

How-to generate EXPRESS-G with JSDAI

Be Curious!



Take it easy

Acknowledgements

A huge thank you to ***Pierre Duchier***, Airbus, for the time he devoted to showing us the method, reviewing the document, and answering our questions.

Also thanks you to LKSoft to provide the JSDAI software.

Learning Objectives

- Install Eclipse Neon
- Install JSDAI
- Create projects
- Compile EXPRESS
- Edit existing EXPRESS-G diagram
- Generate new EXPRESS-G diagram
- Export EXPRESS-G components



Resources

<https://committee.iso.org/sites/tc184sc4/home/quality-committee.html>

- Git wg12-step repository
<https://sd.iso.org/bitbucket-pilot/projects/ISOTC184SC4/repos/wg12-step/browse>
- .exl files and EXPRESS-G diagrams output from JSDAI repository
https://sd.iso.org/bitbucket-pilot/projects/ISOTC184SC4/repos/jsdai_eclipse/browse
- ISO 10303-11
- <https://jsdai.net>

NOTE A slide describing how to use this repository will be added soon.

Ask ChatGPT: Why use JSDAI - <https://jsdai.net>

Standards Compliance

Fully compliant with ISO 10303 (STEP) and EXPRESS standards.

Java Integration

Provides Java classes generated from EXPRESS schemas, making it easy to integrate with Java applications.

Data Interchange

Allows reading, writing, and validating STEP files (ISO 10303) efficiently.

Schema Flexibility

Supports multiple EXPRESS schemas, including custom and extended schemas.

Diagram Support

Facilitates the creation and management of EXPRESS-G diagrams.

Error Checking

Automatically validates data against EXPRESS constraints, reducing runtime errors.

Open Source

Free and open-source, allowing customization and integration in different projects.

Extensibility

Can be extended to support additional file formats, database backends, or custom rules.

Tool Integration

Compatible with IDEs like Eclipse for schema compilation and diagram visualization.

Cross-Platform

Runs on any platform that supports Java, making it highly portable.

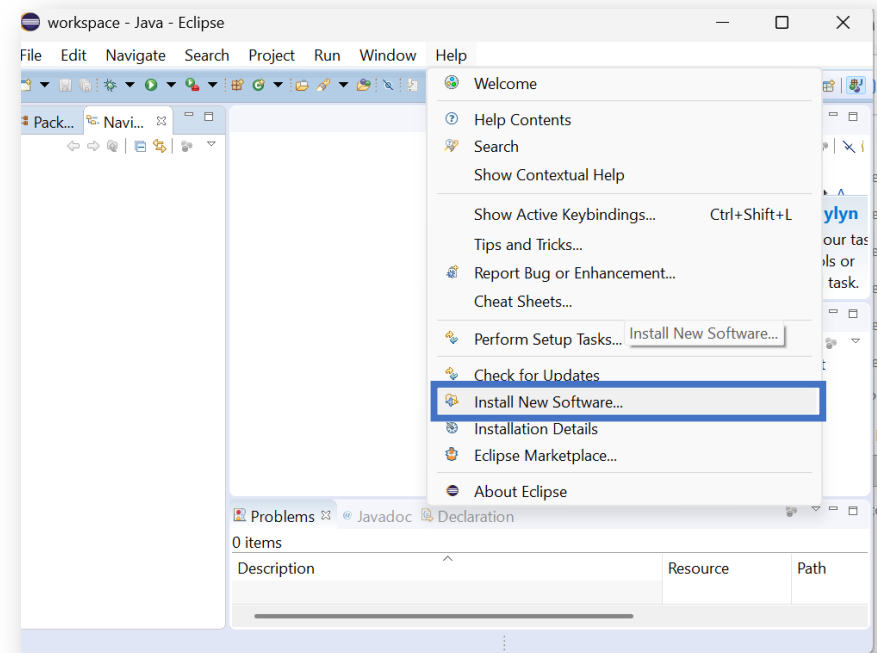
Install Eclipse Neon and JSDAI (1/2)

Download Eclipse

- Go to <https://eclipseide.org>, click “**Other Releases**”, then select “**Neon R**”.
- The team has tested “**Eclipse IDE for Java EE Developers**” and “**Eclipse IDE for Java Developers**”.
- Download the package, extract all and run **Eclipse**.

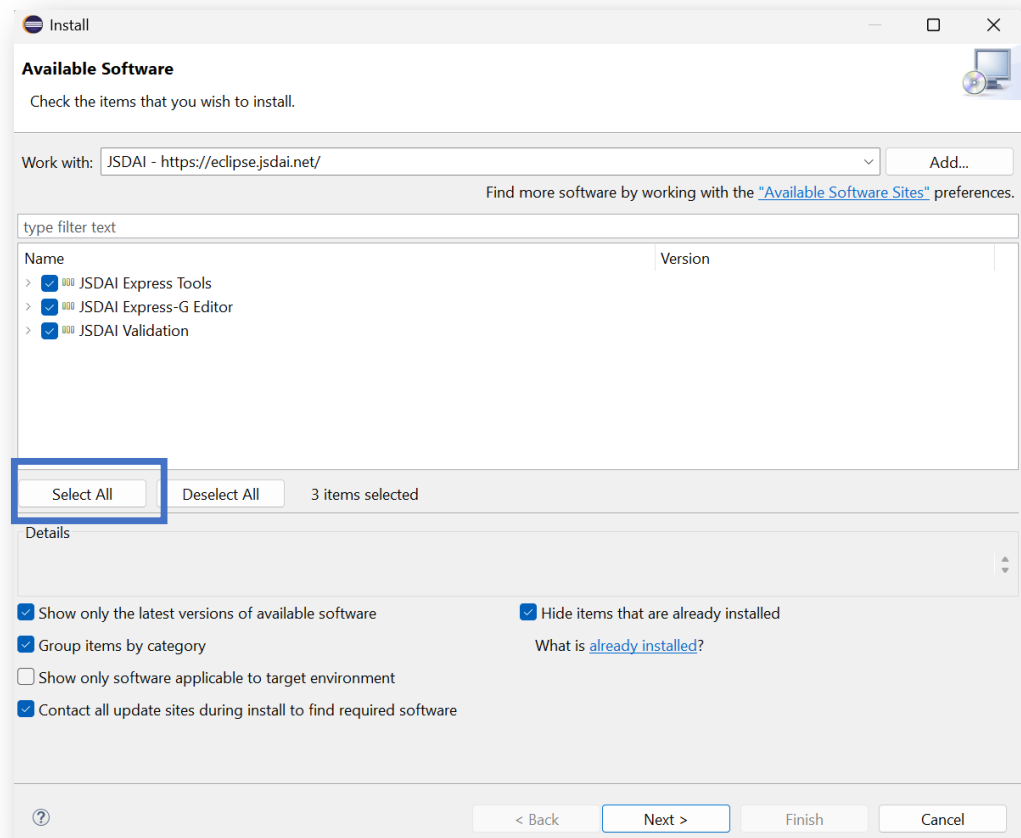
JSDAI

- In the “**Help**” menu of Eclipse, choose “**Install new software**”.



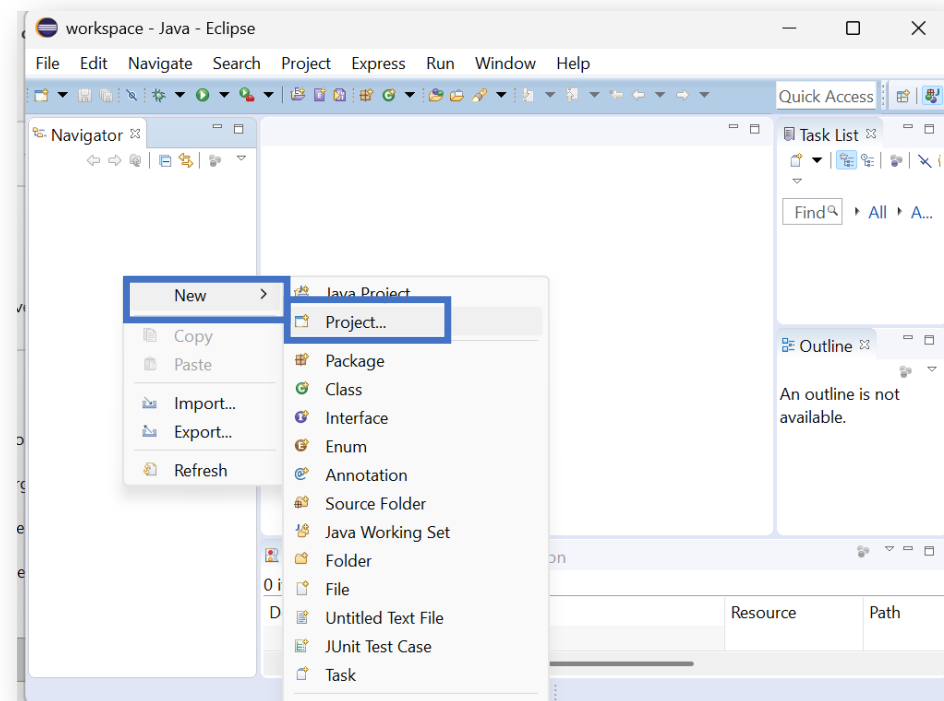
Install Eclipse Neon and JSDAI (2/2)

- Click **"Add"** and enter a name (example: JSDAI), in the location, enter <https://eclipse.jsdai.net> and click **"OK"**.
- Then click **"Select All"**, **"Next"**, **"Next"**, Accept the terms of the licence agreement and click **"Finish"**.
- After the installation is complete, you may need to restart Eclipse.



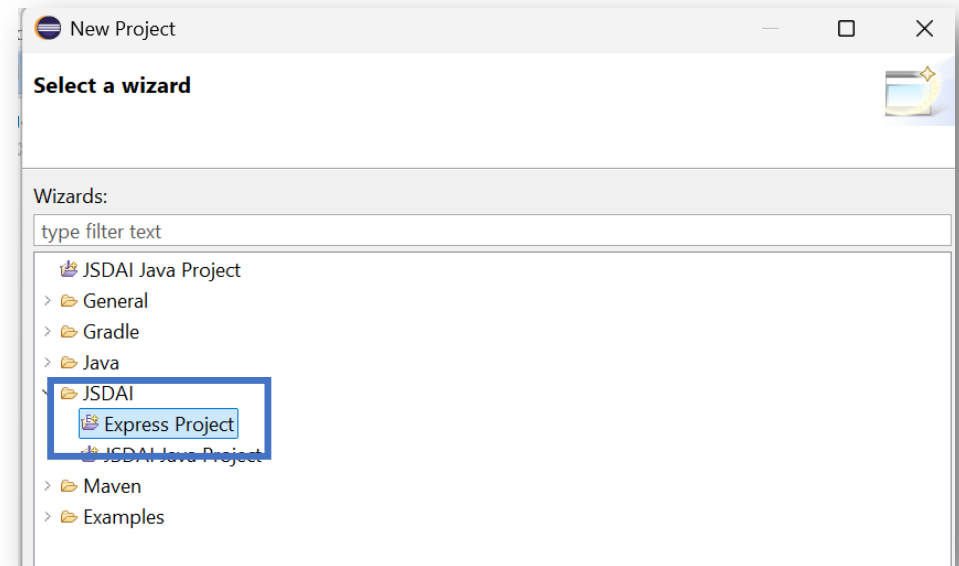
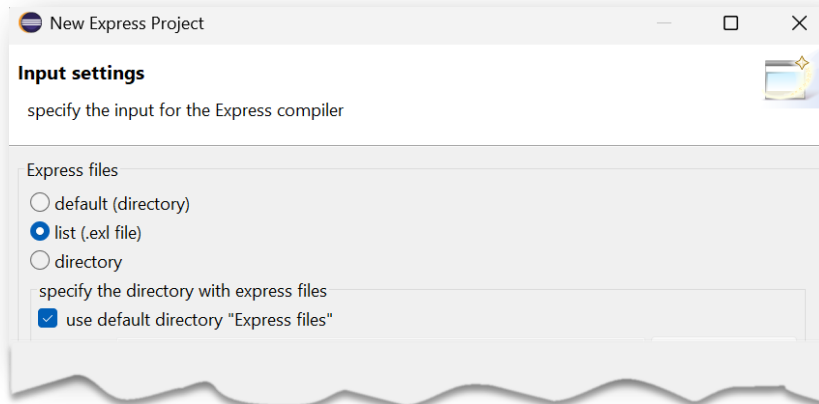
Create projects in the Eclipse workspace (1/3)

1. Back to Eclipse, change the Package Explorer view to the "**Navigator**" view
 - In the "**Window**" menu, select "**Show view**" and then click "**Navigator**".
2. In the Navigator pane, right click and select "**New**" then "**Project**"



Create projects in the Eclipse workspace (2/3)

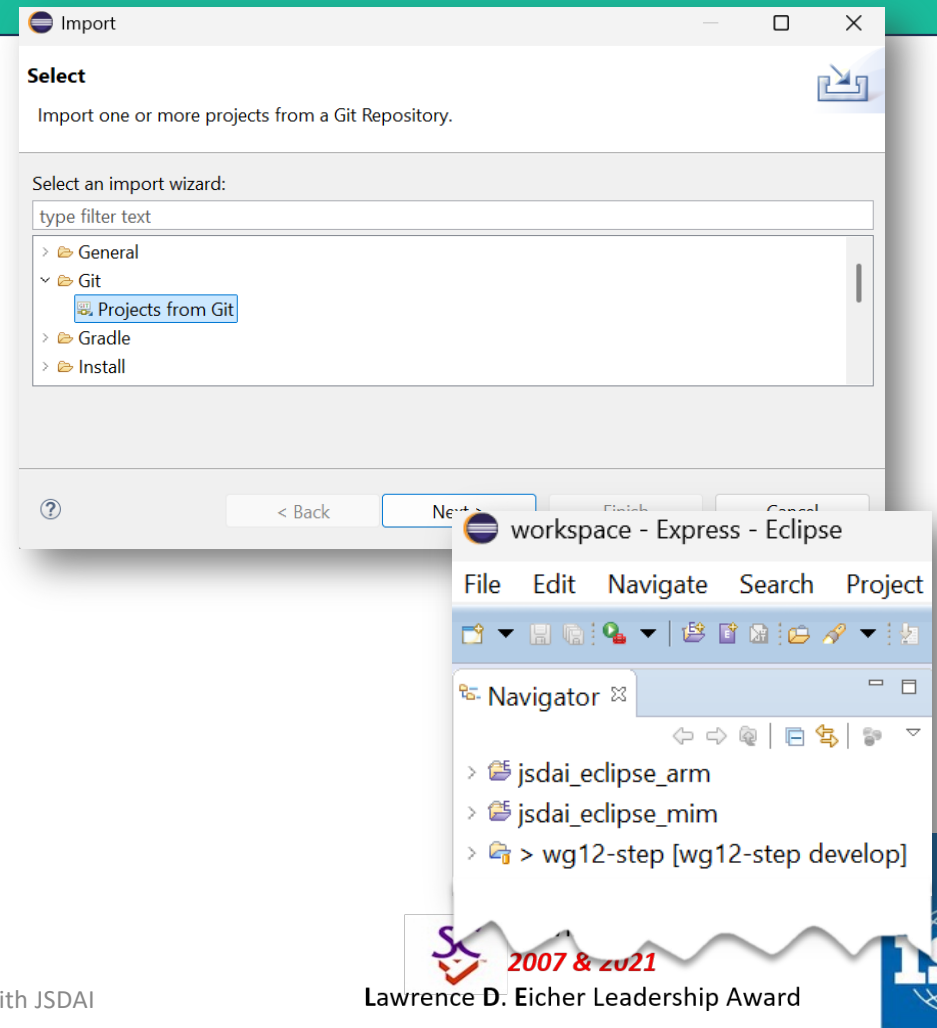
3. Unfold the "**JSDAI**" folder and select "**Express Project**", click "**Next**", enter "**jsdai_eclipse_arm**" as Project name.
4. click "Next" and set the "**Express files**" to "**list (.exl file)**" and click "**Finish**".



Create a second project following 2 & 3 with "**jsdai_eclipse_mim**" as Project name.

Create projects in the Eclipse workspace (3/3)

5. Clone the wg12-step repository from the **Eclipse workspace**. **!IMPORTANT**
6. In the “**Navigator**” pane, right click and select “**Import**”, unfold the “**Git**” folder and select “**Projects from Git**”, click “**Next**”, select “**Existing local repository**”, click “**Next**”, then click “**Add**” and browse your computer to select “**wg12-step**” repository.
7. Click “**OK**”, select the proposed .git and enter “**Finish**”. Eclipse resolves the repository name, click “**Next**”, click import as “**existing eclipse project**”, click “**Next**”, and “**Finish**”.

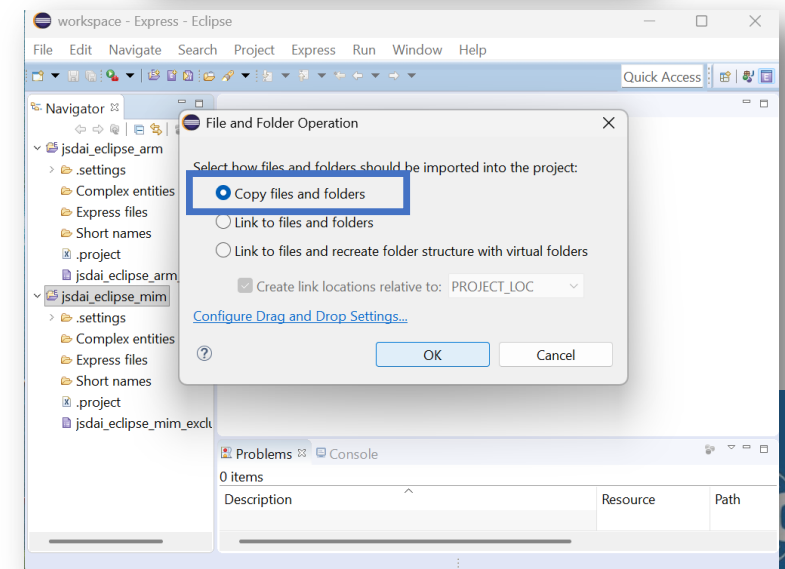
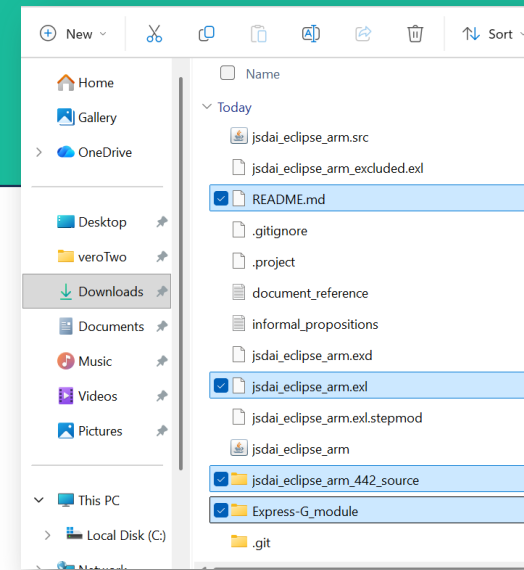


EXPRESS-G package

We will not start from a blank page, but from previous diagrams.

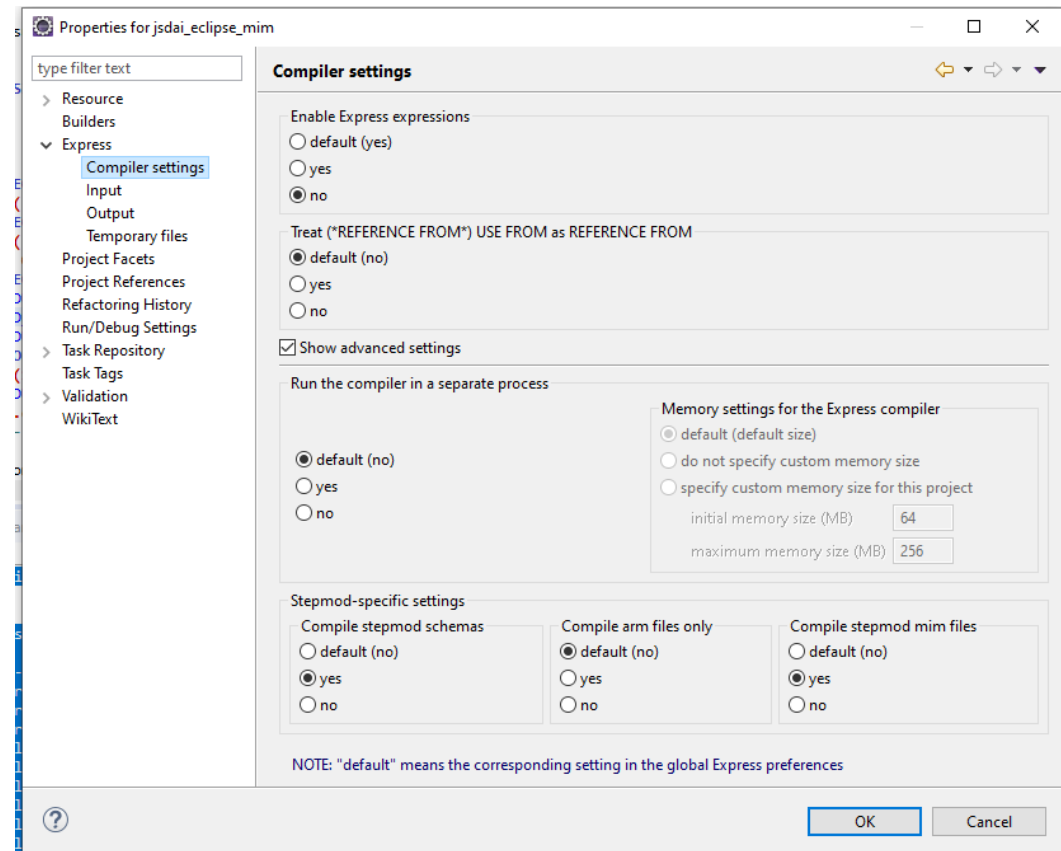
EXPRESS-G Package

- Download the package located at <https://nextcloud.boost-lab.net/nextcloud/index.php/apps/files/files/191535?dir=/wg12-step/JSDAI&openfile=true>
Extract all in a temporary folder, not in the Eclipse workspace.
- From the jsdai_eclipse_arm folder, select the following:
 - README.md
 - jsdai_eclipse_arm.exl
 - Express-G_module (example of EXPRESS-G by module)
 - Jsdai_eclipse_arm_442_source (SMRLv12)
- And drag drop them to the Eclipse project, selecting the "Copy files and folders" option.



Verify EXPRESS compiler settings

The EXPRESS compiler settings should be set according to this screen shot.



Compile EXPRESS for ARM and MIM

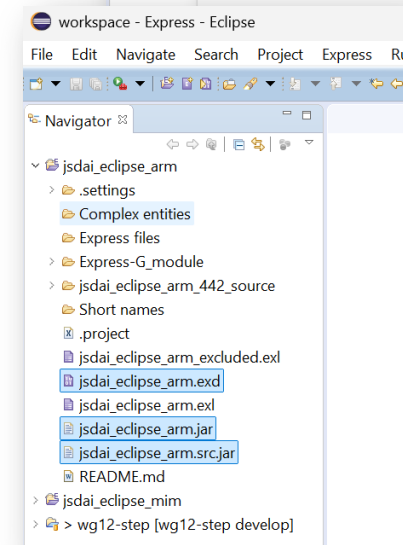
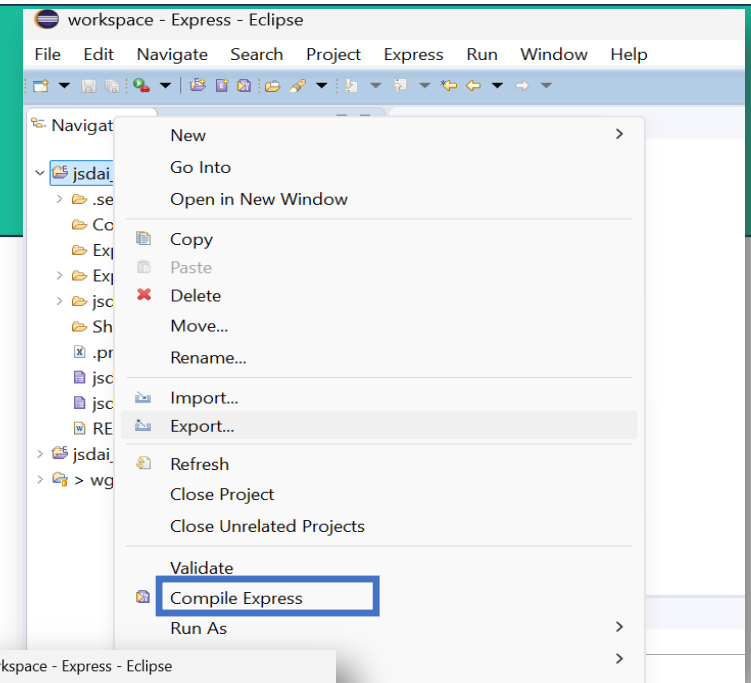
The "jsdai_eclipse_arm.exl" or "mim" file serves as the masterdata for compiling the EXPRESS.

It should be updated depending on the branch of the wg12-step repository.

Right click on the "jsdai_eclipse_arm" or "mim" project and select "**Compile EXPRESS**".

Three files are created as follows:

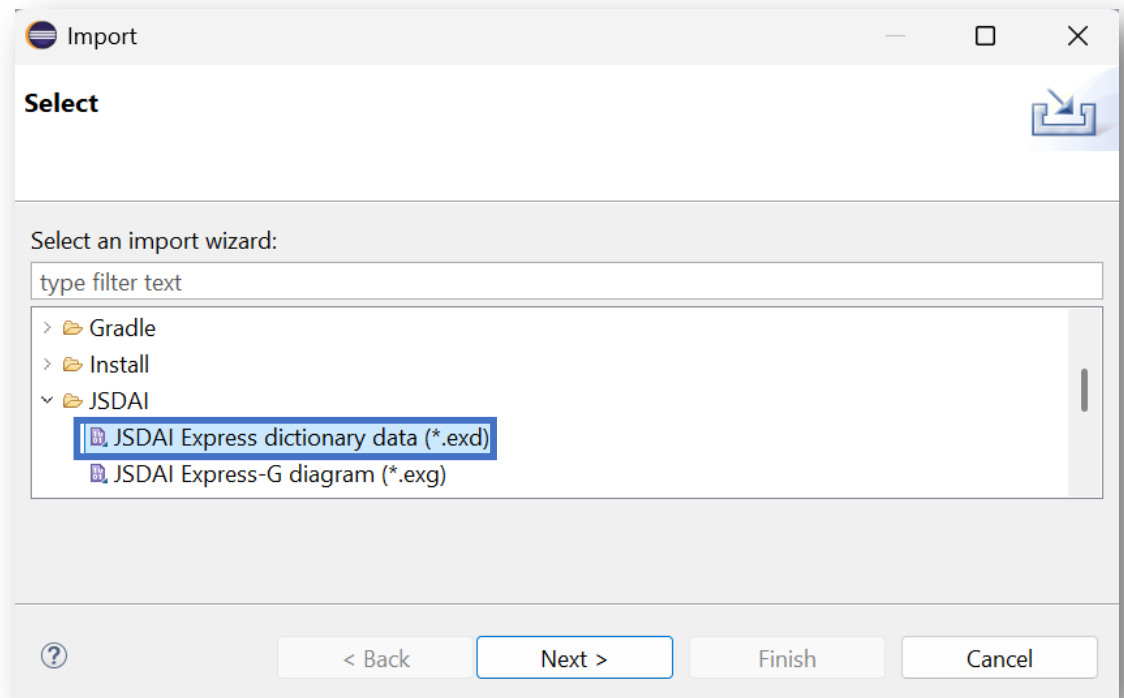
- .jar – Compiled EXPRESS schema in Java format (JSdAI).
- .exd – EXPRESS dictionary used for EXPRESS-G coherence.



Modifying an existing EXPRESS-G diagram (1/2)

From the folder
“jsdai_eclipse_arm_442_source”

1. Double click the
“jsdai_eclipse_arm.exg” to open it
2. To ensure the use of up-to-date data, we import the .exd that was created during compilation.
 - a) Select “**Import**” in the “**File**” menu then unfold “**JSDAI**” folder and select “**JSDAI Express dictionary data (*.exd)**”.
 - b) Verify that the path is correct and click “**Finish**”.

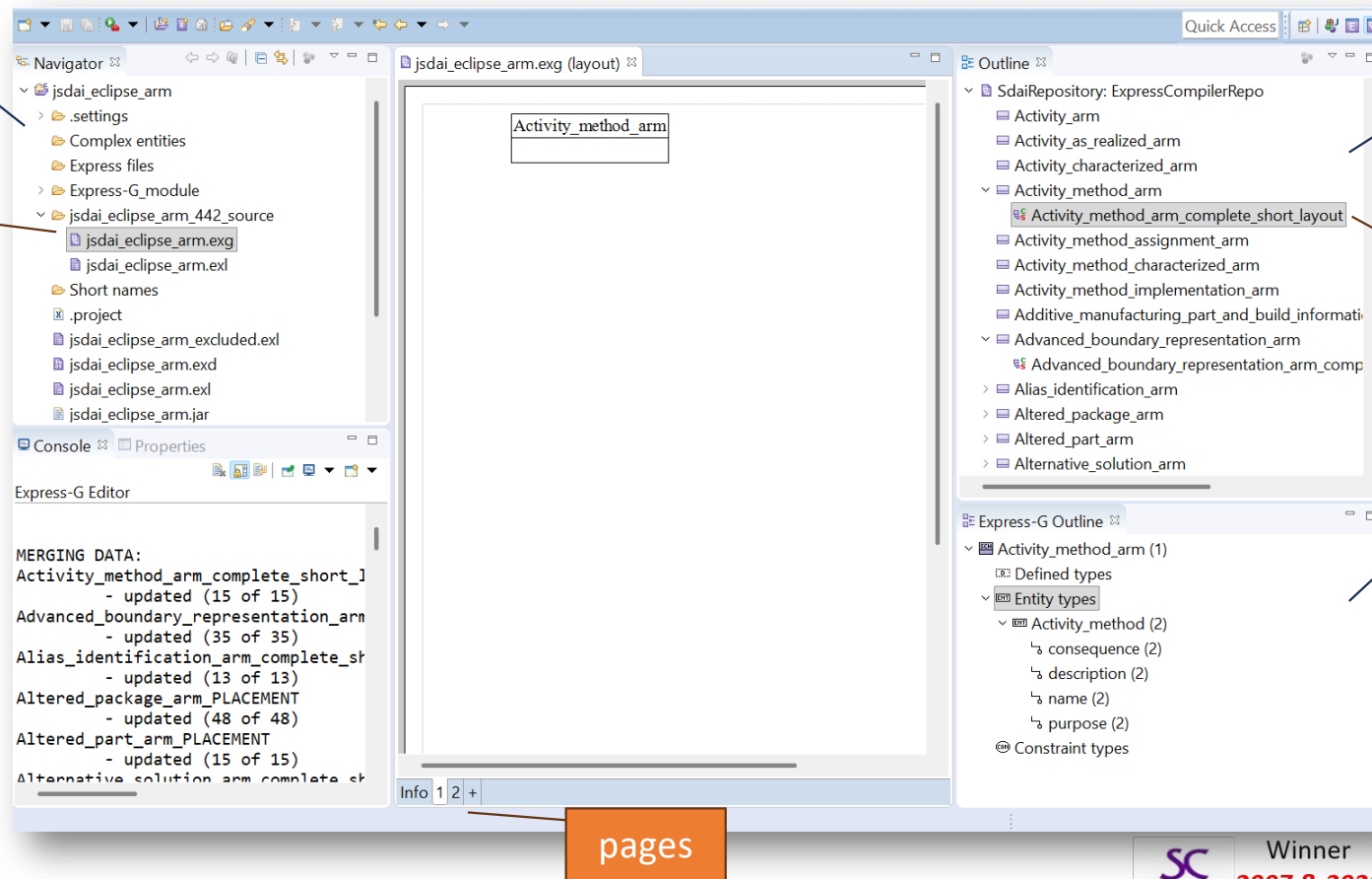


Modifying an existing EXPRESS-G diagram (2/2)

3. In the “**Outline**” pane, Unfold “**Activity_method_arm**” and select “**Activity_method_arm_complete_short_layout**”
4. You can move the “**Activity_method_arm**” box.
5. You can click the page 2 on the bottom of the central pane and move items.

Please note that changing the physical screen or computer can alter the layout of the diagram.

EXPRESS-G diagram in Eclipse



ISO/TC 184/SC 4/QC & WG12 STEP, EXPRESS-G with JSDAI

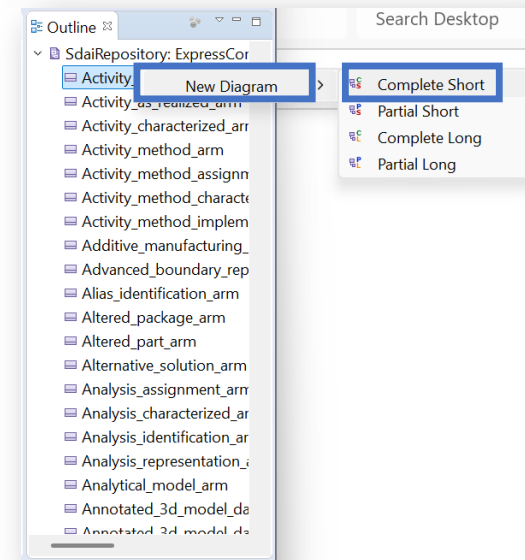
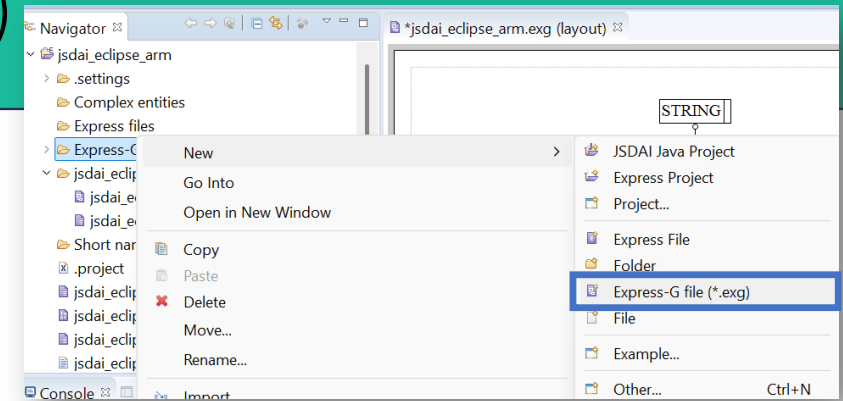
Winner
2007 & 2021
Lawrence D. Eicher Leadership Award



Generating an EXPRESS-G diagram (1/2)

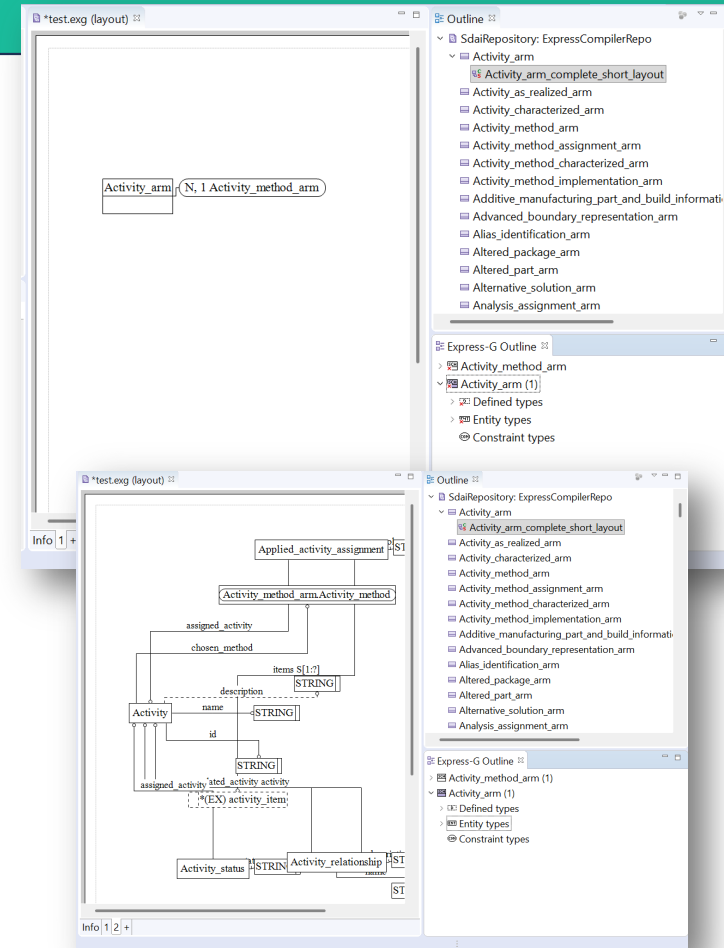
From the folder EXPRESS-G_module

1. Right click, then select "**New**" and "**EXPRESS-G file (*.exg)**", give it a name and click "**Next**" then select "**Import dictionary data at once**" (Verify the path), and click "**Finish**".
2. The new EXPRESS-G is selected in the "**Navigator**" pane.
3. In the "**Outline**" pane, right click on "**Activity_arm**" and select "**New Diagram**" and "**Complete Short**". Change the name if needed and click "**OK**".



Generating an EXPRESS-G diagram (2/2)

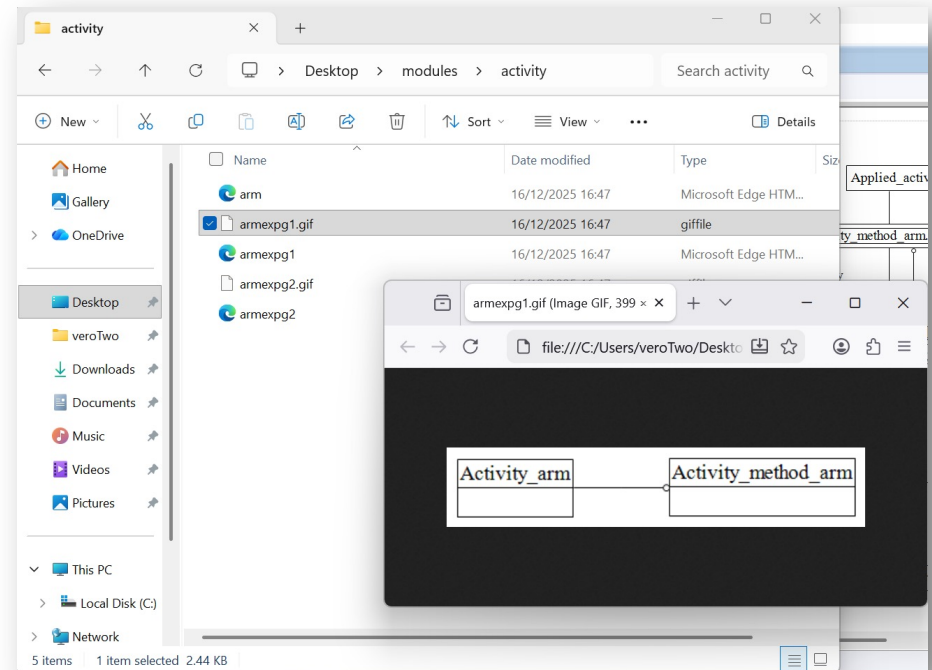
4. From “**EXPRESS-G outline**”, drag and Drop “**Activity_arm**” to the central pane.
5. Then drag and drop “**Activity_method_arm**” to update the relationship.
6. Note that “**Activity_method_arm**” no longer displays a red cross error icon. Please note: because of a bug, every ENUMERATION displays a red cross error icon.
7. Now add the second page, clicking the “+” at the bottom of the central pane, drag and drop the entities and the types.
8. Adjust the layout as required and save.



Exporting the EXPRESS-G components

Right click on the “.exg” file, select "**Export**", unfold the "**JSDAI**" folder and select "**E-G diagrams and stepmod XML**", click "**Next**", select the directory and adjust the other options.

Verify that the EXPRESS-G components are as expected.



Automatically producing the exl list.

Stuart to provide a method and test it.
How to push this to bitbucket? When?
Where? Who? How?

Where : <https://sd.iso.org/bitbucket-pilot/projects/ISOTC184SC4/repos/wg12-step/browse/support>

An automated tool.

On commits.

For now, <https://nextcloud.boost-lab.net/nextcloud/index.php/apps/files/files/191464?dir=/wg12-step/JSDAI>

folder develop and reflect the structure.

 Winner
2007 & 2021
Lawrence D. Eicher Leadership Award



A man in a light-colored suit and yellow tie stands behind a transparent digital screen. He is pointing his right index finger towards a large circular icon in the center of the screen. The screen displays a network of interconnected nodes and lines, with several prominent circular icons containing symbols such as a factory, a brain with gears, a microscope, a gear, a person's head, and a cloud. The background is a soft-focus indoor setting with warm lighting.

Contact Information

EM: Claudia Lueje: lueje@iso.org

© ISO, 2025
All rights reserved ISBN 978-92-67-10648-9


 Winner
2007 & 2021
Lawrence D. Eicher Leadership Award

