Guide to update the isCOBOL development environment

This guide has four sections, the first two discuss how to update to a later build or release of the IDE, and how to update to a later isCOBOL SDK. The last two sections can be followed when you just need to take advantage of a compiler enhancement that you updated on the SDK but can be leveraged by the IDE without having to update the IDE as well or when you want to update the IDE to use any enhancement on it but still use and older compiler version from the SDK.

If your update is for a later year release you'll need to request development licenses for your products for that year and update the properties file where you have your license keys before taking the steps below. To learn more about that you can review the video about licenses on Veryant's YouTube channel: https://www.youtube.com/watch?v=R6ScVpdT1XM

1. Updating the isCOBOL IDE

If you are updating to a later build of the same IDE release, then you can just install in the same folder and continue using the IDE without further steps.
Example: Updating the IDE 2022.1.1060.5 to the 2022.1.1060.7.

If you are updating from a prior release to a later one, especially from a prior year release, you may want to remain with the prior version of the IDE during the transition while you test that the new IDE is working without errors with your applications. This is easily done because the IDE is installed in different folders and they are very self-contained so that they do not interfere with each other.

When installing a different year IDE version, it is a good practice to create a clean new workspace, start the new IDE in that clean workspace and import the projects from the older one.
Example: If you had the IDE 2020R2 and you now install the IDE 2022R1, you might create a new workspace called "workspace22r1". This option is useful to give you an independent workspace for each version of the IDE while you run your general tests during the transition.

Another option, especially useful if you update the IDE every year, is to continue using the same workspace. In this case you should keep a backup of it before accessing the workspace to use with the later IDE.

2. Updating the isCOBOL SDK

If you are installing a later build of the same release, install in the same folder. You do not need further steps to continue working.

If you are installing a different release, the default installation will install in a different folder by default. This will allow you to keep the older version for possible parallel testing or usage during

the version transition.

After installing the new release while keeping the prior, you need to make sure you are accessing the correct version of the SDK. The easiest way to do this in Windows is to use the isCOBOL Shell command prompt, found in the Start menu. You can start the isCOBOL Shell for the specific version you want to run to ensure you are accessing the correct version of the SDK. You can test to see which version you're using by running the command "iscrun -version".

If your SDK was installed on Linux and you use shell scripts to run commands, or you use batch scripts in Windows instead of using the isCOBOL Shell command prompt, you can add SET/EXPORT commands in those scripts to point to the correct version you want.

For instance (example for 2022R1 and JDK 8u321):
Windows

```
  set ISCOBOL=C:VeryantisCOBOL_SDK2022R1   set ISCOBOL_JDK_ROOT=C:Program FilesJavajdk
1.8.0_321   set CLASSPATH=%ISCOBOL%lib*;%ISCOBOL%jars*;%ISCOBOL_JDK_ROOT%lib ools.jar;%
CLASSPATH%   PATH=%ISCOBOL% in;%ISCOBOL_JDK_ROOT in;%PATH%
```

Linux

```
  export ISCOBOL=/opt/veryant/isCOBOL_SDK2022R1   export ISCOBOL_JDK_ROOT=/opt/jdk1.8.
0_321   export CLASSPATH=$ISCOBOL/lib/*:$ISCOBOL/jars/*:$ISCOBOL_JDK_ROOT/lib/tools.jar
:$CLASSPATH   export PATH=$ISCOBOL/bin:$ISCOBOL_JDK_ROOT/bin:$PATH
```

In Windows environments you can also set these environment variables at the computer level in your Windows system properties when you discard your older SDK version and remain with the latest you installed.

3. Updating the SDK compiler and use it in your IDE

In this scenario you are advised that a compiler enhancement was made and you update the isCOBOL SDK with that, but you also want the IDE to take advantage of it. After updating the isCOBOL SDK as described above in section 2 you do not need to update the IDE as well, just configure the IDE to use the external SDK compiler.

To do that use the Window > Preferences > isCOBOL > Installed isCOBOL option. There you will find the "Embedded isCOBOL" libraries.
Click the Add button and browse to the folder where the isCOBOL SDK is installed (the one that contains the lib and bin subfolders) and select it.
Then click the check-box of that new entry. That will unselect the default "Embedded isCOBOL" and select the SDK that was updated, so the compiler of that installation will be used by the IDE from then on.

4. Updating the IDE and have it use a prior SDK compiler

The last scenario is when a later IDE release has a development enhancement that you want to use, but you need to still use an older compiler from the SDK. This may happen for instance if your production environment still remains on a prior release.

In this case you can update the IDE as described above in section 1, then use the Window > Preferences > isCOBOL > Installed isCOBOL option described above in section 3 to point to the SDK compiler of your prior version.

Online URL: https://support.veryant.com/phpkb/article.php?id=328