

AST (Asynchronous System Trap)

What is an *Asynchronous System Trap*? This is similar to Unixes signals or OS interrupts. Technically this is callbacks from the Lock Manager into the Engine, called when somebody acquires a lock at a level not compatible with the current level. In this case all processes who own the lock receive notification and should take appropriate action to release the lock (immediately or deferred-ASAP).

The main trick with AST's handlers is that AST is sent asynchronously and the handler must take care to work correctly. Since v.2.5 all ATS handlers run in a separate thread. Before v.2.5 it works using signals on Unixes and threads on Windows.

P.S. the issue with old `glibc` and Firebird 2.5 is that since v.2.5 we abandoned the usage of old `System V semaphores` (do you remember semaphores are exhausted error? now you can forget it completely) and started to use `pthread` mutexes. And old `glibc` have bugs in the pthread mutex implementation.

Source: <https://www.firebirdnews.org>

From:
<https://www.ibexpert.com/docu/> - **IBExpert**

Permanent link:
<https://www.ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:asyncronous-system-trap>

Last update: **2023/08/13 19:32**

