

# P6 File Corruption

**by Ron Winter, PSP, FAACE and  
Marina G. Sominsky, PSP**

**Copyright © October 16, 2018**



## Introduction

The subject of the Oracle® Primavera P6<sup>SM</sup> File Corruption is a multi-faceted issue involving both reality and perception. Practically every P6 scheduler and claims consultant will, at some time, ask themselves if they are seeing this fault occurring. Real instances of corrupt data in the P6 database can be identified and some possible solutions sometimes exist. In other cases, the information only appears to be corrupt due to P6's complexity. In these cases, a scheduler needs to be armed with a list of potential causes for unusual occurrences.

This paper is applicable to both the Oracle® Primavera P6 Professional<sup>SM</sup> and the Oracle® Primavera EPPM<sup>SM</sup> software offerings. They are collectively referred to as "P6". It is applicable to both stand-alone and multi-user installation types. This paper solely focuses on major causes of database data corruption, the way to identify it and use available utilities to troubleshoot, repair or apply workarounds.

P6 Critical Path Method (CPM) scheduling software consists of three major components: The P6 Client, the Database Server and its Databases. The P6 Client is the visible portion of the software and enables users to access the project management database to plan and control projects. The Database Server is a separate piece of software that communicates with the P6 Client and creates and maintains the P6 project management database. Finally, a Database is required to store all of the P6 data.

The P6 database holds an enormous amount of data: millions of dates, durations, documents, and values in more than 180 tables. Whether MS SQL, Oracle, or SQLite is used to manage the P6 database, some corruption of the data is almost inevitable. Sometimes specific instances of data corruption do not affect the performance or accuracy to the schedule. Most of the time, the user does not even know that the problem exists.

Most of the solutions described in this paper are directly based on Oracle Knowledgebase articles, have been thoroughly tested, and applicable for both Oracle Express and MS SQL Express databases. Nevertheless, it is strongly recommended to backup the P6 data in question prior to executing any of the procedures listed. Some of the procedures described here should not be attempted by P6 users without a through IT background. The authors are not responsible for any damage or delay due to the misapplication of techniques described here.

## What is Database Corruption?

A database is made up of tables that are normalized. This means that redundant information is eliminated, which saves space and makes database operation magnitudes of times faster. There are a lot of ways that normalized tables can become 'corrupt.'

### *Record Normalization Integrity*

Instead of storing the activity's name in every table that references the activity data, it is stored once in one table and then 'hooked up' to the necessary tables to say, list a relationship ID with its successor's name. If the relationship record cannot find the related activity table record, then the information is unusable and it is said to be corrupt. If more than one activity record matches the relationship record, then this is also a corruption, because now it may generate different information on subsequent requests.

### *Required Fields Integrity*

Records have required fields and optional ones. The required fields are usually links to other tables. Sometimes they are necessary to define the record (like activities need to have Activity IDs.) If the record's field information is missing, it is called "NULL". Required fields should never be set to NULL.

### *Field Token Integrity*

Record fields sometimes represent distinct states. For example, relationship types can only be one of the four types of Starts and Finishes. If this field is set to say, "FRED", then this field is corrupt.

### *Other Possible Corruption Issues*

Backup, export files can also be corrupt. This can cause a failure to restore the schedule or may restore the schedule incorrectly. Backups may also be incorrectly imported, such that they corrupt the existing scheduling environment.

## **How Does This Matter?**

When a P6 database is corrupt, a wide range of occurrences can result. The most obvious result is when P6 is unable to open the schedule. More subtle results may show when the schedule provides the user with incorrect information, such as incorrect dates, costs, and totals. Worse yet is when the program displays different results at different times using the same information.

When a schedule database becomes corrupt, sometimes only peripheral information, such as attached documents or logs, are affected with lost or scrambled information. If the user is lucky, the database corruption may not cause any observable errors or loss of information.

The first line of defense against any of the above occurrences is to prevent corruption when possible. The next step is to learn how to detect it. P6 is not particularly strong in notifying the user of possible database corruption. Finally, the smart P6 user and P6 Administrator should learn how to repair the damage once discovered.

## Causes for Corruption

The number of causes of database corruption is large and varied. Loss of power, errors within the software program, and multi-user complexity are among the leading causes.

### *Loss of Power*

Like any database-oriented system, P6 is particularly sensitive to database corruption when the physical world interrupts the writing or reading process. This can be the result of a micro-second glitch in the electrical power or a complete unexpected loss of power. Even after the computer power is restored, the database thinks the user is still logged on. The following message is displayed: "This user is already logged in. Please use another login name." when trying to log on to P6. The cause of this error is from records that track the user while logged in. Because they did not get deleted in the proper manner, the system does not allow 'another' user to login under the same user name. The technical fix is to log onto the database server program and execute direct commands to locate and delete the user session. [1]

A simpler way to fix this problem is to locate the P6 initialization file, PM.INI (usually located at C:\Program Files\Oracle\Primavera P6\P6 Professional\PM.INI) and open the file with Notepad or another text-only editor program. Locate the line, "DeleteCurrentUserSessions=FALSE" and change the FALSE to "TRUE" (in capital letters) and then save. Reboot the computer and re-try the P6 login. This time, it will notify the user that the user is already logged on and do you want to clear the user? Answer, 'Yes' and the P6 user can now log on.

### *Errors in the Software Program*

P6 is a hugely complicated set of programming instructions, many operating at the same time. Software errors (or, 'bugs') can cause a database corruption. Even correct programming instructions that interfere with other programming instructions in unforeseen ways can cause database corruptions.

Finally, P6 communicates with a database server to have it store and/or retrieve the information. The database server is a set of programming instructions, which can also be subject to software bugs, as well as simple mis-configurations of the database structure. These have been known to cause database corruptions. Add to this the complexity of servicing requests from multiple users, who may not even be running P6 but some other data intensive program such as accounting, and the chances for database corruptions increases dramatically.

## *Common Database Corruption Instances*

Research has uncovered numerous instances of corrupt relationship table records. The most common is when the relationship type is present, but not one of the four allowable types. One cannot compute the CPM dates without knowing which relationship type is indicated. Apparently P6 must assume some setting (such as "FS") and continues on without even mentioning this in the calculation report.

Another form of observed relationship corruption is a missing relationship duration (or lag.) When this field is set to NULL, it is unknown what value is used during the CPM schedule calculation. The most likely 'guess' is zero, but once again, there is no notification of the assumption in the scheduling log.

The workaround offered by Oracle involves exporting the schedule as an 'Activity Relationship' type in the form of a spreadsheet, making sure that the 'Lag' field is selected to be part of the export template. While it becomes relatively easy to review the results in an Excel spreadsheet for indications of relationship corruptions (including blank fields for Lag), it is a lot more efficient to be able to conduct similar analysis right out of P6.

## **Corrupt Backup Files**

There are two basic types of P6 backup files used to export and import schedules from one P6 database to another: XER and XML. They are called this partly because this is the name of their file extension. Both are text files, but work differently. XER backups have existed since the original P6 Version 1, and the Extensible Markup Language (XML) version was introduced in P6 Version 6. Oracle Primavera has announced their intention to 'phase out' XER backups and have already completely eliminated this type of import/export facility in EPPM Version 8, but has so far retained it in P6 Professional Version 15.

An obvious symptom of corrupt backup files is when the backup file will not load the schedule into the receiving P6 system. This can be caused by an error in creating the original backup file or if the file has later been modified during copying/transmission. In this case, P6 will display an error message during the loading process and the schedule will either be non-existent or incomplete in the receiving system.

Another instance of 'corrupt backup files' is when the import incorrectly overwrites existing data that should not have been changed. The most often seen error of this type is the modifying of existing Global Calendars.

The default setting used causes existing Global Calendars to be replaced by the importing files if they have the same name. This overwrite could possibly change the scheduled dates in every other existing schedule in the system. The other option setting of not importing the Global Calendars

may cause the incoming schedule to not compute the same dates as the same schedule on the exporting side.

The correct 'solution' to this problem is to only use Project Calendars and to first import the unknown-quality schedule into a test database to inspect and possibly convert the calendars before importing it to the user's P6 database.

Even more difficult to observe and thus detect is the instance where data is lost during the import, but no warning is given. In this case, activities, relationships, or other data may be lost without the user being aware of the fact. It is extremely difficult for the user to spot such changes in a schedule with thousands or even tens of thousands of activities.

### *XML File Corruption*

Some of the most common corruption issues involve bad relationship records. Research has indicated that many of the corrupt relationship records are caused by XML imports. Corrupt XML files are most prevalent in P6 Version 6 when the 'new' type of export/import format was introduced.

In fact, even Primavera eventually recommended that users not use XML import until P6 Version 7 was released. In P6 Version 6.2 the existing hierarchical XML importer/exporter schema was replaced by a new flat XML functionality. XML import accuracy has improved greatly since then, but patches continue to be issued including fixes to the XML system.

In their P6 Version 7, "Frequently asked Questions" document, Primavera states, "It is recommended to use the Primavera XML format (for importing files without corrupting Global data) which adheres to security settings and can lock down the global data during import."

It is important for P6 users to understand differences between XML and XER import/export in order to decipher exceptions and errors, ultimately making informed decision about which option to select. The main difference is that P6 XML import/export uses the Application Program Interface (API), which enforces user's security privileges on import. In fact, according to Oracle's documentation, XML import will never corrupt data. [2] This is because the XML importer will not allow invalid data to be imported.

### *XER File Corruption*

XER files are tab delaminated text files. There have been several dozen XER-related bug fixes released by Primavera over the past 16 years that P6 has existed. POBS Table Corruption is a current file corruption issue that was identified last year and fixed in the most recent releases.

### POBS Table Corruption

According to the Oracle/Primavera Knowledgebase, the Public Operational Breakdown Structure (POBS) Table should exist in the database, but should also be empty. The P6

functions related to the POBS table have not been implemented as yet, and the table may even be removed in a future release. The POBS Table Corruption issue manifests itself as extremely long XER export and import times, while the POBS table information is being processed. Further research finds the existence of hundreds, thousands, or even more records in the POBS table. [3]

These thousands of POBS records are exported every time a schedule is exported in XER format. They are then added to any of the existing thousands of POBS records in the receiving database during an import. Both groups of POBS records are then exported with the next XER backup made. It does not matter which schedule is being exported, the POBS table is always backed-up along with it.

In other words, the POBS table is kind of like a virus; ever growing larger and increasingly slowing down imports. This ever-growing size might eventually become an important issue for standalone P6 systems with the old 4 GB space limit. The XER import time can increase from 30 seconds to 30 minutes for the same schedule as the XER file steadily becomes larger. Note that the XML export and import process seems to be immune to the POBS corruption issue.

One can purge the POBS table from the database, and one can delete the incoming information out of the XER file. The purging of infected XER files can be performed manually [4], or by 3<sup>rd</sup> party software [5][6][7]. The cleaning of the POBS Table itself is a more complicated 'manual' process best undertaken by IT personnel using the process described in Oracle Doc ID 1967569.1.

The most recent update to this Knowledgebase document dated 29FEB16 says that the issue has been fixed by programming the software to not import or export any existing POBS data. This fix does not actually remove the bad data from the database, but prevents its further spread. The fixed versions of P6 include r8.3.12.0, r8.4.8.0, r15.1.5.0, r15.2.1.0, r16.1.0 and all later releases. P6 versions earlier than r8.3 (including Versions 7 and 6) have not been fixed.

Research has indicated that the POBS corruption issue appears to be caused by P6's failure to remove POBS records marked for deletion during its housekeeping duties. Studies routinely show that more than 99% of the POBS records found are set for deletion. Obviously, if the P6 housekeeping processes that 'cleans-up' the database were to clean-up the POBS table, then only a small number of the existing records would remain and the problem would be minimized.

## Misdiagnosing Corrupt Database Tables

Not every apparent instance of database corruption is in fact a case of corruption. P6 is such a dynamic and complex set of inter-connected functions and processes that quite often the P6 user

does not understand or recognize the cause of the unexpected output. P6 may be working correctly, but the user does not appreciate the cause of the result observed.

### *Schedules Apparently Reverting to Earlier Versions*

Some P6 users suspect P6 database corruption when their data reverts to previous, earlier settings after importing the schedule. Screen layouts may reflect past settings, schedules may display dates for earlier updates. [8]

In this instance, the apparent corruption is caused by the XER import settings that incorrectly overwrites the new data with old. This is easier to do than one would think, because the default modify settings on the import setup is to update with the old table data. Reviewing the XER import action settings requires an additional, non-required step. In the Update Project Options screen, click on the highlighted Layout Name and then the Modify button. The Modify Import Configuration screen lists all import data types and what P6 is supposed to do (Action). Clicking twice on any action will reveal the drop-down options,

- Do Not Import
- Insert New
- Keep Existing
- Update Existing

The default, Keep Existing will ignore the new, updated information in favor of keeping the older. It is very difficult to maintain good quality control on imports as the action settings are not automatically shown during the import. The user must consciously check them to be sure before importing.

What is not widely known is that P6 does create an import log of actions performed. This log is stored in the Windows temporary directory of the user. If the user name is "Ron", then the import log file is usually located at C:\users\Ron\AppData\Local\Temp\PrmlImportLog.txt . The user must locate and save a copy of this file if permanence is needed, because the file is automatically replaced with the results of the next XER import by that user.

### *Schedule is 'Frozen' and Immobile*

One common complaint about P6 refusing to follow CPM rules is the result of the P6 function, External Constraints. The suspect schedule does not respond to 'what-if' scenarios. Activity dates refuse to change when all succeeding predecessor activities are advanced. Activities generate negative float, even though they should have plenty of positive float. [9]

The 'frozen' activities respond as if activity date constraints are assigned. The user is positive that no constraints are entered for that activity, and cannot identify any such constraints upon investigation. What may not be known, is that under certain circumstances, P6 automatically generates these constraints without the user's knowledge or request. Users who never intended to insert such constraints may possibly be harboring dozens of them in their schedules.

## External Constraints

Multiple P6 schedules can be related to each other using external relationships. This allows the schedules to be operated as a fragment of the entire enterprise schedule (i.e. fragnet), or combined together to compute a full CPM schedule of all related schedule parts. When the user creates an XER export of the group of related schedules, these external relationships are exported as well.

The issue is what happens when a schedule that has external relationships is exported without the accompanying schedules? P6 tries to force the schedule to maintain its computed dates and structure by automatically replacing the missing external relationships with external date constraints. It does this without notifying the user that it has performed this function.

Once created, the external constraints can be located and deleted from the schedule, if desired. The constraints can be avoided altogether if the scheduler first checks the schedule calculation setting, “Ignore relationships to and from projects” before creating the export. It is not obvious to the casual P6 user that this setting also applies to XER exports.

External Constraints are difficult to observe. They are not listed in the section where the other activity constraints are located. To locate and manage them, the user must add the “External Early Start” and “External Late Finish” columns (under the Date tab) to the activity view and sort on these. Another method of finding imbedded external constraints is to review the schedule log to find the section reporting this condition.

### *Spurious LOE Actuals and Durations*

In the past, some P6 users have complained of apparent database corruption issues involving the summary level of effort (LOE) tasks. In some of these cases, the LOE tasks show an actual start and/or actual finish dates when they have no physical progress recorded and none of the tasks they are spanning have actual dates. In other cases, the LOE task durations unexpectedly go to zero. This is usually the result of external relationships and not of database corruption. [10]

In the first case, LOE activities, which are apparently only related to un-started activities, mysteriously acquire an Actual Start indication and date. We know that LOE activities automatically acquire an actual start of the earliest activity being summarized with an actual start assigned. This includes activities in separate schedules that may be related to the mysterious LOE activity via an external relationship. Even when the other schedule is not present, this actual start will persist.

The instance of LOE durations reverting to zero may also be from external relationships. In this case, if the CPM calculation setting of, “Ignore relationships to and from other projects” is checked and no other activities in the current schedule are related to the LOE, then the duration of the ‘orphaned’ LOE task reverts to zero.

### *All Durations Go to Zero on Import/Export with P3*

Another instance of durations reverting to zero on standard activities occurs after importing or exporting schedules between P3 and P6. The Original Durations and Remaining Durations on Task Type activities are puzzlingly changed to zero, even when the Duration Type is set to Fixed Duration and Units/Time. [11]

The cause of this apparent file corruption problem is driving resources. In the activity's resource tab window, the task resource has the Resource Driving box checked. This should be unchecked before export. During the conversion from P6 to P3, the drive activity dates by default is checked, which causes the activity to have durations driven by the missing (non-existent) resources.

### *Added/Modified Audit User is "NOTPRMUSER"*

Almost every record in the P6 database has audit fields that indicate by whom and when the record is created and last modified. Several of these fields are viewable by P6 users. The 'who' name in these fields is the P6 login name of the person performing the action.

Confusion arises when the 'who' field states, "NOTPRMUSER". [12] This 'non-user name' is not a corruption, and is applied instead of the P6 user name in the following instances:

- When P6 Job Services has created or updated this particular record,
- The "Automatically fix errors" option was used when running Check Project Integrity, and
- The project has been scheduled in P6 Web Access (which is a known bug).

### *Wrong User ID Corruption*

P6 allows some apparent duplicates in login names, because the P6 login name is case sensitive. It does not allow addition of a second user with the exact same name, yet it will allow the same letters to be used with different capitalizations. For example, "adam", "Adam", and "ADAM" may all exist on the same P6 system. To P6, those are three different names. This is not the standard practice from other software makers. For instance, Oracle and MS SQL logins will be fooled because they are case insensitive to User IDs. This case insensitivity issue only pertains to P6 logon IDs, not to P6 logon passwords.

## **What Can Oracle Primavera Do About This?**

There are quite a few things that Oracle Primavera can do to improve database integrity. Unfortunately, progress is proceeding in the wrong direction; fewer tools instead of more. Oracle Primavera needs to bring back the Check Project Integrity utility and to improve it. The old checks need to be modified, so that naturally occurring situations, which are not the result of database corruption, are not reported (called 'false positives'.) A better explanation of what the user should

do once the check is complete in order to repair the conditions detected should also be made available.

When P6 encounters a corrupt data instance, it needs to report this fact to the user. When it encounters something as basic as missing data (say in a relationship's lag value) it needs to make an entry to this fact in the scheduling log that it currently working on, and should state what assumption was made in order to proceed.

Operation of P6 should be written to industry norms. This means that User IDs should be case insensitive, so that P6 User IDs work along the same guidelines as Oracle database logins. One would have thought that this would have been on the 'purchase checklist' of a database company such as Oracle when they bought Primavera Inc in 2009.

Oracle Primavera needs to prevent the POBS Table virus from spreading. This is not simply a 'feature enhancement' that needs the appropriate number of requestors before the software development team is directed to work on it. This is a virus that is building in size and spreading around to most P6 world-wide installations. Stand-alone installations are filling-up their allotted database size and major installations are experiencing one half to an entire hour load times for a single XER file.

The solution to this virus problem is actually quite simple, since there is no use for the POBS table - stop trying to import the POBS table. A better solution possibly lies in understanding the nature of the problem and deal with that by having the P6 database housekeeping duties delete POBS records already scheduled for deletion. Then they would not build-up over time.

## What Can P6 Users Do About This?

It is important to go over some best practice recommendations which can be used to properly manage P6 databases, because often times, these tend to be overlooked:

- Have a concise data backup and archival strategy,
- Use at least two P6 databases – one for the production data, and one as a 'testing/cleansing' area,
- Maintain separate production databases for managing different types of projects, [13]
- Use the Primavera XML importer to minimize corruption of global data, and
- Check the P6 XER Import Log, "PrmImportLog.txt" after importing an XER file to see if errors occurred.

However, even the most diligent followers of the above recommendations are not fully protected from losing and/or corrupting a project, critical layout or report. There are several options available to P6 users who need to investigate, troubleshoot and possibly address data corruption issues:

- Check Project Integrity utility,
- XER File Parser & Builder,
- P6 Validate.bat utility,
- Primavera – PRIM (MOSC),
- Database management utilities, and
- 3rd Party Oracle/Primavera Providers.

### *Check Project Integrity Utility*

Check Project Integrity is a P6 utility used to identify and fix errors or inconsistent data in the database. It performs two types of integrity checks: Data Join and Data Value. The Data Join integrity check examines the entire P6 database for data join inconsistency, e.g. when activity code assigned to a task does not match activity code stored in the database. Data Value integrity check scans only currently open projects looking for inconsistencies such as 'Actual Units = 0' on completed activity resource assignments.

Starting with P6 Version 8.0, the Check Project Integrity utility has been removed from the Project Management Client Application. However, P6 users can still perform a Data Join integrity check against the database by running single queries directly from the database server. A complete list of the scripts and Data Join Integrity check details can be found in the Oracle Knowledgebase [14]. The document lists 62 Data Value integrity checks that used to be part of Check Project Integrity before P6 Version 8.0. This list can be used as a reference to create custom reports and/or layouts that help identify issues with the schedule.

The Data Value integrity check routine that is still available to P6 users with release prior to P6 Version 8.0 has its own problems. It pinpoints some instances as integrity errors that are not so (i.e. false positives.) It reports on events that are sure to occur, like activities without a predecessor relationship or ones without a successor relationship (Checks # 36 & 37.) Other error reports include events that can occur through normal use. The following checks should be ignored when reported:

9. Check Activity Resource Assignment for: Completed assignment Actual Units = 0
10. Check Activity Resource Assignment for: Assignment with no cost account
11. Check Activity Resource Assignment for: Assignment with Remain Units/Time = 0
12. Check Activity Resource Assignment for: Assignment with Price/Time = 0
13. Check Activity Resource Assignment for: Assignment with resource Price/Time or Overtime Factor out of sync
34. Check Activity for: Activity does not have primary resource
38. Check Activity for: Completed activity Actual Units = 0

Prior to P6 Version 8, this utility could be run through the Project Management Client Application by choosing **Tools, Check Project Integrity**. Options to '*Automatically Fix Correctable Errors*' and

'Log to File' could also be selected. The results could be saved to a log file (overwritten each time a data integrity check is performed) and reviewed.

It is comforting to have the software automatically correct the corruption errors found. This does not mean that the correction will be accurate, only that it will no longer be identifiable as a corruption error. For instance, if an Actual Finish date is found that is earlier than its Actual Start date, the auto-correct response is to set the Actual Finish date equal to the Actual Start. If an accurate As-Built schedule is desired, this is most likely an error. The correct Actual Finish date should be researched and manually entered.

Even if the 'Automatically fix correctable errors option is checked,' the report will most likely state that the errors are not fixed. This is due to the fact that the P6 program does not know what the correct entry is supposed to be. The software may note that the project's WBS setting does not correspond to any known WBS, but what is the correct one? That is up to the P6 user to decide.

The proper procedure for fixing the corruption is for the P6 user to reference the activity, project, or cost account mentioned in the error report and find the field mentioned as being corrupt. This field will most likely be displayed in P6 as a blank. The P6 user should click on the field's down arrow and select the correct setting. This selection will be transferred to the P6 database and the corruption will be overwritten.

Listed below are some of the pros and cons of using the Check Project Integrity utility:

Pros of Check Project Integrity utility:

- Free tool,
- Provides 21 Data Join integrity checks and fixes some of them,
- Can be included in a database maintenance plan (SQL) or scheduled job (Oracle.) [15]

Cons of Check Project Integrity utility:

- Most of the problems identified must be corrected manually,
- Automatic 'fixes' for 62 Data Value checks are not accurate,
- Removed from the **Tools** Menu starting with P6 Version 8.0,
- Checking and fixing of Data Join issues must be done manually via database server starting with P6 Version 8.0,
- Does not check for Data Value type problems (e.g. corrupt relationship types) starting with P6 Version 8.0,
- Provided by Oracle 'as is' without warranty.

### *XER File Parser & Builder*

The XER File Parser & Builder utility distributed by Oracle Primavera is helpful for fixing corrupted XER files by allowing the user to open the P6 project in a customized Excel workbook, run

diagnostics and, subsequently, remove orphaned records from database tables in question. P6 users can download 'XERFileParser&Builder.xls' from the location referenced in this document. [16]

To illustrate how XER File Parser & Builder can be used to clean up and re-export a corrupted P6 project, consider the example of a failed import due to orphan records in TASKACTV table. The Import log file states the following:

```
Import Failed: PRM DBAccess Error: Database Server Error: The statement has been terminated.
Cannot insert the value NULL into column 'actv_code_id', table 'P6_ITPRJ.dbo.TASKACTV';
column does not allow nulls. INSERT fails.
```

Figure 1: XER Import Log Excerpt

To resolve this problem, the P6 user should use the XERFileParser&Builder.xls utility and select, 'Load XER file' on the 'General' tab to import the problem file. Next, click on 'Diagnostics' and select 'cross check FK' (where FK stands for 'foreign key'). In the 'Select Table' field, choose 'ACTVCODE' parent table from the drop-down (because this is where activity codes are stored). In the 'Select Field(s)' list, select the parent table key 'actv\_code\_id' (because import file log points to this column specifically). In the 'Select Secondary Table', choose 'TASKACTV,' and in the 'Select Secondary Field(s)' choose 'actv\_code\_id'. Figure 2 displays this action being taken.

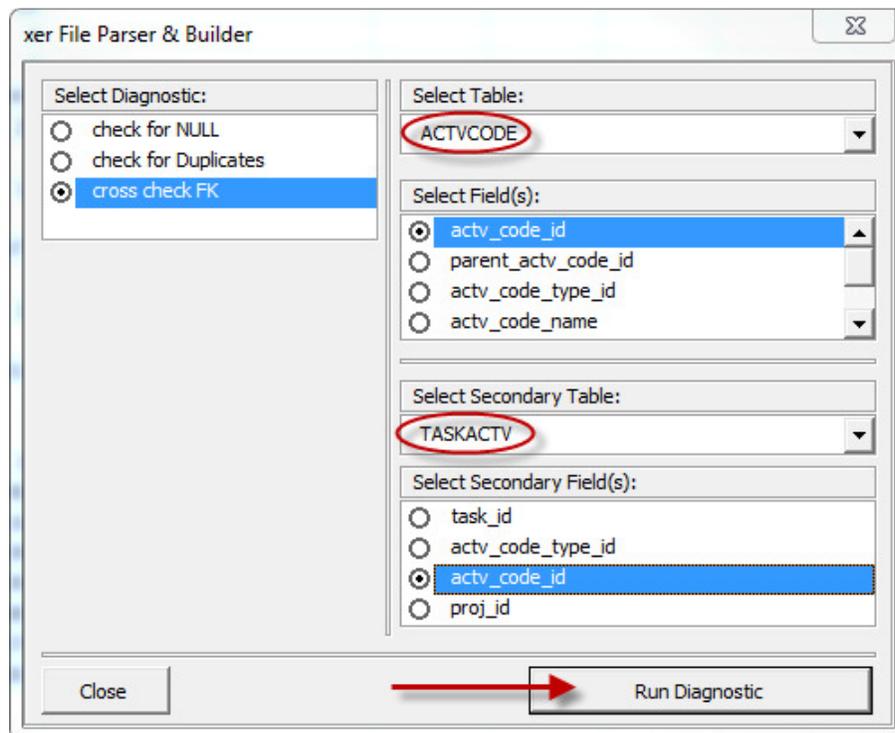


Figure 2: XER File Parser Screen

The goal is to run a diagnostic that will create a list of 'actv\_code\_id' values in the 'TASKACTV' table where the corresponding parent record does not exist on the parent table 'ACTVCODE.' Then, the P6 user must go to the 'TASKACTV' tab in 'XERFileParser&Builder' worksheet and delete all the

rows with those orphan 'actv\_code\_id' values. The last steps are to create a new XER file with 'Build XER file' and re-import into P6.

Listed below are some of the pros and cons of using XER File Parser & Builder utility:

Pros of using XER File Parser:

- Free tool,
- Contains diagnostic function for troubleshooting.

Cons of using XER File Parser:

- Does not work with XER files greater than 90MB in size,
- Not compatible with Microsoft Excel 2010,
- Manual and complicated process,
- Provided by Oracle 'as is' without warranty.

### *Validate.bat Utility*

This utility supplied with P6 is a Java based schema validation tool to find missing, extra, and modified database structure descriptions in the P6 database. An incorrect setup can cause the entire P6 program to crash. Only use the Validate.bat program designed for the database version being checked. This editable text file is available in the installation media under the database\scripts\source\release folders.

Oracle Primavera recommends that users run this utility prior to conducting a P6 upgrade. Noted issues, such as missing objects, permissions, etc. must be resolved before moving forward with an upgrade of a P6 database to a later version. Location of Validate.bat, as well as instructions on how to validate P6 databases using this utility can be found in Oracle Documentation [17][18].

An HTML report of the findings is created to display the Missing, Extra, and Changed objects from the database such as displayed in Figure 3 below.

PPMDB Schema Validation Report						
<b>Missing Objects</b>						
Object Type	Object Name	Table Name	Message			
<b>Changed Objects</b>						
Object Type	Object Name	Table Name	Message	Source	Target	
<b>Extra Objects</b>						
Object Type	Object Name	Table Name	Message			
INDEX	ndx_task_code	TASK	Extra index TASK.ndx_task_code			
PROCEDURE	CHOBJOWNER		Extra procedure CHOBJOWNER			

**Figure 3: PPMDB Schema Validation Report**

It is important to repeat that the appropriate version of the utility must be used that matches the current database. For example, if P6 database was upgraded to release 8.1 or later, then the validate utility from that software release must be used as well.

Listed below are some of the pros and cons of using validate.bat utility:

Pros of using validate.bat:

- Free tool,
- Ensures smooth upgrade to the next version.

Cons of using validate.bat:

- Issues found must be researched and fixed by user,
- Does not identify corrupt data, only incorrect database structures.

### *Primavera – PRIM (MOSC)*

My Oracle Support Community (MOSC) is a collaboration and engagement tool released by Oracle that allows users to create new enhancement requests (or ideas), as well as vote on the existing ideas. All Primavera product enhancements are captured in the Primavera – PRIM community. The authors of this paper created an Idea (reference # 11492) to bring back the Check Project Integrity utility to P6 Version 8.0 and later. Detailed description on how to access Primavera – PRIM community, search or vote for an existing idea, as well as how to create new enhancement requests can be found in Oracle documentation [19].

### *Database Management Utilities*

The ultimate tool for identifying and correcting P6 database corruption issues are the built-in utilities supplied by the companies distributing the database manager software. Depending upon manufacturer, there are two different utilities: Oracle Database Homepage and Microsoft SQL Server Management Studio.

The use of these utilities should be limited to trained IT specialists. A thorough understanding of SQL commands and P6 database structure is required. While technically complicated, this is the proven method to identify and fix database corruption. Both utilities allow the user to query for errors and to insert or update the record to resolve the problem. SQL scripts provided by Oracle Primavera can be run using this same facility. This is how an IT expert would correct corruption problem.

### *3rd Party Oracle/Primavera Providers*

There are at least four 3<sup>rd</sup> party add-on software products that are able to help the P6 user to identify and perhaps repair P6 file corruption. Two of software packages are designed to help the P6 Administrator. These include,

- Administrator™ [20]. While Administrator is primarily intended to give the P6 Administrator better auditing capabilities, it does include the function to purge locked user sessions.
- PROJECT WATCH™ [21]. PROJECT WATCH is also primarily designed to view P6 operations in real time. Its reporting module has a corrupt files checking function that performs 22 Value Integrity checks on any given project.

One 3<sup>rd</sup> party software package can run the published list of P6 Check Project Integrity SQL commands, even on P6 Version 8 and later installations. Its name is,

- Baseline Checker™ [22]. In addition to performing schedule quality checks, this module can optionally run the P6 Check Project Integrity SQL checks and report back on the results. The program does not attempt to correct the errors found.

Finally, one 3<sup>rd</sup> party software package is designed solely to research out instances of P6 database corruption. Its name is,

- Corrupt Files Checker™ [23]. Corrupt Files Checker works by investigating the database one project (schedule) at a time. The entire database can be included in one report. It makes 20 different Field Integrity checks and 68 Value Integrity checks for each project selected. The statistical analysis of database corruptions found in large P6 production systems for this paper was made using the Corrupt Files Checker program.

## Evaluation: How Widespread is P6 Database Corruption?

In order to quantify how wide spread P6 corruption issue is, the authors test three of the suggested options listed in the previous section on four production P6 databases. These production databases are typical of most large Construction Management firms and are maintained by a trained and competent IT staff. Various platforms and versions of P6 are included.

The three methods employed to test for database corruption included eForensic Corrupt Files Checker, P6 Check Project Integrity, and the P6 Validate.bat Schema Checking utility. When the P6 version being tested does not include a built-in Check Project Integrity function, the SQL commands are manually executed by an IT expert. Data Value checks could not be executed for the last two databases due to lack of program availability in those P6 versions. The results are detailed in Figure 4 as follows:

Database	Unique Characteristics						eForensic Corrupt Files			Check Project Integrity		Schema Validation
	Single User	Multi User	3rd Party Access	# of Projects	Average # of Tasks	Resource /Cost Loaded	# of Corrupted Projects	% of Corrupted Projects	# of Errors in each Project	Data Join Errors	Data Value Errors <sup>1</sup>	
#1	x			123	10	x	1	1%	1	0	40 <sup>2</sup>	Clean
#2		x	x	235	759	x	36	15%	1	2 <sup>3</sup>	0	Clean
#3		x		593	1170		195	33%	9	1 <sup>4</sup>	NA	Extra Object
#4		x		50	10346	x	45	90%	19	1 <sup>5</sup>	NA	Clean
1. Number of projects effected. Check is not available (NA) for database v8.x and greater												
2. All Data Value Errors found are related to resource assignments (checks #17, 18, 25, 26, 46)												
3. Two (2) Data Join errors found are generated by checks #15 and #16 (Project mis-match between Calendars & Activities, and Project mis-match between Activity Code Assignments & Activity Codes)												
4. One (1) Data Join error found is generated by check #5 (WBS mis-match between document assignment & activities)												
5. One (1) Data Join error found is generated by check #18 (Project mis-match between Projects & WBS)												

**Figure 4: P6 Database Corruption Evaluation Results**

The result of this test reveals that even the most straight forward, compact schedules created in P6 via pre-defined templates (database #1) are not immune to data corruption. As the average number of activities in project grows, so does the number of errors (database #4).

The most common corruption across all databases is found to be excessive size of the POBS table, as well as blank (NULL) values in relationship lag fields.

## Conclusion

This paper describes the nature of what is Oracle Primavera P6 database corruption. It discusses major causes of database data corruption, the way to identify it and use available utilities to troubleshoot, repair or apply workarounds. It also attempts to help P6 users to avoid apparent errors that only look like instances of P6 database corruption.

Various tools available to P6 users to investigate and repair corruption problems are identified. One primary tool is the now-missing P6 Check Project Integrity utility. Oracle Primavera needs to return this utility to the administrator's toolbox, as well as to improve it to remove instances of false positives (over-reporting) in the report.

IT personnel overseeing P6 databases need training and guidance in preventing, spotting, and correcting database corruption issues. They should consider 3<sup>rd</sup> Party training or just direct support from qualified Oracle Primavera Providers to deal with this important issue.

P6 database corruption is a real and recognizable issue when evaluating the quality of a P6 schedule. When 90% of the projects in one production database have been found to have corruption problems, one cannot blindly assume that any given P6 schedule is whole and intact. An understanding of this issue is essential for both the IT expert, as well as the schedule analyst.

## References

1. Oracle Knowledgebase. "Error: 'The user is already logged in. Please use another login name' When Logging into Project Management or Methodology Management" (Doc ID 891328.1)
2. Oracle Knowledgebase, "Differences Between XML and XER Import / Export in Project Management" (Doc ID 907653.1)
3. Oracle Knowledgebase, "POBS Table Has Grown in Size and Affects Performance With Project Import/Export" (Doc ID 1967569.1)
4. "CLEANING UP A POBS TABLE", D.R. McNatty, November 2015 Tech Tip, <http://www.drmcnatty.com/wordpress/wp-content/uploads/2015/11/November-Cleaning-upPOBS-tables.pdf>
5. SAe Purge POBS, Schedule Analyzer Software, Ron Winter Consulting LLC, [http://scheduleanalyzer.com/sae\\_p\\_POBS.htm](http://scheduleanalyzer.com/sae_p_POBS.htm)
6. Emerald Tools, Emerald Associates, <https://www.emerald-associates.com/software/emerald-tools/emerald-tools.html>
7. XER Schedule Toolkit, <https://www.xertoolkit.co.uk/>
8. TENSIX Consulting, 'Is Your P6 Activity Layout Really Corrupt?' by TheP6Pro, <https://tensix.com/2014/08/is-your-p6-activity-layout-really-corrupt/>
9. Weird Negative Float, <http://www.primaverascheduling.com/weird-negative-float/>
10. Planning Planet blog, <http://www.planningplanet.com/forums/oracle-primavera-pm6/579478/level-effort-spurious-actuals>
11. Planning Planet blog, <http://www.planningplanet.com/forums/primavera-version-pm5-pm6/420465/od-rd-equal-zero-when-converting-p6-p3>
12. Planning Planet blog, <http://www.planningplanet.com/forums/oracle-primavera-pm6/579724/notprmuser-displaying-p6>
13. TENSIX Consulting, 'Is Your P6 Activity Layout Really Corrupt?' by TheP6Pro, <https://tensix.com/2014/08/is-your-p6-activity-layout-really-corrupt/>
14. Oracle Knowledgebase, "Check Project Integrity Utility: Description of Utility Including Detailed SQL Statements Executed" (Doc ID 1083481.1)

15. Oracle Knowledgebase, “How To Include Project Integrity Checking In A Database Maintenance Plan” (Doc ID 1315895.1)
16. Oracle Knowledgebase, “How to Open an XER file in Excel to see All the Data Lined Up with the Column Headings” (Doc ID 902053.1)
17. Oracle Knowledgebase, “How to Validate a Project Management / Methodology Management database Schema – Versions 6.2 through 8.0” (Doc ID 911561.1)
18. Oracle Knowledgebase. “How To: Validate a P6 EPPM or P6 PPM Database Schema Using 8.1 and Later Releases” (Doc ID 1342977.1)
19. Oracle Knowledgebase, “Oracle Primavera Product Enhancement Request to My Oracle Support Community FAQ” (Doc ID 1995605.1)
20. XENA Administrator, <http://www.xenautilities.com/terry-schaefer/new-xena-administrator-v1200-now-available>
21. PROJECT WATCH by Ron Winter Consulting LLC, [http://scheduleanalyzer.com/pwatch\\_brochure.htm](http://scheduleanalyzer.com/pwatch_brochure.htm)
22. Baseline Checker module of Schedule Analyzer for the Enterprise by Ron Winter Consulting LLC, [http://scheduleanalyzer.com/sa\\_baseline.htm](http://scheduleanalyzer.com/sa_baseline.htm)
23. Corrupt Files Checker module of Schedule Analyzer Enterprise Forensic by Ron Winter Consulting LLC, [http://scheduleanalyzer.com/eforensic\\_brochure.htm](http://scheduleanalyzer.com/eforensic_brochure.htm)