entries=split(/,/, $entry);
  $i=0;
  foreach $tempa (@entries) {
    if ( $tempa =~ /^[0-9]+-[0-9]+/ ) {
      push(@tempentries,ranges($tempa,$units));
    } else {
      push(@tempentries,$tempa);
    }
  }
  @entries=@tempentries;
} elsif ( $entry =~ /^[0-9]+-[0-9]+/ ) {
  @entries=ranges($entry,$units);
} elsif ( $entry =~ /\*/ ) {
  $_=$units;
  SWITCH: {
    /m/ and do {
      @entries=(0 .. 59);
      last;
    };
    /h/ and do {
      @entries=(0 .. 23);
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
        last;
      };
      /d/ and do {
        @entries=(1 .. 31);
        last;
      };
      /o/ and do {
        @entries=(1 .. 12);
        last;
      };
      /w/ and do {
        @entries=(Ø .. 6);
        last;
      };
    }
  } else {
    push (@entries,$entry);
  }
  return @entries;
}

sub process_cronfile {

  my $cronf=shift;
  open(CF,"$cronf") || die "Cannot open crontab file $cronf\n";

  CRONFILE: while (<CF>) {
    next CRONFILE if /^\n|^#/;
    ($mins,$hrs,$dom,$moy,$dow,@cmmd)=split;
    $ccmd="";
    foreach $tempa ( Ø .. $#cmmd ) {
      $ccmd.=" ".$cmmd[$tempa];
    }
    $ccmd.="#$file";
    @mins=get_when($mins,"m");
    @hrs=get_when($hrs,"h");
    @days=get_when($dom,"d");
    @months=get_when($moy,"o");
    @wkdays=get_when($dow,"w");
    #
    # Now check if the entry is relavent to the day we are looking at
    # If entry fits then $fortoday should equal 3 by the end
    #
    foreach $tempa (@wkdays) {
      $fortoday++ if ($tempa == $currwday);
    }
    foreach $tempa (@months) {
      $fortoday++ if ( ($tempa - 1) == $currmon);
    }
    foreach $tempa (@days) {
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
      $fortoday++ if ($tempa == $currmday);
    }
    if ( $fortoday == 3 ) {
      foreach $tempa (@hrs) {
        foreach $tempb (@mins) {
          $mins=$tempa*60+$tempb;
          $mins=sprintf "%04d",$mins;
          push(@{$hash{$mins}},$ccmd);
        }
      }
    }
    $fortoday=0;
  }
  close CF;
}

open(PF,"/etc/passwd");
while ($usern=(split(/:/,<PF>))[0]) {
  if ( $usern =~ /^\+:/ ) {
    open(YP,"/usr/bin/ypcat passwd|");
    while ( $usern=(split(/:/,<YP>))[0] ) {
      $users{$usern}=1;
    }
    close YP;
  } else {
    $users{$usern}=1;
  }
}
close PF;
if ($opt_f) {
  die "Error : file $opt_f does not exist\n" if ( ! -e $opt_f );
  die "Error : file $opt_f is not a valid file\n" if ( ! -f $opt_f );
  $file="?";
  process_cronfile("$opt_f");
} else {
  opendir(DH,$crondir) || die "Cannot open crontab dir $crondir\n";
  CRON: while($file=readdir(DH)) {
    next CRON if ( -d $file );
    process_cronfile("$crondir/$file") if ( $users{$file} );
  }
  closedir DH;
}
$is_today="y" if ( $currwday == $origwday && $curryear == $origyear &&
$currmon == $origmon );
foreach $tempa (sort keys %hash ) {
  $hrs=int $tempa/60;
  $mins=$tempa-($hrs*60);
  foreach $tempb ( 0 .. $#{ $hash{$tempa} } ) {
    $usern=(reverse split(/#/, $hash{$tempa}[$tempb]))[0];
    $hash{$tempa}[$tempb]=(split (/#/, $hash{$tempa}[$tempb]))[0];
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
    if ( $is_today eq "y" && (( $hrs > $orighr ) || ( $hrs == $orighr &&
$mins >= $origmin )) ) {
      printf "(%2d:%02d)*[%8s]
%70s\n",$hrs,$mins,$usern,$hash{$tempa}[$tempb];
      $is_today="d";
    } else {
      printf "(%2d:%02d) [%8s]
%70s\n",$hrs,$mins,$usern,$hash{$tempa}[$tempb];
    }
  }
}
```

Phil Pollard
Unix and Tivoli Administrator (UK)
© Xephon 2005

A software distribution tool

Where I work, we do software distribution on client/server architecture consisting of an IBM mainframe, AIX/Unix and RS6000 systems, and Microsoft XP. Pronto is a tool I have developed using a combination of Delphi and Korn shell programming to simplify our work. Pronto gives us an opportunity to work in a user-friendly environment and automates software distribution.

Pronto consists of two parts. On the left side of Figure 1 is the software distribution starting monitor. And on the right side is the software distribution watching monitor.

The packages we distribute to Unix servers are all compressed tar files, for example *kisim9039.Z*. We have already developed the Unix Korn shell scripts to distribute these files and Pronto triggers or schedules these scripts by using a batch file that does **rexec** to our main server sb918, for example:

```
rexec sb918 -l  username -n < path of passwd file "/u/username/bin/
scriptname %1 %2 %3 >> outputfile 2>&1"
```

The script is now activated on sb918 taking the parameters %1, %2, %3 like cargo_list, plan_name, or max_rcp. The

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

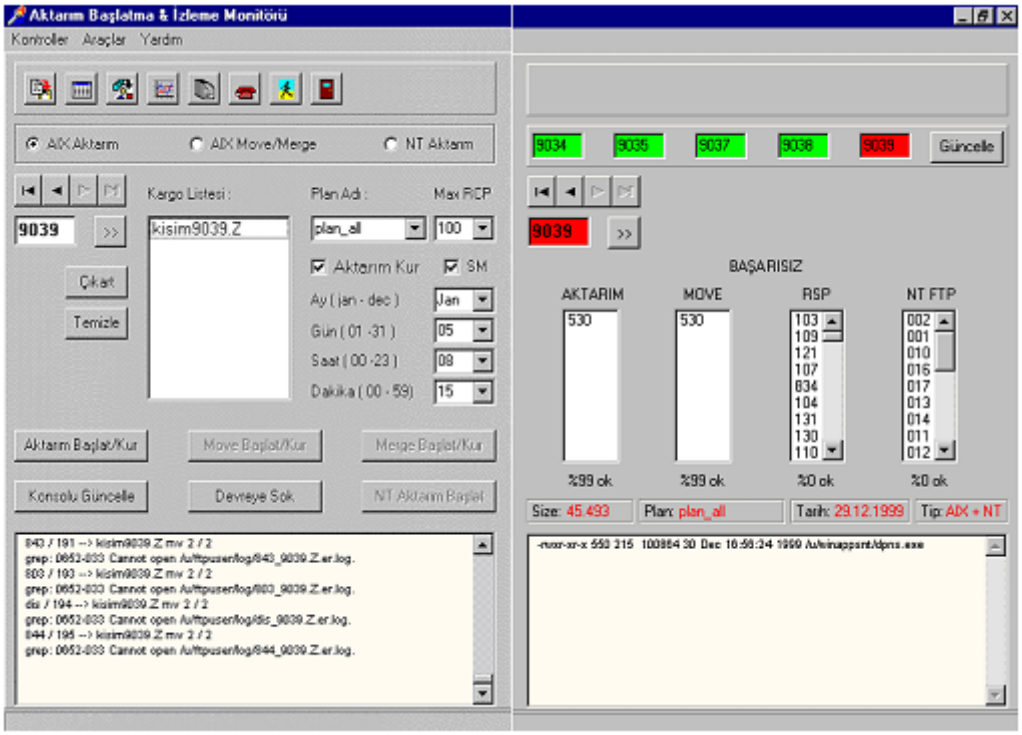

Figure 1: Monitor display

script starts the distribution and the results are written to our tables on our main server. A yellow rectangle visible at the bottom of the form is the Pronto console displaying standard and error output from the Unix scripts.

The right side of Figure 1 is the watching monitor. We need to control the distribution results on tables (DB2), which are updated by the Korn shell scripts. Pronto can reach the tables via ODBC. The packages are green when the distribution process is finished to both AIX and XP servers. It's red if distribution is not completed. At the top we can see the status of the last five packages. Software distribution is done in four steps – transfer of the data, moving it to a given path, extracting the package, and finally receiving the results from the remote servers saying whether the operation has been successful or not.

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

You can move around the package table by using the arrow buttons and every time you move you get the information about the package you have viewed.

In the example you can see information about 9039. The information you get includes: which servers are unfinished in which steps, completion percentages, size of the data, plan name, a list of the servers you want the data to be distributed to, the date the package was created, and its type. Some packages are distributed only to AIX servers or XP servers, and some to both. So type tells the type of server/servers the data is distributed to. Finally, the yellow rectangle lists the inside of the compressed tar package so that we can see which applications are distributed and where. When every step is finished with 9039 the edit box will turn green, meaning that the package has completed every step and distributed successfully according to the given plan.

The names of the files/applications that we will distribute are given to us in an Excel file. Converter (see Figure 2) shows us the inside of these files and converts them into comma-separated format so that our scripts can create the command files that our OS/2 servers will use for generation.

Pronto talks to Unix using ODBC. Pronto uses three tables: package, AIX server, and XP server. You can display these tables if you want to. These are shown in Figure 3. Unfinished steps are shown in red.

Below is the Korn shell script that is started from the Pronto menu.

LAUNCHER.SH

```
#!/bin/ksh
################################################
# transfers the schedule request coming from
# Pronto to swd.sh script.
################################################
# it is executed by swd3_sc.bat batch file
# from Pronto.
################################################
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

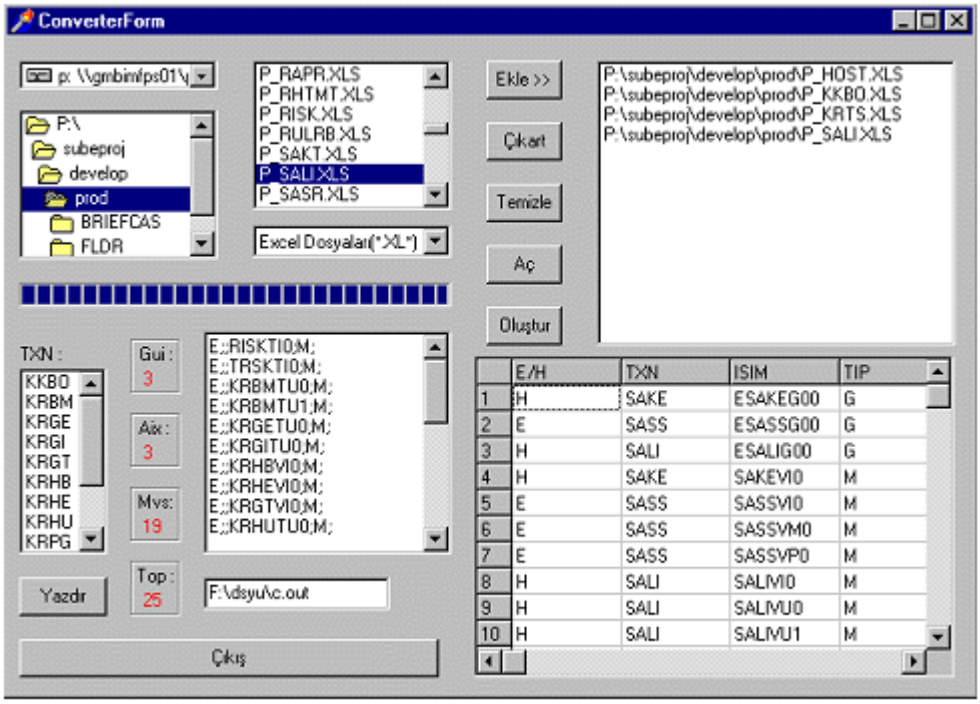

Figure 2: Converter example

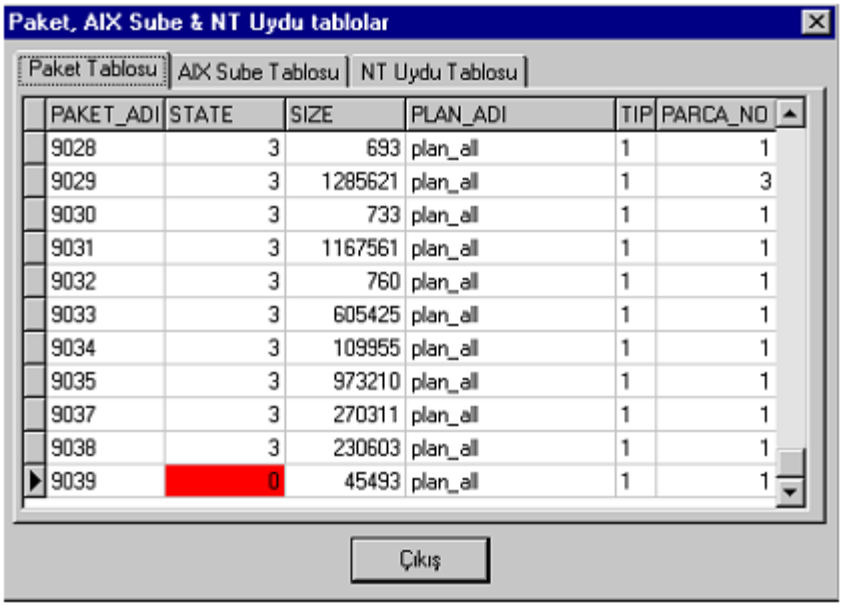

Figure 3: Displaying tables

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
# Adnan Akbas
# October 2003
################################################

function file_name_creator {
   'date' | read a b c d e f
   file_name_suffix=${b}${c}${d}
    cargo_list=/u/ftpuser/fileuse/cargo_lists/cargo_${file_name_suffix}
}
################################################
# main program
################################################
# parametreler
# 1:plan
# 2:max_rcp
# 3:cargo_list
# 4:Hour
# 5:Minute
# 6:Month (Jan-Dec)
# 7:Dyy (1-31)
sleep 5
plan=${1}
max_rcp=${2}
#############################################################################################
# makes cargo list a uniq file.
#############################################################################################
file_name_creator
cat ${3} > $cargo_list
#############################################################################################
# schedules swd3.sh script.
#############################################################################################
echo /home/ftpuser/bin/swd.sh $plan $max_rcp $cargo_list | at ${4}${5}
${6} ${7}
```

SWD.SH

```
#!/bin/ksh
# SWD version 2.1
# by Adnan Akbas
# auto-start version
#############################################################################
# parameter initialization
#############################################################################
sleep 5
plan_all_curr=1
#
wc -l /u/ftpuser/plan_all | read subelist_max junk
#
done_branch=0
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
#
plan=${1}
#
max_rcp_limit=${2}
curr_rcp_num=Ø
#
cargo_list=${3}
#
'date' | read a b c d e f
file_name_suffix=${b}${c}${d}
#
cat $plan > /u/ftpuser/fileuse/tmp/plan_${file_name_suffix}
plan=/u/ftpuser/fileuse/tmp/plan_${file_name_suffix}
#
cat $cargo_list > /u/ftpuser/fileuse/cargo_lists/
cargo_${file_name_suffix}
cargo_list=/u/ftpuser/fileuse/cargo_lists/cargo_${file_name_suffix}
wc -l $cargo_list | read c_l_size junk
#
function dist
{
       let cnt=Ø
       while read cargo_item ; do  # cargo list control
     print " $sube_kodu / $done_branch --> $cargo_item tekrar : $don / "
          /home/ftpuser/bin/dist.sh $sube_kodu $cargo_item > /dev/null &
             wait
             kisim_no=${cargo_item#kisim}
        grep completed /u/ftpuser/log/${sube_kodu}_${kisim_no}.log |gre2
             if [[ $t2 -eq 1 ]] ; then
                 let cnt+=1
             fi
       done < $cargo_list  # cargo list control
       if [[ $cnt -eq $c_l_size ]] ; then
  print ${sube_kodu} >> /u/ftpuser/fileuse/tmp/full_ok_${file_name_suff}
       fi
}

function control
{
 let curr=$plan_all_curr
#
    tail +$curr plan_all | read sube_kodu
    let done_branch+=1
    grep $sube_kodu $plan > /dev/null # plan control start
    if [[ $? = Ø ]] ;then  # plan control
       ps -ef -o ruser,comm | grep ftpuser | grep rcp | wc -l | read
curr_rcp_nt
       while [[ $curr_rcp_num -ge $max_rcp_limit ]] ; do
        sleep  4Ø
        ps -ef -o ruser,comm | grep ftpuser | grep rcp | wc -l | read
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
curr_rcp_m
       done # rcp control finish
       desicion=do_it
    fi # plan control finish
}
################################################################################
# MAIN PROGRAM
################################################################################
tekrar=2
let don=1
while [[ $don -le $tekrar ]] ; do
  while [[ $done_branch -lt $subelist_max ]] ; do
     desicion=do_not_it
     control
     if [[ $desicion = "do_it" ]] ; then
         dist &
     fi # desicion
     let plan_all_curr+=1
  done
  let don+=1
  #
  done_branch=Ø
  #
  plan_all_curr=1
  #
   ################################################################################
  # Finished AIX servers subtracted from the plan.
   ################################################################################
  sort -n -u -o $plan $plan
  if [[ -a /u/ftpuser/fileuse/tmp/full_ok_${file_name_suffix} ]] ; then
     sort -n -u -o  /u/ftpuser/fileuse/tmp/full_ok_${file_name_suffix} /
u/ftpuse
r/fileuse/tmp/full_ok_${file_name_suffix}
comm -23 $plan /u/ftpuser/fileuse/tmp/full_ok_${file_name_suffix} > /u/
ftpuser/fileuse/tmp/temp_${file_name_suffix}
     cat /u/ftpuser/fileuse/tmp/temp_${file_name_suffix}  > $plan
  fi
  wc -l $plan | read kalan junk
  if [[ $kalan -lt 5 ]] ; then
     exit Ø
  fi
  #
done #
```

DIST.SH

```
#!/bin/ksh
# Sends a file to a AIX server.
#
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
# By Adnan Akbas.
#
usage='dist.sh server_name file_name'
version='Ø.Ø9'
function trycmd {
  for trycmdcntr in 1 2 Ø ; do
    "$@"
    let ret=$?
    if (( ret != Ø )) ; then
      print "\a'$@' : error code=$ret"
      if [[ $trycmdcntr = Ø ]] ; then
        return 1
      else
        sleep $((trycmdcntr*trycmdcntr*1))
        continue
      fi
    fi
    print "'$@' completed."
    break
  done
  return Ø
}
function transfer {
for try in 1 2 ; do
  print "\n $bn ${try}. deneme $(date)"
  trycmd rcp -p $fn AN$bn:$fn 2>&1
  let ret=$?
  if (( ret != Ø )) ; then
    continue
  fi
  nfn=$yad${fn#kisim}
  print "\n $bn  Bitis $(date)"
  exit Ø
done
print '\a*** COPY OPERATION FAILED!!!\n\a'
return $ret
}
################################################################################
# main program
################################################################################
if (( $# > 3 )) ; then
  print "Kullanim: $usage" >&2
  exit 2
fi
bn=${1:?"server name not given"}
fn=${2:?"file name not given"}
yad=${3:-part}
if [[ ! -r $fn ]] ; then
  print "filename can not be read"
  exit 3
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
fi
################################################################################
# control
################################################################################
kisim_no=${fn#kisim}
logfile=/u/ftpuser/log/${bn}_${fn#kisim}
logfile=${logfile}.log
grep completed /u/ftpuser/log/${bn}_${kisim_no}.log | grep -c AN${bn} |
read2
if [ $t2 = 0 ] ; then
################################################################################
# checks if there is an rcp working for that server.
################################################################################
   ps -ef | grep ftpuser | grep rcp | grep AN${bn} > /dev/null 2>&1
   ret=$?
   if [[ $ret != 0 ]] ; then
      transfer >$logfile 2>&1 </dev/null
      exit $?
   fi
   exit 5
else
   print $(date) was sent before >>$logfile
   exit 7
fi
```

PRONTO

This is the Delphi program.

```
program Pronto;
uses
Forms,
Main in 'Main.pas' {MainForm},
MyUnit in 'MyUnit.pas',
FmxUtils in 'fmxutils.pas',
Tables in 'Tables.pas' {TblsForm};
{$R *.RES}
begin
{Application.Initialize;}
Application.CreateForm(TMainForm, MainForm);
Application.CreateForm(TTblsForm, TblsForm);
Application.Run;
end.
MAIN.PAS
unit Main;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
Dialogs,
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
ComCtrls, ExtCtrls, Menus, StdCtrls, Buttons, Mask, DBCtrls, Db,
DBTables,
MyUnit, FmxUtils;
type
TMainForm = class(TForm)
AktarimPanel: TPanel;
IzlemePanel: TPanel;
StatusBar1: TStatusBar;
MainMenu: TMainMenu;
kon1: TMenuItem;
Aralar1: TMenuItem;
Yardm1: TMenuItem;
SpeedPanel1: TPanel;
SecimGroup: TRadioGroup;
AixRBtn: TRadioButton;
MoveRBtn: TRadioButton;
NtRBtn: TRadioButton;
AKTNavigator: TDBNavigator;
IzlemeNavigator: TDBNavigator;
EkleSBtn: TSpeedButton;
GosterSBtn: TSpeedButton;
AktDBEdit: TDBEdit;
IzleDBEdit: TDBEdit;
CargoListBox: TListBox;
PlanCBox: TComboBox;
RCPCBox: TComboBox;
KURCheckBox: TCheckBox;
SMCheckBox: TCheckBox;
AyCBox: TComboBox;
GunCBox: TComboBox;
SaatCBox: TComboBox;
DakikaCBox: TComboBox;
AyLabel: TLabel;
GunLabel: TLabel;
SaatLabel: TLabel;
DakikaLabel: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Cikart: TButton;
Temizle: TButton;
AktBtn: TButton;
MvBtn: TButton;
MrBtn: TButton;
KonsolBtn: TButton;
DvrBtn: TButton;
NtAktBtn: TButton;
KonsolMemo: TMemo;
SpeedPanel2: TPanel;
IcerikMemo: TMemo;
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
Son5Panel: TPanel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Son5Guncelle: TButton;
AktListbox: TListBox;
MoveListbox: TListBox;
RSPListbox: TListBox;
NTListbox: TListBox;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Tablolar1: TMenuItem;
Versiyon1: TMenuItem;
GenerationSaylar1: TMenuItem;
k1: TMenuItem;
Converter1: TMenuItem;
ubeTelefonNumaralar1: TMenuItem;
Prosedrler1: TMenuItem;
ProntoHakknda1: TMenuItem;
Database1: TDatabase;
PaketTbl: TTable;
PaketDSource: TDataSource;
PaketTblPAKET_ADI: TStringField;
PaketTblSTATE: TIntegerField;
PaketTblSIZE: TIntegerField;
PaketTblPLAN_ADI: TStringField;
PaketTblTIP: TStringField;
PaketTblPARCA_NO: TIntegerField;
AIXQuery: TQuery;
AIXDSource: TDataSource;
NTQuery: TQuery;
NTDSource: TDataSource;
AIXQueryPAKET_ADI: TIntegerField;
AIXQuerySUBE_KODU: TStringField;
AIXQueryAKTARIM_STATE: TIntegerField;
AIXQueryMV_ER_STATE: TIntegerField;
AIXQueryRSP_STATE: TIntegerField;
NTQueryPAKET_ADI: TIntegerField;
NTQuerySUBE_KODU: TStringField;
NTQueryAKTARIM_STATE: TIntegerField;
NTQueryMV_ER_STATE: TIntegerField;
```


© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

```
NTQueryRSP_STATE: TIntegerField;
GroupBox1: TGroupBox;
Label17: TLabel;
SizeLabel: TLabel;
GroupBox2: TGroupBox;
Label18: TLabel;
GroupBox3: TGroupBox;
GroupBox4: TGroupBox;
Label19: TLabel;
TarihLabel: TLabel;
Label20: TLabel;
TipLabel: TLabel;
PlanLabel: TLabel;
AktPopupMenu: TPopupMenu;
MovePopupMenu: TPopupMenu;
Aktar1: TMenuItem;
AktarmLogunuGster1: TMenuItem;
MistralMove1: TMenuItem;
PegasusMove1: TMenuItem;
MoveLogunuGster1: TMenuItem;
ConvSBtn: TSpeedButton;
TblSBtn: TSpeedButton;
VerSBtn: TSpeedButton;
GenNoSBtn: TSpeedButton;
ProseSBtn: TSpeedButton;
TelSbtn: TSpeedButton;
AboutSBtn: TSpeedButton;
ExitSBtn: TSpeedButton;
{!~ Things done when the main form is created}
procedure FormCreate(Sender: TObject);
{!~ To determine color of DBEdit object}
procedure IzleDBEditChange(Sender: TObject);
{!~ To show unfinished FTPs in Listboxes}
procedure GosterSBtnClick(Sender: TObject);
{!~ Button to update console}
procedure KonsolBtnClick(Sender: TObject);
{!~ To empty the listboxes}
procedure IzlemeNavigatorClick(Sender: TObject; Button: TNavigateBtn);
{!~ Enables components if schedule done}
procedure KURCheckBoxClick(Sender: TObject);
{!~ Enable needed components and disabled the ones not needed}
procedure AixRBtnClick(Sender: TObject);
procedure MoveRBtnClick(Sender: TObject);
procedure NtRBtnClick(Sender: TObject);
{!~ To subtract the item selected from CargoCargolistbox}
procedure CikartClick(Sender: TObject);
{!~ Delete all items from CargoListbox}
procedure TemizleClick(Sender: TObject);
{!~ Add package number to CargoListbox}
procedure EkleSBtnClick(Sender: TObject);
```

© 2005. Reproduction prohibited. Please inform Xephon of any infringement.

```
{!~ Button to start Distribution}
procedure AktBtnClick(Sender: TObject);
{!~ Button to start Move}
procedure MvBtnClick(Sender: TObject);
{!~ Button to start}
procedure MrBtnClick(Sender: TObject);
{!~ Button to apply programs}
procedure DvrBtnClick(Sender: TObject);
{!~ Button to start XP ftp}
procedure NtAktBtnClick(Sender: TObject);
{!~ Start distribution to one server}
procedure Aktar1Click(Sender: TObject);
{!~ Start move to one server (mistral)}
procedure MistralMove1Click(Sender: TObject);
{!~ Start move to one server (pegasus)}
procedure PegasusMove1Click(Sender: TObject);
{!~ See the distribution log from the selected item}
procedure AktarmLogunuGster1Click(Sender: TObject);
{!~ See the move log from the selected item}
procedure MoveLogunuGster1Click(Sender: TObject);
{!~ Update Button for the last 5 packages}
procedure Son5GuncelleClick(Sender: TObject);
{!~ Menu Hints on StatusBar1}
procedure HintGoster(Sender: TObject);
{!~ Disable items in AktPopupMenu}
procedure AktPopupItemDisable(Sender: TObject);
{!~ Disable items in MovePopupMenu}
procedure MovePopupItemDisable(Sender: TObject);
{!~ Exit Pronto}
procedure k1Click(Sender: TObject);
{!~ Open TblsForm}
procedure Tablolar1Click(Sender: TObject);
{!~ Define HotKeys}
procedure FormKeyDown(Sender: TObject; var Key: Word;
Shift: TShiftState);
procedure Prosedrler1Click(Sender: TObject);
```

Editor's note: this article will be concluded next month.

Adnan Akbas
Senior System and Database Administrator
TURKCELL (Germany)

© Xephon 2005

© 2005. Xephon USA telephone (214) 340 5690, fax (214) 341 7081.

Hummingbird has announced Version 6.0 of Hummingbird Genio.

Hummingbird Genio is a data integration solution that spans the functional areas of data extract, transformation, and load (ETL), and enterprise application integration (EAI). At the core of Hummingbird Genio is an engine controlling the flow of data from various sources in a hub-and-spoke data exchange architecture. The product provides content integration capabilities between the Hummingbird Enterprise content management solution and external applications and content repositories. It allows migration and consolidation of content repositories without programming or data staging.

Genio 6.0 supports targeting and extraction into new data formats and environments including AIX 5.3.

For further information contact:
URL: www.hummingbird.com/press/2005/genio6.html.

* * *

BlackBall has announced Version 4.0 of SearchIn Server Editions, expanding the current SearchIn product line to include enterprise networks. SearchIn's technology enables instant file retrieval, regardless of format.

SearchIn indexes not just the contents of a file, but also the metadata that describes the specific details of unstructured files. So SearchIn users have all the benefits of search and retrieval, file management, and an unstructured database in one product, says the company.

SearchIn Server agents support AIX, HP-UX, and other platforms.

For further information contact:
URL: 206.171.112.59/b/nav//products/searchin/searchinserver.

* * *

Oracle and Zend Technologies have announced Zend Core for Oracle for AIX, Linux, Sun, and Solaris platforms. These versions are available as free downloads from Oracle Technology Network and from Zend's Web site.

The Zend announcement simplifies the process of integrating applications with PHP-based front ends with Oracle back-end technology. The Zend Core for Oracle is a pre-built stack of all the components needed to have PHP work with Oracle.

For further information contact:
URL: www.zend.com/core/oracle.
URL: www.oracle.com/technology/tech/php/zendcore.

* * *

IBM has announced System p5 Express servers, which are equipped with POWER5+ microprocessor technology and are specifically designed for the processing requirements of small to mid-sized companies.

The new servers range from a new 8-way server for scale-up environments and server consolidation, to dense rack form factors for e-mail, Web, file and print serving and dense clustering in scale-out environments, to new 'scale within' capabilities with fast, easy-to-use virtualization.

The p5 systems are available with a combination of AIX 5L, Red Hat, or SuSE Linux operating systems.

For further information contact:
URL: www.ibm.com/systems/p.