Did you know you can encrypt the communication with the isCOBOL Server?

isCOBOL Server can be configured to encrypt the data transmitted on the TCP/IP connection. isCOBOL Server relies on JSSE (JavaTM Secure Socket Extension) for communication. In the Sun/Oracle version you need to also get the JCE (JavaTM Cryptography Extension) in order to get unlimited strength cryptography. In the JSSE specification, certificates are stored in a file called keystore. Thus you need to have a keystore with a key entry (with both private and public keys) on the server side, and a trusted certificate entry on the client side. JavaTM supports the JKS (JavaTM KeyStore) format, and it may contain both key entries and trusted certificate entries. In order to handle this file format, the command line program keytool is provided with the standard JDK distribution. The keytool program is located in the bin directory under the JavaTM Home. For the sake of simplicity let's assume that we can invoke keytool supplying only the name. To create a new keystore from scratch, containing a single self-signed Certificate, execute the following from a terminal command line:

```
keytool -genkeypair -alias iscobol -keyalg RSA -keystore myKeystore
```

After executing this command, you will first be prompted for the keystore password. You can choose any password you like, at least 6 characters long. Then you will be asked about general information on this Certificate, such as company, contact name, and so on. This information will be displayed to users who attempt to access a secure page in your application, so make sure that the information provided here matches what they will expect. Finally, you will be prompted for the key password, which is the Certificate specific password (as opposed to any other Certificates stored in the same keystore file). The keytool prompt will tell you that pressing the ENTER key automatically uses the same password for the key as the keystore. The JSSE framework, and isCOBOL by consequence, requires these passwords to be identical. If everything was successful, you now have a new file, named myKeystore under the working directory. Now you can establish a secure connection between client and server as follows. In the server configuration file set:

```
iscobol.net.ssl.key_store=/path/to/myKeystoreiscobol.net.ssl.key_store_password={choose
n-password}
```

In the client configuration file set:

```
iscobol.net.ssl.trust_store=/path/to/myKeystoreiscobol.net.ssl.trust_store_password={ch
oosen-password}
```

Online URL: https://support.veryant.com/phpkb/article.php?id=282