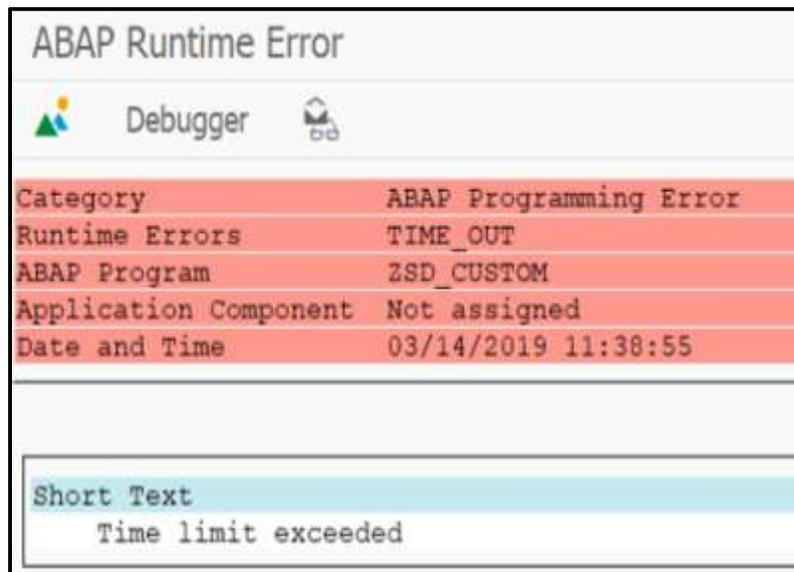


Fixing Time-Outs and Long-Running ABAP Reports

The dreaded “TIME_OUT” error message – we’ve all been there.



Reports that used to take seconds are now taking minutes, and reports that used to chug away for a while and generate a much-anticipated month-end report instead return the dreaded “ABAP Runtime Error”. It’s a common scenario for many SAP customers.

Slow-running ABAP reports or time-outs are a common source of frustration for SAP users. More often than not, the culprit is a custom-written ABAP program – the dead giveaway is the “Z” at the start of the program name.

Some common causes of long-running programs:

- The ABAP program was not properly optimized, and while it may run quickly at first it slows down over the years due to increase in volume of data.
- A report started out simple, but over the years modification after modification was performed to create a “Frankenstein” report that strayed far from the purpose of what the original was designed to do.
- Reports designed years ago may have been coded without optimization functionality that is available in more recent versions of SAP. Optimizations are often possible after SAP upgrades.

One way to prevent a report from timing out is to ask your Basis team to modify system parameters allowing for longer run times. This might temporarily get the report to run again, but it's not a permanent solution. We don't recommend it, as the report is sure to time-out again as the database grows, not to mention the possible degradation to system performance.

Solutions

Fixing a performance issue with a report will require some ABAP expertise, and the solution is highly dependent on the underlying cause and specific logic of the program, however here are some tips and techniques that can be applied to fix a majority of problems:

- Analyze the problem using SE30. This standard SAP transaction will do an analysis of the report performance and show you where the bottlenecks are.
- The “Extended Program Check” and “Code Inspector” in SE38 can also help identify issues with optimization.
- In the vast majority of cases (practically all the time in our experience), the main issue is database access and retrieval of information. Optimize the database retrieval methods, staying away from nested SELECT statements.
- Consider splitting Frankenstein reports into multiple separate reports. The removal of that column or additional subtotal that is only used by one person might be dragging down performance.
- Investigate the creation of new database indexes. This can result in significant performance improvements in report runtimes, however there is a trade-off with increased overhead.
- Consider a complete rewrite. As unappealing as it sounds, sometimes it actually takes less time to start again from scratch rather than to decipher the pages and pages of twisted logic in legacy reports that have been modified by generations of ABAP'ers who are no longer with the company.
- Evaluate whether you are using the right tool for the job. Maybe that ABAP report really should be a BW cube, or generated through a data analytics tool such as SAP Cloud Analytics.