Living with P6 Databases, Part 2

by Ron Winter, PSP

August 19, 2014

Introduction

Two years ago, the first part of this series on maintaining P6 databases was well received. Who would have guessed that so many people were interested in Oracle/Primavera P6 databases? Perhaps it has something to do with keeping their business running.

Small companies using P6 stand-alone installations suffer from the poor training of their Information Technology (IT) personnel (i.e., **you**.) Whatever the motivation, we present a continuation of our series on fixing and improving your databases for P6 stand-alone users. This paper does not directly address the concerns of large, multi-user server installations.

The author is not in the business of providing general IT support. We have learned these lessons from supporting ourselves and our Schedule Analyzer™ software customers over the past ten years. Much of what we present here is drawn directly from the Oracle on-line Knowledgebase as well as from other sources. We do not imply any guarantee or offer of support for people reading this paper.

Before proceeding with any of the examples listed here, we strongly encourage the reader to first backup the entire Oracle or MS SQL database**.** Instructions for this easy procedure can be found in Part 1 of this paper. Then you will be able to reset everything back to the way it was before you decided to hire yourself as your IT Department. (Remember the part in the Abstract that says that I am not a person who is in the business of selling IT services?)

Overview

Users of Oracle/Primavera P6 software, including P6 Professional and Enterprise Project Portfolio Planner (EPPM) actually use two programs to accomplish their scheduling functions. The first software presents the visible face of P6 where schedules and data are viewed. The second is one that runs behind the scenes and provides the requested data to P6 and stores the information returned. This is called a Database Server.

When you install P6 on a stand-alone machine, both the P6 software and the database server (database for short) software are installed at the same time. For standalone installations, the type of database software can be one of two types; Oracle Express or MS SQL Express. There are other offerings, but not for stand-alone P6 installations.

Both Express software offerings are distributed ‘free’ and P6 can interface with either of them. Saying that they are ‘free’ is somewhat of a misnomer; kind of like saying that the car companies supply you with free air in the tires. The car would not travel very far without the free air and they would be unable to sell the car if they could not make it roll. Oracle/Primavera could not sell P6 without the accompanying database software.

The price of the database software does not prevent it from needing IT support. Large companies have personnel dedicated to just supporting the P6 database server. You are the IT support for your stand-alone P6 installation. It is necessary for you to understand the basics of the database server in order to keep P6 running smoothly. Because there are two types of database software with their own interfaces, we shall have to cover both types in this paper.

Types of Database Servers

P6 Version 5 & 6 stand-alone installations automatically installed the Microsoft (MS) SQL Express database. The more current P6 Versions 7 & 8 automatically install Oracle Express as the database. If the other type of database is already installed when you install P6, you can configure it to use your existing database instead of the default.

This means that you have the choice of which type to use. You even have a choice of what version of the database software to use. All of the available types and versions purposely ‘throttle-down’ their free Express software in order to not compete with their paid versions. Express software is limited to only using one CPU and 2 GB of RAM memory (even if you have more.) The only obvious way that they differ is in their maximum size and number of databases supported. These limitations are by far the most important properties and worth updating our previous paper with the versions that have been released since then.

Oracle Express

Oracle Express is the current default database and is likely to remain that way. The current version installed is called, “10g”. It only supports one database and is limited to 4 GB database size. Once you reach that very limited size limit, the entire P6 system freezes.

There is an Oracle Express 11g out now that has an 11 GB limit on the size of the database. This is almost triple the size of the current version used in P6. If it is installed, P6 will use this version but the trick is to get it installed.

The Oracle Knowledgebase [1] says that one must first install the full (paid) version of Oracle 11g and then use the Database Upgrade Assistant utility included with it to upgrade your existing Oracle Express 10g database and server to Version 11g. The 'chicken and the egg' question is, "How do you get a free server if it takes a paid server to make a free server?" I leave this answer to the professionals.

It appears that Oracle 11g is the only correct one to use with Windows 8. When you specify that platform, Oracle Express 11g is installed for a standalone system. Oracle Express 10g is currently installed for Windows 7. In addition the Oracle XE Home Page setting is missing from the Oracle XE 11g Windows All Programs folder. It can still be found directly at C:\oraclexe\app\oracle\product\10.2.0\server\Database_homepage.url.

Many of the installed Oracle Express database settings may be read from a text file called TNSNAMES.ORA and another called LISTENER.ORA. You can typically find this file in directory,
"**C:\oraclexe\app\oracle\product\10.2.0\server\NETWORK\ADMIN**\".
Use Windows Notepad or a similar plain-text editor to review.

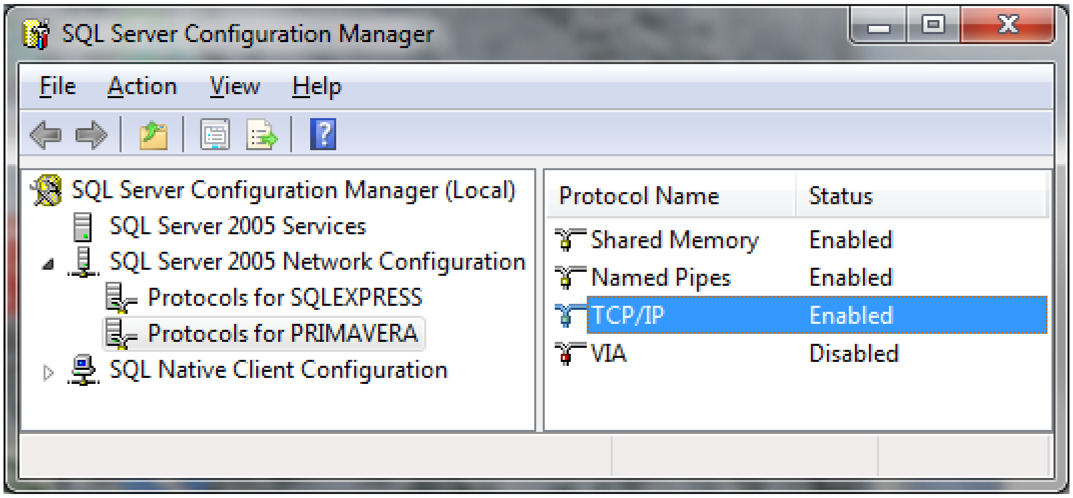
MS SQL Server

MS Server has a long history of versions used with P6. P6 Version 4 & 5 used MS Data Exchange (MSDE) and P6 Version 6 used MS SQL 2005 Express. MSDE was severely limited by its maximum of 2 GB database size and MS SQL Express 2005 was limited to 4 GB (just like the current Oracle Express g10.)

Beginning with MS SQL 2008 R2, the maximum database size has been lifted to a respectable 10 GB. Microsoft has since released the free versions MS SQL Server Express 2010 and MS SQL Server Express 2012. These newer versions also allow for a much roomier maximum database size of 10 GB.

In addition, all of the MS SQL versions also allow you to install multiple databases. This second feature difference from the Oracle XE database allows people with multiple clients to keep their databases separate from each other.

Determining the current SQL database host port can be difficult. The default number is 1433 but if a dynamic one is assigned, this number will change. To look up the current host port number, select the **Windows Start Button** / **All Programs** / **Microsoft SQL Server 2005** / **Configuration Tools** / **SQL Configuration Server Manager**. Expand the SQL Server 2005 Network Configuration root and left-click on your Primavera instance (‘Protocols for PRIMAVERA’ in this case.) Then left-click on the **TCP/IP** protocol as shown below in Figure 1.

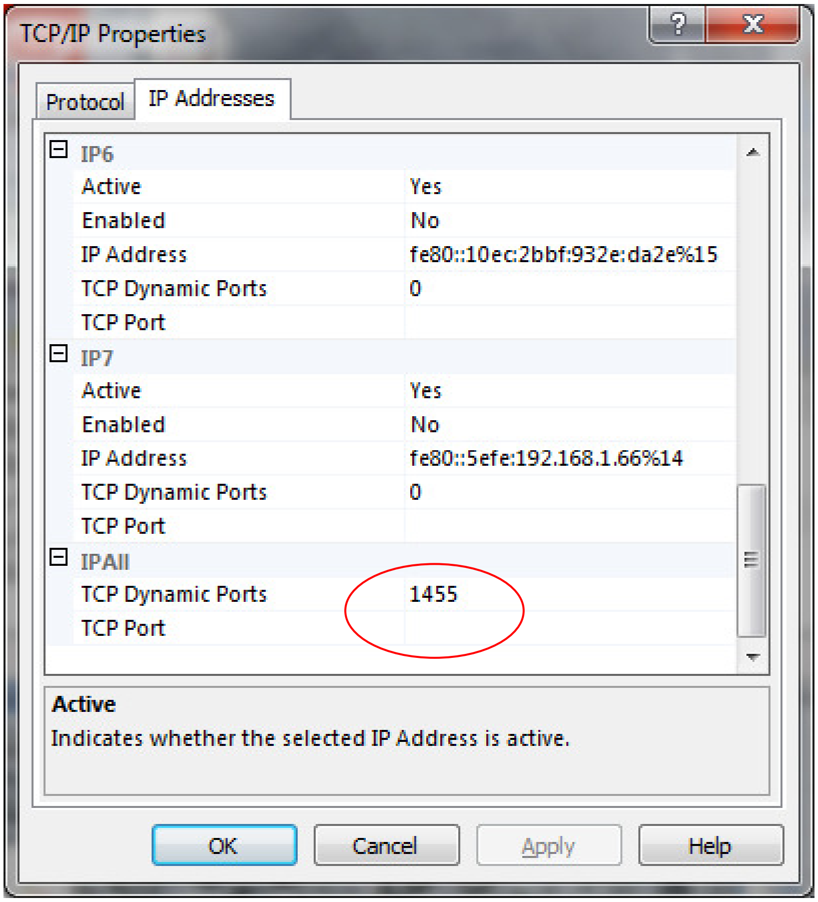

SQL Server Configuration Manager
File
Action
View
Help
SQL Server Configuration Manager (Local)
SQL Server 2005 Services
SQL Server 2005 Network Configuration
Protocols for SQLEXPRESS
Protocols for PRIMAVERA
SQL Native Client Configuration

Protocol Name	Status
Shared Memory	Enabled
Named Pipes	Enabled
TCP/IP	Enabled
VIA	Disabled

Figure 1: Configuring SQL Port

Please note that as the database software recognizes and requires a combination of using the left mouse button (left-click) and the right mouse button (right-click), we will only use the term, “click” to indicate that the reader should use the normal, default left mouse button. When we indicate that the right mouse button should be used, then we will say, “right-click”.

Confirm that the TCP/IP protocol is enabled (use right-click, **Enable** to change.) Now right-click and select, **Properties**. Scroll to the bottom of the IP Addresses Tab and locate the **IPAll** section as displayed below in Figure 2.



IP6	
Active	Yes
Enabled	No
IP Address	fe80::10ec:2bbf:932e:da2e%15
TCP Dynamic Ports	0
TCP Port	
IP7	
Active	Yes
Enabled	No
IP Address	fe80::5efe:192.168.1.66%14
TCP Dynamic Ports	0
TCP Port	
IPAll	
TCP Dynamic Ports	1455
TCP Port	

Active
Indicates whether the selected IP Address is active.

Figure 2: Determining SQL Port Number

The instance above is using dynamically assigned port 1455. This setting was assigned by the computer when the database was added, which allows for multiple applications using different ports. On the other hand, if that line is blank and the TCP Port line has a 1433 listed, the default port 1433 will always be used. Either way will work and should not be changed manually.

Correcting Database Full Problems

In Part 1 of this paper, we stated that there was no reliable 'fix' for exceeding the maximum allowable Express database size. Without warning, P6 simply stops working. We are now able to offer a little more hope. We cannot guarantee that these fixes will solve your problem; only that we forced a test computer into this 'database too large' error and was able to recover without having to restore the database.

The Oracle Knowledgebase [2] lists the following methods for reducing the database's size enough to allow you to log onto P6 and delete unneeded projects.

- Truncate the REFRDEL Table and then Compress the Database Storage
- Reducing the Size of Methodology Manager database (pre P6 V8)
- Deleting the Methodology Manager database
- Restoring the database from a previous backup
- Trimming history tables

Truncate the REFRDEL Table

The recommended option requires you to log onto the Database Manager and execute a few SQL commands. Oracle says that the instructions apply to P6 Version 7 and later, so Version 6 users may not be able to use these suggestions.

P6's 'undo' information is contained in the referential delete table, REFRDEL. This information can grow to be quite large and can be truncated to free up space. This should only be done with all users logged OUT of the Project Management database or database corruption issues may be generated. All retention of data in the REFRDEL table will be lost after this process has been completed, which only means that you will not be able to 'undo' prior P6 events. You will still be able to undo any new actions back to the time you ran this process.

When we performed this process, it reclaimed 10 MB on my system. We did not need to proceed to the two optional steps listed in the Knowledgebase document. This was enough to allow us to log into P6 and delete unneeded schedules or baselines.

Oracle Recommended Solution for Database Full

In the following examples, we are assuming that your database files are stored at the default location, "**C:\oraclexe\oradata\XE**\" on your computer. Look for several files that have the file name extension of ".DBF". You should check to see if this is correct.

1. Launch the Oracle Database Homepage by going to **Windows Start Button** / **All Programs** / **Oracle Database 10g Express Edition** / **Go To Database Home Page** and log in as the ADMPRM$PM account using the password you provided during installation (not your P6 admin password.)

Step 1: Clear out your 'Undo' Information.

2. On the Database Home Page, click **SQL** and then **SQL Commands.**

3. In the top window, type the command shown in Figure 3.

```
Truncate table REFRDEL;
```

Figure 3: Fix Oracle Database Full Script

4. If you typed it correctly, the words will automatically be colored in the manner shown above. Do not forget to include the semicolon at the end of the line. Now click the **Run** button. If successful, a message will appear in the bottom window stating "***Table truncated***"

Step 2: Compact your database storage:

5. Return to the Database Home Page. Click the **Administration** icon and then **Storage**.

6. On the **Tasks** list, click **Compact Storage**. The Compact Storage page appears and displays the following:

   Space Allocated: Displays the amount of space in Megabytes (MB) currently allocated to database user objects.
   Available: Displays the amount of space in MB currently available.
   Physical Limit: Displays the amount of space in MB available for database objects.
   Percent Used: Displays a percentage that indicates the space used in comparison with the maximum database size permitted.

MS SQL Express Solution for Database Full

To free up space in the database so that you can log back in using MS SQL Express database,

1. Select the **Windows Start Button** / **All Programs** / **Microsoft SQL Server 2005** / **SQL Server Studio Server Management Express** to access the login screen. You will be asked to identify your database and connection type. Thankfully, stand-alone users will find that the information is already provided if you elect the Windows Connection; otherwise use the **sa** account with a password of '**sa**'. Just click on the "**Connect**" button.

2. Expand the "**Databases**" folder on the left and highlight the project management database name. This might be, “pmdb$primavera” or just “PMDB.” Then click the "**New Query**" button at the top left. Then type the following script shown in Figure 4 in the blank window on the right-hand side.

```
truncate table REFRDEL
```

Figure 4: Fix MS SQL Database Full Script

3. If you typed it correctly, the words will automatically be colored in the manner shown above. Now, be sure that the correct database name is shown in the drop-down box to the right and below the New Query button and then click the **Execute** button to the right of drop-down box. The Messages tab should then display, “***Command(s) completed successfully***”.

4. Right-click on the PMDB database name and select **Task** / **Shrink** / **Database**. A new window should open up with the title, “**Shrink Database –** *(database name)*”. Click on the **OK** button to execute the procedure.

Compacting the storage recovers unused free space in the database. This is actually accomplished in the background using a database job and may take several minutes to complete.

The other optional recommended solutions should not be needed and will not be detailed here. They are available on the Oracle Knowledgebase as Document ID 1235263.1. Just type this into the query box to go directly to the instructions.

Now your total database size should be just slightly under the maximum limit. Log onto P6 and start deleting unneeded schedules, one at a time. You should now have recovered from the dreaded Maximum Database Size Exceeded error.

Background Job Processing

In order to speed up processing requests from P6 users, the database server does not actually delete records when you delete information like activities, relationships, or resources. These records are stored in the database in carefully measured ‘slots.’

When you delete a record, something else has to be put there. One method is to ‘slide’ everything up one to fill the empty record. As you can imagine, this would take a lot of time to accomplish, even at computer speeds. Multiply this by dozens or hundreds of users all deleting things at the same time and you can just imagine the potential for a really slow operation of P6.

When a P6 user deletes a task resource or even an entire project, the database server just marks each deleted record with the time and date and the person's user ID who deleted the record. This is called, a “logical delete.” When you ask P6 to display something, the request automatically also includes the requirement that it not show any record that is marked as being logically deleted. This condition can go one for some time, with the database just getting bigger and bigger as the deleted records build-up.

Remembering that the database server is just a program that runs all of the time, we can program it to periodically purge the logically deleted records, preferably when the number of people currently using the database server is low. This is called, Background Job Processing.

The database server has a series of jobs to perform to catch up on all of the delayed processes saved up over the span of time since the last background processing was accomplished. It does this on its own without any P6 user having to activate it or to wait for it to finish. This series of processes can take from milliseconds to hours, depending on the size of the installation and the length of time since the last run was accomplished. P6 standalone users typically ‘see’ the process take milliseconds to run.

In theory, skipping a background job processing session is no cause for alarm. We need it to run eventually but if it was scheduled weekly and the computer was turned off at the time chosen, nothing terrible should occur; well at least until P6 Version 8.3. The background processing job of logically deleted records before this version typically occurred once a week on Saturday. You would see a notice that the background processing was missed when the last time it was accomplished was longer than the set time interval. There is also a short background job that logically deletes log on information (and other short tasks) that runs every second. Sometimes, these settings use different time intervals.

Investigating your Background Job Settings

So how often does your P6 system run background processes and when was the last time? This information is important to know if you want to see if it is operating properly. This information is fairly simple (and fun) to ascertain.

Oracle Express Background Settings

To find out about your Oracle Express background processing,

1. Launch the Oracle Database Homepage by going to **Windows Start Button** / **All Programs** / **Oracle Database 10g Express Edition** / **Go To Database Home Page** and log in as the ADMPRM$PM account.

2. On the Database Home Page, click **SQL** and then **SQL Commands.**

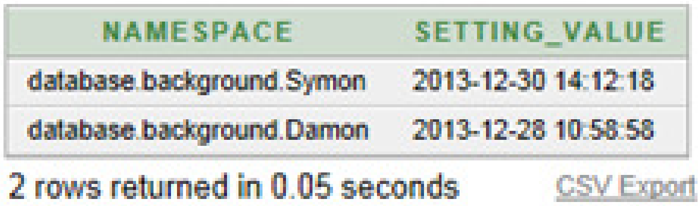
3. In the top window on one line, type the command shown in Figure 5.

```
select namespace, setting_value from settings where setting_name = ‘HeartBeatTime’;
```

Figure 5: Oracle Background Heartbeat Script

4. Do not forget to include the semicolon at the end of the line. Now click the **Run** button.

5. The Results section should display something like the that shown in Figure 6, which indicates that the short system monitor (SYSMON) process last ran on 30DEC13 at 14:12:18 hours and the longer data monitor (DAMON) process last ran two days earlier on 28DEC13 at 10:58:58 hours.



NAMESPACE	SETTING_VALUE
database.background.Symon	2013-12-30 14:12:18
database.background.Damon	2013-12-28 10:58:58

2 rows returned in 0.05 seconds

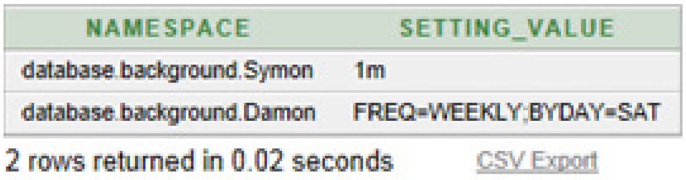
Figure 6: Oracle Background Heartbeat Time

6. Now return to the top window and change the word, “**HeartBeatTime**” to the word, “**Interval**” as shown below in Figure 7.

```
select namespace, setting_value from settings where setting_name = ‘Interval’;
```

Figure 7: Oracle Background Interval Script

7. Now click the **Run** button once more. This time the programmed time between the two background processes are displayed like in our Figure 8 example below.



NAMESPACE	SETTING_VALUE
database.background.Symon	1m
database.background.Damon	FREQ=WEEKLY;BYDAY=SAT

2 rows returned in 0.02 seconds

Figure 8: Oracle Background Interval

In the example above, we see that our SYSMON process is supposed to run every minute (‘1m’) and that the DAMON process is programmed to run once a week on Saturday. Other time abbreviations include: s = seconds and h = hours.

<u>MS SQL Express Background Settings</u>

To find out about your SQL Express background processing,

1. Select the **Windows Start Button** / **All Programs** / **Microsoft SQL Server 2005** / **SQL Server Studio Server Management Express** to access the login screen. Now login.

2. Expand the "**Databases**" folder on the left and highlight the database name. Then click the **New Query** button at the top left. Now you are ready to type the following script in the blank window on the right-hand side.

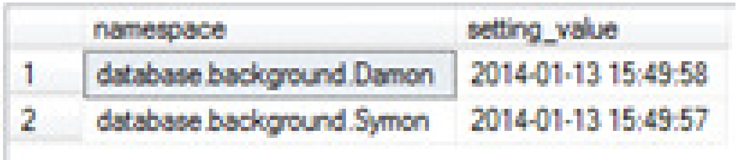
3. In the top window, type the command shown in Figure 9.

```
select namespace, setting_value from settings where setting_name = ‘HeartBeatTime’
```

Figure 9: MS SQL Background Heartbeat Script

4. MS SQL does not use a semicolon at the end of the line. Now click the **Execute** button on the screen.

5. The Results section should display something like the following Figure 10, which indicates that the longer data monitor (DAMON) process last ran on 13JAN14 at 15:49:58 hours and the short system monitor (SYSMON) process last ran one second earlier on 13JAN14 at 15:49:57 hours.



	namespace	setting_value
1	database.background.Damon	2014-01-13 15:49:58
2	database.background.Symon	2014-01-13 15:49:57

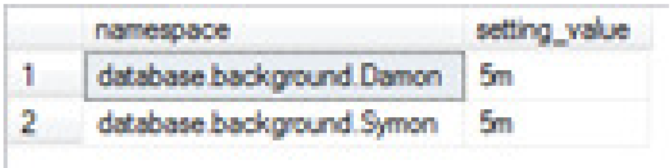
Figure 10: SQL Background Heartbeat Time

6. Now return to the top window and change the word, “**HeartBeatTime**” to the word “**Interval**” as shown below in Figure 11.,

```
select namespace, setting_value from settings where setting_ name = ‘Inerval’
```

Figure 11: MS SQL Background Interval Script

7. Now click the **Execute** button once more. This time the programmed time between the two background processes are displayed like in our Figure 12 example below.



	namespace	setting_value
1	database.background.Damon	5m
2	database.background.Symon	5m

Figure 12: MS SQL Background Interval

In the example above, we see that our SYSMON process is supposed to run every 5 minutes (‘5m’) and that the DAMON process is also programmed to run every 5 minutes. Other time abbreviations include: s = seconds and h = hours.

If your interval between processing the background jobs is greater than the last time the background ran (heartbeat time) or if the heartbeat time is missing, then you should suspect that you have a database background processing problem.

Manually Running your Background Jobs

Two things changed for P6 standalone users with Version 8.3. First, the data monitor (DAMON) time interval was changed from weekly to every 5 minutes. Secondly, the log on routine was changed to not ignore logically deleted logon IDs. This means that if your system did not process your background processes since you last logged-off, that the computer would incorrectly state that you were already logged-on and refuse to let you log on ‘again’. [3]

Oracle/Primavera is well aware of this change (we sent in an error report.) The support group blames the failed background processing for the error instead of the mistaken coding that fails to note and skip logically deleted records.

This explanation means that while P6 Version 6 and 7 users just get a warning about the failed processes and can continue working, that P6 Version 8.3 users are actually locked out of using P6 when the short process background job fails to run. Even if the background processing is working, the fact that it is set to occur every 5 minutes means that you will be unable to log out of P6 and then log back in without first waiting several minutes.

How simple is it to fix the background processing if it fails? Well, the process depends upon which version of P6 you are using as well as which version and type of database server. It will involve sorting through the Windows Control Panel (which changes which every version of Windows.) In short, it is a mess and far too complicated to detail in this paper. Oracle has dozens of Knowledgebase papers posted online on this subject. I have a sincere appreciation of the type of talent it takes to routinely fix this problem in various computers.

What this paper can do is to show you how to manually run the background processes so that you can at least log into P6 and get your work done. This process is actually fairly easy to do.

Manually Run the Oracle Express Background Process

To manually run the background processes on your MS SQL Express 2005 database,

1. Launch the Oracle Database Homepage by going to **Windows Start Button** / **All Programs** / **Oracle Database 10g Express Edition** / **Go To Database Home Page** and log in as the ADMPRM$PM account.

2. On the Database Home Page, click **SQL** and then **SQL Commands.**

Step 1: Manually run the System Monitor for Short Jobs

3. In the top window, type the command shown in Figure 13.

```
begin system_monitor; end;
```

Figure 13: Execute Oracle SYSMON Process

4. Do not forget to include the two semicolons. Now click the **Run** button. If successful, a message will appear in the bottom window stating "***Statement processed. 0.03 Seconds***".

Step 2: Manually run the Data Monitor for Longer Jobs

5. In the top window, change the word fragment, "**system**" to "**data**" as displayed in Figure 14.

```
begin data_monitor; end;
```

Figure 14: Execute Oracle DAMON Process

6. Do not change anything else from the previous line. Now click the **Run** button. You should see a statement saying, "**SQL Statement Running**".

7. After a grueling 23 seconds of worry and angst looking at the message, "**SQL Statement Running**", you will see something like shown in Figure 15 in the Results box (your situation will be slightly different),

```
Keep Date: 01/16/2014 03:52:09
Oldest Refrdel: 01/13/2014 11:18:40
Cleared: 1 records from 01/13/2014 11:18:40 to 01/14/2014 11:18:40
Total Rows Deleted/Changed: 0

Statement processed.

32.82 seconds
```

Figure 15: Oracle DAMON Response

Manually Run the SQL Express Background Process

To manually run the background processes on your MS SQL Express 2005 database,

1. Logon by selecting the **Windows Start Button** / **All Programs** / **Microsoft SQL Server 2005** / **SQL Server Studio Server Management Express** to access the login screen.

2. Expand the "**Databases**" folder on the left and highlight the database name. Then click the **New Query** button at the top left.

Step 1: Manually run the System Monitor for Short Jobs

3. The type the following script shown in Figure 16 into the blank window on the right-hand side.

```
Exec system_monitor
```

Figure 16: Execute MS SQL System Monitor Process

4. Do not include any semicolons. Now click the **Execute** button. If successful, a message will appear in the bottom window stating "***Command(s) completed successfully***".

Step 2: Manually run the Data Monitor for Longer Jobs

5. In the top window, change the word fragment, “**system**” to “**data**” as shown in Figure 17.

```
Exec data_monitor
```

Figure 17: Execute MS SQL Data Monitor Process

6. Do not change anything else from the previous line. Now click the **Execute** button.

7. After a short wait, you will see something like shown in Figure 18 in the Messages box (you situation will be slightly different.)

```
10 row(s) affected
Delete Date is 2014-01-09 03:58:27
Total Rows Deleted/Changed            0.

 (0 row(s) affected)
```

Figure 18: MS SQL DAMON Response

If you were locked-out of P6 due to “**User already logged-on**” error, you will now be able to log back in.

Speeding-up Database Operations

Many P6 users note that their system seems to be slowing down. What used to take a schedule 30 seconds to open now takes 5 minutes to perform the same task. You never noticed a delay before but now copy and paste operations now take a long time to complete. Most likely, the cause is database-related. There are some things that you can try to speed up P6 operations.

Summarize Projects

The Summarize Projects function in P6 allows you to display totals for various categories such the total number of activities in the schedule. These values can be displayed in the Projects Window. [4]

There is a second benefit to summarizing project; it speeds-up the loading of your list of projects. When you summarize projects, this information is saved in a special set of related tables for use next time. If the information in these tables exists, P6 will not bother to load and organize the project information directly but quickly retrieve this information from the summarization tables instead. This saves P6 from having to load a large amount of detail information, allowing you to get to work quicker.

To use the summary service to roll up and save resource quantity, cost, and/or custom user field information for projects at any level of the WBS or EPS, select the project or list of projects that you want to summarize and then choose **Tools** / **Summarize** and then choose **Open Projects** or **All Projects**.

P6 Professional can automatically summarize project data at regularly scheduled intervals using scheduled job services under **Tools** / **Job Services**.

This tip is more useful for large, multi-user systems but still has some performance enhancement to even Standalone Users.

Gather Statistics

Both Oracle and MS SQL sped-up their operation by watching how you use the system. They then try to improve their performance by organizing their internal operation to better match the way your use P6 and the parts that you most normally use. These information is saved as, 'statistics' and these statistics can grow to be quite large in size.

It is possible that your system is not performing this 'clean-up' process on its own. If so, manually causing MS SQL Express to gather and then update the usage profiles based upon these statistics should speed-up a system that has been running slower and slower.

Oracle Express

There is good news for Oracle Express users; beginning with Oracle 10g, there is a process that runs automatically each day to gather statistics; stats are automatically updated. All P6 users who installed the Oracle Standalone option are using Oracle 10g. This process should not be the cause of any system slow-downs..

MS SQL Express

MS SQL Express users can often see a speed-up of P6 database operations by manually gathering statistics. Use MS SQL Server Management Studio to run the queries. To manually gather statistics on your MS SQL Express database,

1. Logon by selecting the **Windows Start Button** / **All Programs** / **Microsoft SQL Server 2005** / **SQL Server Studio Server Management Express** to access the login screen.

2. Expand the "**Databases**" folder on the left and highlight the database name. Then click the **New Query** button at the top left.

   There is a second way to make sure that you are sending your commands to the correct database; add the command to specify a database as part of your query. As an example, say that your Oracle/Primavera database is named, "pmdb$primavera." Then the first line of your script would be as shown in Figure 19.

```
use pmdb$primavera
```

Figure 19: MS SQL Alternate Designation of Database

We are going to add this into the next example so that you will understand this method of making sure that you are running the right command on the right database.

3. Gather Statistics. There is a slight difference in scripts for MS SQL Express 2005 and those of later versions. Chose the correct version below.

   For MS SQL Express 2005, type the following script in the blank window on the right-hand side as shown in Figure 20.

```
use pmdb$primavera
update statistics project with fullscan ,full
update statistics task with fullscan, full
```

Figure 20: MS SQL 2005 Gather Statistics

For MS SQL Express 2008 and later, type the script shown in Figure 21 in the blank window on the right-hand side,

```
use pmdb$primavera
update statistics project with fullscan, all
update statistics task with fullscan, all
```

Figure 21: MS SQL 2008 Gather Statistics

4. For both types of MS SQL Express, click the **Execute** button. If successful, a message will appear in the bottom window stating "***Command(s) completed successfully***".

5. Apply Statistics. Delete the previous script and then type the following script in the blank window on the right-hand side as shown in Figure 22.

```
use pmdb$primavera
exec sp_updatestats
```

Figure 22: MS SQL Apply Statistics

6. Now click the **Execute** button. If successful, a message will appear in the bottom window stating "***Command(s) completed successfully***".

Note that updating your database statistics is not a ‘golden bullet’ guaranteed to fix your slowing P6 operations. There may be other issues causing your scheduling environment to perform slowly.

Speeding-Up Copy and Paste

One can copy and paste something as small as a letter and as large as an entire P6 project. Later, it is often possible to Undo this process and return everything to the way it was before. P6 accomplishes this small miracle by saving all changes in the Referential Delete Table (REFRDEL.) The data monitor is supposed to delete out the oldest information to keep the table from getting too big, but some settings and conditions can get in the way of this balancing process.

There are several cases where copy and paste of 400-activity schedules went from needing 30 seconds to over 6 minutes to perform. The RFRDEL table had close to 3 million entries. Besides using up your precious 4 GB of data space, it can also greatly slow down operations.

The same truncation solution to fixing the Database Is Full error explained at the start of this paper can be used to improve copy and paste speed issues (see Figure 3 or Figure 4, depending upon database type.) In similar cases, this procedure has completely cured the issue, restoring the system back to normal operations [5].

Corrupt Data

As enterprise scheduling programs such as P6 grow into massive database systems, the problem of database corruption becomes increasingly more relevant to software users. When features multiply and shortcuts are taken to improve response time, it is inevitable that subtle errors will creep into the database by overworked data systems.

Many of the software features use tokens in the database to represent one of many possible settings. The database itself does not oversee the process to ensure that only valid tokens are used. The busy software developer is often too harried in producing the latest schedule feature to invest time in devising a way to enforce only valid tokens being used because then they would have to develop some sort of process to report and correct the errors in addition to creating the new feature that they were charged with creating in the first place.

In general, the best that a software developer will do to deal with the issue of possible database corruption is to make sure that if a correct setting is not found, then a default one will be used. Without reporting on the database corruption, these errors are bound to multiply and untraceable errors based on assuming the wrong default setting will creep into the CPM scheduling process.

Oracle/Primavera lists the possible reasons for how data corruption may occur [6]:

- If the computer operator turns off the computer power without shutting down Primavera, it is the same as a power outage, and therefore can cause data corruption.
- If the computer operator uses the Task Manager of Windows to abort the Primavera program, without allowing it to shutdown appropriately, some of the data that was destined for the database may not get written, which again can cause data corruption.
- If all users were not disconnected from Primavera before copying, moving or backing up your database, then errors are likely to occur.

This list does not contain any suggestion for database corruption conditions that are not beyond the control of Oracle/Primavera. There are in fact several conditions that we have discovered beyond this list. One such known cause was the importing of XML schedule backups, especially under Version 6. XML import and export options were added to P6 beginning with Version 6. Version 6 corruption problems after importing XML formatted schedules became so bad that Oracle/Primavera cautioned users to not use XML until the software was updated to Version 7.

Our experience has indicated that the most common corrupt field in actual schedules being used in the field is the Activity Relationship Type field. This field can have one of four settings indicating either Finish-to-start, Finish-to-finish, Start-to-start, and Start-to-finish relationship types. When another setting is found in this field, P6 inevitably lists the relationship with a blank in the Relationship Type column. This missing data is not reported in the diagnostic and CPM calculations are still made (we assume using some default setting.)

In addition to incorrect entries in fields, not all data fields are required to be filled-in. In database parlance, this means that they may optionally be “Null.” We were quite surprised to see that the Activity Relationship table considers the relationship duration (often called a “lag”) as an optional field. In other words, a lag may be legally positive, zero, negative, or just plain missing.

If you are using P6 Version 8.2 or earlier, you can easily prove that P6 allows blank relationship lags to exist. Open a P6 schedule and display any activity’s relationship tab.

Look to the lag column of an relationship, highlight it, and then press the space bar. The field becomes and remains blank. Apparently, P6 Version 8.3 now prevents you from doing this. If that field is blank, you can move your cursor away from showing that activity and return at any other time and see that the activity's relationship remains blank. Re-schedule the project and the schedule report produced does not mention anything about there being a corrupted or missing relationship duration. We assume that P6 used that relationship in its calculation. We wonder what values it used when calculating the CPM?

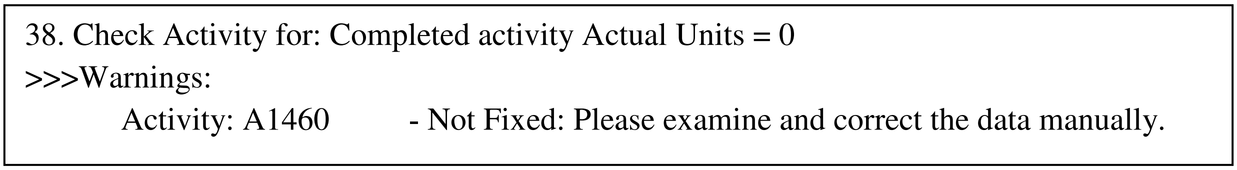
The key to this problem is to identify the missing or corrupt setting. Once the setting is known, you can fix that particular issue using P6. One only has to open the schedule, find the blank entry and replace it with a correct one. This will overwrite whatever bad information was there previously.

Prior to Version 8, P6 software had a feature called, "Check Project Integrity." While incomplete in its analysis and usually unable to do anything about the corruption when found, this process at least marks a beginning toward rational enterprise database management. Sadly, this feature has been removed from P6 Version 8 software.

What to do about P6 Integrity Check Results

The warnings produced during the P6 Integrity Check indicate that you should use P6 and go directly to that particular activity or other feature and look where indicated. You will see that something is wrong or missing (possibly a required field is blank such as no relationship type indicated in a relationship line.) Manually type in the correct information to have P6 fix the problem.

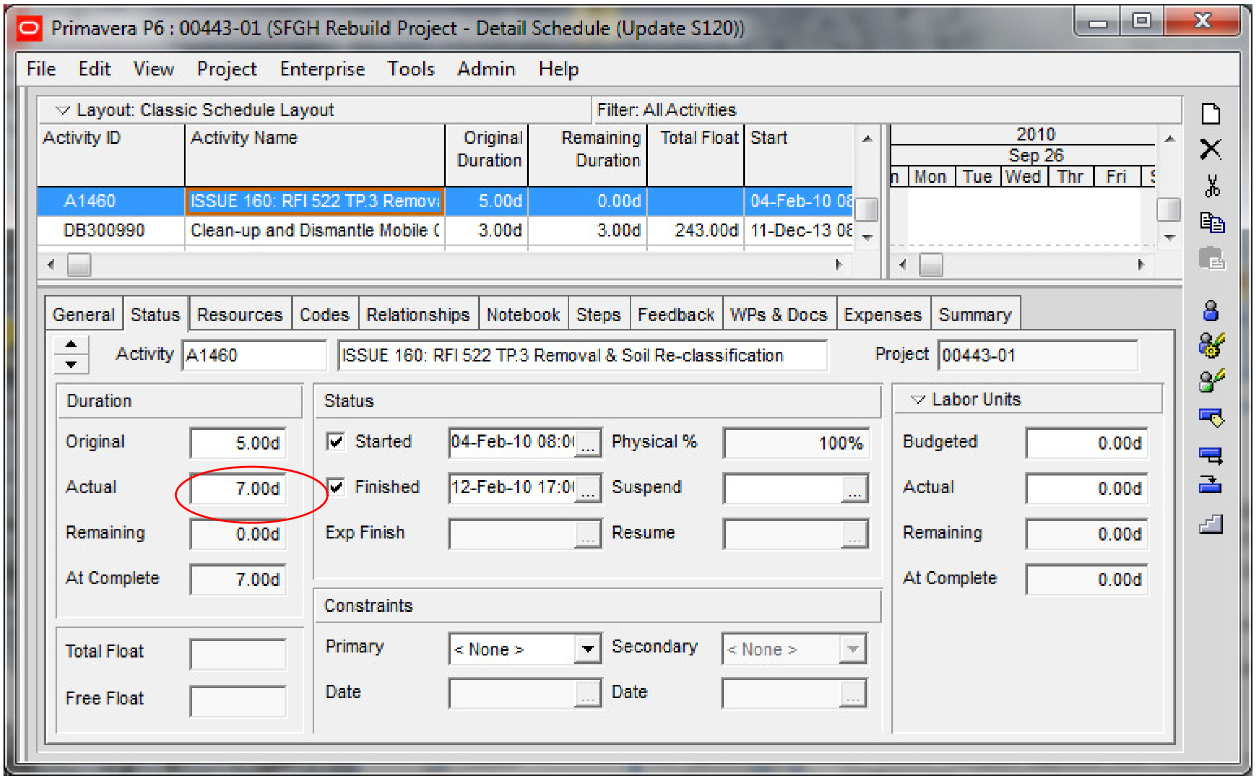
For example, Figure 23 displays an actual report result on a P6 project.



```
38. Check Activity for: Completed activity Actual Units = 0
>>>Warnings:
        Activity: A1460        - Not Fixed: Please examine and correct the data manually.
```

Figure 23: Integrity Check for Actual Units

Using P6, the user is supposed to find the activity indicated and enter the correct information (in this case, 7 days) in the Actual location as shown below,



Primavera P6 : 00443-01 (SFGH Rebuild Project - Detail Schedule (Update S120))

Layout: Classic Schedule Layout | Filter: All Activities

Activity ID	Activity Name	Original Duration	Remaining Duration	Total Float	Start
A1460	ISSUE 160: RFI 522 TP.3 Remov	5.00d	0.00d		04-Feb-10 08
DB300990	Clean-up and Dismantle Mobile (	3.00d	3.00d	243.00d	11-Dec-13 08

Activity A1460 ISSUE 160: RFI 522 TP.3 Removal & Soil Re-classification Project 00443-01

Duration
Original 5.00d
Actual 7.00d
Remaining 0.00d
At Complete 7.00d

Total Float
Free Float

Status
☑ Started 04-Feb-10 08:0 Physical % 100%
☑ Finished 12-Feb-10 17:0 Suspend
Exp Finish Resume

Constraints
Primary < None > Secondary < None >
Date Date

Labor Units
Budgeted 0.00d
Actual 0.00d
Remaining 0.00d
At Complete 0.00d

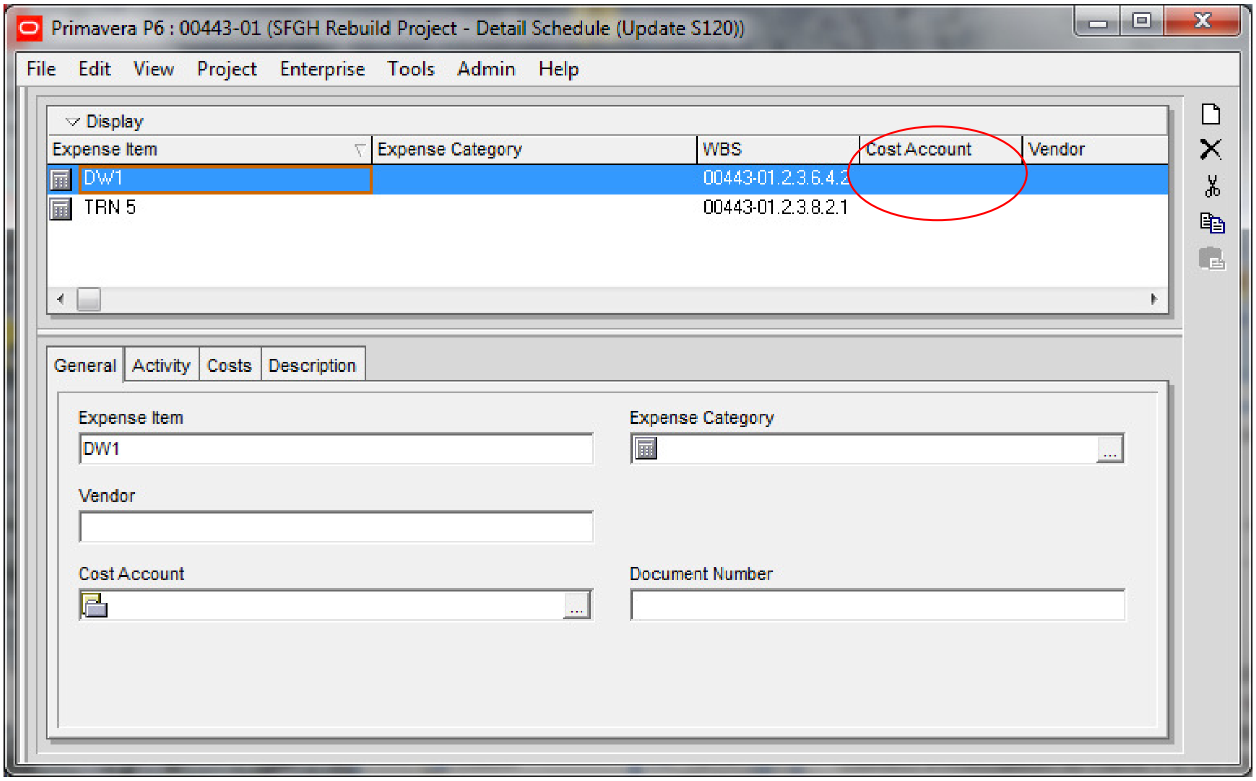
Figure 24: Corrected Example for Missing Actual Units

Another example of a Check Integrity report notification is shown in Figure 25,

```
62. Check Project Expense for: Expense with no cost account
>>>Warnings:
        Project Expense: AB009840, DW1 - Not Fixed: Please examine and correct the data manually.
```

Figure 25: Integrity Check for Project Expenses

In this case, display all Expense Items as shown in Figure 26, locate the one listed (in this case, “**DW1**”) and insert a cost account as appropriate,



Expense Item	Expense Category	WBS	Cost Account	Vendor
DW1		00443-01.2.3.6.4.2		
TRN 5		00443-01.2.3.8.2.1		

Expense Item: DW1

Expense Category:

Vendor:

Cost Account:

Document Number:

Figure 26: Example of Corrupt Expense Item

Summary

Now that your IT staff has had its second P6 Database training session, you can feel competent to set them to work on maintaining your standalone P6 system. They are prepared to deal with database full problems, backup job processing, speeding up sluggish systems, and overcoming corrupt data issues. Don't forget to remind them about the requirement to backup the entire database before beginning work.

Even if you have a service agreement with a Primavera Authorized Service Provider, you will now be in a better position to identify and better explain your P6's database problems. This will translate to quicker and better service. You might even save some money by covering the simple things first. We wish you and your IT staff a pleasant journey through the world of databases!

About the Author

Ronald M. Winter, PSP has over 20 years of experience in construction scheduling, delay claims preparation, programming, information management, training, and project management in construction and manufacturing industries. He has developed and implemented CPM scheduling and Management Information Systems for an impressive variety of projects. Since 2002, Mr. Winter has presented numerous papers at professional conventions such as AACE International, PMI College of Scheduling, Primavera User Conventions, and the Construction CPM Conferences. Ron has served in the past as the Chair of the AACE Planning & Scheduling Subcommittee and AACE North American Marketing Committee. He is also recognized as a Fellow of AACE.

Since 1997, Ron Winter has developed and distributed the popular software packages, Schedule Analyzer™, Schedule Analyzer Forensic™ and PROJECT WATCH™. Please visit the website at http://scheduleanalyzer.com/ . You can contact Ron Winter, PSP via the web at ron@ronwinterconsulting.com.

References

1. Oracle Knowledgebase, “Frequently Asked Questions (FAQs) on Upgrades to 11g (11.1 11.2)”, Document ID 727736.1
2. Oracle Knowledgebase, “ORA- 12952: The Request Exceeds The Maximum Allowed Database Size of 4 GB”, Document ID 1235263.1
3. Oracle Knowledgebase, “This user is already logged in. Please use another login name" When Logging Into Project Management or Methodology Management”, Document ID 891328.1
4. Ten Six Consulting, “Oracle Primavera P6 Professional – Summarization, by TheP6Pro, http://www.tensixconsulting.com/2012/04/oracle-primavera-p6-professional-summarization/#Primavera
5. Oracle Forums, “Project copy and paste taking longer with each iteration”, https://forums.oracle.com/thread/978644
6. Oracle Knowledgebase, “How Can Data Become Corrupted?”, Document ID 898588.1