# Table of Contents

Simple integration of multiple databases is an easy task with JVx. Configuration is usually completed via the application's config.xml file. This allows for easy access in the life cycle objects:

[config.xml](config.xml)

```xml
<?xml version="1.0" encoding="UTF-8"?>

<application>
  <securitymanager>
    <class>com.sibvisions.rad.server.security.DBSecurityManager</class>
    <database datasource="mydb" />
  </securitymanager>

  <datasource>
    <db name="mydb">
      <url>jdbc:oracle:thin:@localhost:1521:mydb</url>
      <username>user</username>
      <password>password</password>
    </db>
    <db name="masterdb">
      <url>jdbc:derby://localhost:1527/masterdb</url>
      <username>master</username>
      <password>master</password>
    </db>
  </datasource>
</application>
```

To access:

```java
IConfiguration config = SessionContext.getCurrentSessionConfig();

dba = DBAccess.getDBAccess(DBSecurityManager.getCredentials(config));
dba.open();
```

using the data source of the SecurityManager. To access data sources independently from the security Manager we use DataSourceHandler:

```java
IConfiguration config = SessionContext.getCurrentSessionConfig();

DBCredentials cred = DataSourceHandler.createDBCredentials(config,
"masterdb");

dba = DBAccess.getDBAccess(cred);
dba.open();
```

# Note

The [DB Security Manager](DB Security Manager) allows the use of a standard data source without any additional

configuration effort:

config.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>

<application>
  <securitymanager>
    <class>com.sibvisions.rad.server.security.DBSecurityManager</class>
  </securitymanager>

  <datasource>
    <db name="default">
      <url>jdbc:oracle:thin:@localhost:1521:mydb</url>
      <username>user</username>
      <password>password</password>
    </db>
  </datasource>
</application>
```

The security manager uses the data source named "default" as the standard connection.