

## PeopleTools and Trace Facilities

PeopleSoft architecture actually includes robust tracing facilities and PeopleTools traces are the first line of defense for any performance issues and problems. We will review the options and usage in this document.

### PeopleSoft online processing

In online processing, we have to deal with SQL, Peoplecode and Remote call traces. There are other trace options such as component processor and panel processor buffers but we do not normally use those trace options unless we are trying to debug an issue. We most commonly use the traces to diagnose a performance bottleneck.

You can make the system to perform traces either of the two ways.

- 1.0 Modify psappsrv.cfg, thus enabling globally
- 2.0 Configure enableTrace = true in configuration.properties in the PIA. This will facilitate tracing for just that user who wants to trace.

If you choose to enable tracing at the application server, be aware that every active user in the system will produce traces. I do not recommend you to do so. However while working with Sign-on peoplecode, for example, LDAP authentication, some of the SQL statements will not be captured unless you enable the tracing at the application server level.

### Modifying PSAPPSRV.CFG

In order to set the trace at the application server level, you will have to modify the parameter TraceSQL. When you set the TraceSQL parameter to a value higher than 1, all users / clients communicating with the application server will produce a SQL trace. You can control the granularity of the trace output by choosing an appropriate value for TraceSQL parameter. The application server components use the value to decode in a bit wise manner and produce the desired trace results. A listing of various bits can be found in the psappsrv.cfg file. Typically a value of 7 is far more than sufficient to diagnose a SQL performance issue. A value of 32 is useful in diagnosis and is also most commonly sought from PeopleSoft customer support analysts. I typically choose 7 unless additional information is necessary.

Similarly in order to trace Peoplecode, the parameter TracePeoplecode is used. I typically choose 4072. This option helps me to understand the Peoplecode logic more clearly. The trace options are dynamic and you do not need to restart the application server domain to do the tracing.

Since the application server level trace option produces trace for every active client in the system, it is generally not recommended.

### Trace option in the Web server.

As said before, you can also configure the PIA in such a way that the users can themselves choose to enable the traces.

The property enableTrace = (true or false) in the configuration.properties file of the PIA domain on the web server is used to control this behavior. When this parameter is set, the sign on screen presents a hyperlink where the users can choose to select appropriate trace options and granularity. The parameter TraceSQLMask in the application server psappsrv.cfg controls the

maximum granularity level for the trace output. Usually it is set to the maximum value, 12319 so that the users can select all trace options. But if the TraceSqlMask is set to 7, then the users can select only the first three SQL trace options. Even though they are able to select additional trace options, the application server limits the trace output to 7 (SQL statements, variables and connect / disconnect / rollback).

Similarly they can choose to set appropriate Peoplecode traces themselves. You have to restart the web server after a change to the configuration.properties in 8.4x Peopletools releases. However the change is dynamic in 8.1x Peopletools releases.

For a production system, it will be very useful if you set the enableTrace option to true in the configuration.properties and a TraceSqlMask of zero in the psappsrv.cfg. When a user wants to set a trace, then enable the TraceSqlMask in psappsrv.cfg, so that the tracing can be completed. This enables you to continue the operation without having to restart any servers.

### **Tracing Remote call COBOL processes**

Remote call COBOL processes are batch COBOL programs, which when invoked from a Peoplecode executing in the application server, run in the application server machine but communicate with the database for SQL requests. Note that Application Engine process can also invoke Remote call but in this case the Remote called COBOL program will execute in the machine where the application engine is executing. It is likely to be the Process Scheduler environment.

In order to perform a Remote Call trace, set the Remote Call redirect flag in the psappsrv.cfg to true. Choose the TraceSql option 128 (COBOL statement timings). Again it is recommended to choose a trace value of 131.

### **Tracing within a PORTAL environment**

Once you import the PIA content into the PORTAL, the ability to choose the trace options from the portal sign-on page severely restricts your ability to trace. Any SQL trace captured will not reflect the actual transaction performed at the PIA site. The only option in such a situation is to use the trace option within the application server. Hence in a typical PROD environment it is a good practice to deploy two additional web environments just for tracing as we recommended before for the application server. These additional services do not pose extra overhead they will be down for most of the time. You only need to bring those services when you want to capture a trace to diagnose an issue.