Hi all,

Yes, you can do this. However, here are some caveats:
1. This is not officially supported by ITTVIS. No whining allowed.
2. We have only done limited testing. We have an excellent test group,
but there was only so much time for testing IDL7.0.

There are two ways you can do this:

A. If you have just a few, small languages, like Python, Perl, etc,
then you can *probably* (emphasis already added) just drop these
plugins into your IDL70/idlde/plugins directory, and everything
should (more emphasis) work fine. Stop reading here!

B. For a much larger feature, like the Eclipse Java Development Tools
(JDT), then you probably want to go the other route and drop IDL into
your existing Eclipse install. To do this:

1. Find the three core IDL plugins in your IDL70/idlde/plugins
directory:
    com.rsi.idldt_7.0.0.jar
    com.rsi.idldt.core_7.0.1.jar
    com.rsi.idldt.ui_7.0.0.jar
2. Copy these JAR files into your Eclipse/plugins directory.
3. If you want the IDL documentation, then you'll need to copy all of
the com.rsi.idl.doc.*.jar plugins as well.
4. If you want the language packs, you'll need to copy the ones with
"*.nl*.jar" as well.
5. Create a new shortcut to your "eclipse.exe", with the following
command-line options:

C:\Eclipse3.3\eclipse.exe -vmargs -DDMLDir="C:\Program Files\ITT
\IDL70\idlde" -DJDMLDir="C:\Program Files\ITT\IDL70\idlde" -DBMLDir="C:
\Program Files\ITT\IDL70\idlde" -DIDLDir="C:\Program Files\ITT
\IDL70\bin\bin.x86" -DIDL_DIR="C:\Program Files\ITT\IDL70\"

You will obviously need to change the above paths if you are on Unix,
or have installed IDL in a different directory.

6. Start up Eclipse. The first time you start, you will not see the
IDL perspective, and the IDL process won't be started.
7. Go under Window->Open Perspective->Other, and choose IDL. You will
get a bunch of Java null exception errors, and IDL will attempt to
start but fail.

8. Quit Eclipse and restart. At this point, you should be "good to go".

Things I have noticed:
 * I always get a Java null pointer exception when I start the "IDL FrankenDE". This seems harmless.
 * To get the IDL process to start, you must have the IDL Command Line visible.
 * You cannot seamlessly debug a Java program and an IDL program - for example, if you are using the Java export bridge, this will fire up an IDL process that is not the same as the IDL process within your Workbench - so you can't step from your Java code into your IDL code.

Good luck! If you try this, be sure to post to the newsgroup with any tips or feedback.

-Chris
ITTVIS