

## Runtime Infrastructure - Bug #7377

### ServerDriver.connect() should not force server-mode to be turned on

05/23/2023 06:27 AM - Greg Shah

<b>Status:</b>	New	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>case_num:</b>	
<b>billable:</b>	No		
<b>vendor_id:</b>	GCD		
<b>Description</b>			

#### History

##### #1 - 05/23/2023 06:30 AM - Galya B

Check [#5703#note-610](#).

##### #2 - 05/23/2023 06:31 AM - Greg Shah

As discussed in [#5703-608](#) and previous notes, the `ServerDriver.connect()` will call `bc.setServer(true)` and try to load the directory when no keystore is provided. The problem here is that in any mode other than `MODE_START` this is **unsafe** since the directory is potentially already open and being edited in a running server. We must not try to read or process that XML from a separate JVM. I guess this was added as an alternate approach so that the server's keystore does not need to be in the file system. I don't think it is a safe approach.

Here is the code at issue:

```
if (bc.getString("security", "keystore", "filename", null) == null)
{
    // if no keystore is specified, then we need to initialize as "server" and load it
    // from the directory.
    bc.setServer(true);
    bc.setConfigItem("directory", "xml", "must_exist", "true");
    DirectoryService ds = DirectoryService.createInstance(bc);
    if (!ds.bind())
    {
        LOG.warning("Could not bind directory!");
    }

    SecurityManager sm = SecurityManager.createInstance(bc);

    if (bc.getString("security", "truststore", "alias", null) == null)
    {
        // explicitly set the trust alias as the server's alias
        String alias = sm.getServerAlias();
        bc.setConfigItem("security", "truststore", "alias", alias);
    }
    // force loading of the transport security
    sm.getSecureSocketContext();
    // go back in non-server mode
    bc.setServer(false);
}
```