

PDX-Workspace/PDX lib managing ASAM MCD- 2D (ODX) data exchange via PDX packages

“Our customers rely on standards like ASAM MCD-2D (ODX) to develop future electronic and software systems. ESG bridges the gap between current and future needs.”

Thomas Weyrath
IT Systems & Function Development
ESG Elektroniksystem-
und Logistik- GmbH

I Summary

Challenge: An easy, manageable exchange of diagnostic and flash data between process partners is essential. For this purpose, the ASAM MCD-2D (ODX) standard introduces a file container called “PDX package”, where all files to be exchanged are packaged. However, the content of the PDX package, the hierarchical directory structures and the naming conventions can vary due to different processes or software systems. Hence the data exchange can be difficult, even erroneous in particular data exchange between different companies.

Solution: On behalf of our client BMW, ESG has developed PDXlib, a java library with functions to manage the data exchange via PDX packages. The PDXlib was integrated as an eclipse “Rich-client-Platform” plug-in, called PDX-Workspace, where the user can easily execute the exchange functions.

Key Benefits: With PDX-Workspace/PDXlib, process partners can exchange and process their data with respect to their own system environments.

- When the ECU is developed by a system supplier or through a joint venture between two or more OEM’s then the PDX package must be exchanged between the project participants.
- Inside the OEM engineering department, various development testers verify the system and subsystems with a “single source” data. In this case several ECU specific PDX packages must be merged into a combined PDX package.
- The engineering department in turn transfer the verified data to the software logistic department, where a verified vehicle database is maintained.
- This database is then delivered to the responsible manufacturing and service departments for end-of-line tests respectively for diagnostic and programming purposes by the software logistic department.

II Situation

In the exchange process, PDX packages are used as follows: one process partner builds the PDX package by including all files he wants to transfer; another process partner receives the PDX package and extracts the files. The following use cases can occur:

- The OEMs software logistic department defines the common set of files (e.g. the non ECU specific parts) and transfer them to the OEM engineering department.
- The OEM engineers get the common data and enrich them with the ECU specific parts (e.g. the base variant and the ECU variants).

III Challenges

Due to varying processes and infrastructure through out the process, partners the requirements for the data management can vary significantly. Hence, there is a general need to harmonize the handling of PDX packages to perform all above mentioned use cases without manually work overhead.

IV Success strategy

An automatic harmonization of names and structure can here be achieved by using the PDX-Workspace and PDXlib and helps the user to:

- package arbitrary source files in the file system into a new or existing PDX package
- unpack a PDX package in whole or in parts
- merge several PDX packages into a combined PDX package
- verify the content of a PDX package

PDX-Workspace/PDX lib managing **ASAM MCD-2D** (ODX) data exchange via PDX packages

- update the common set of files in a PDX package

Functions, which create or modify PDX packages, automatically update the PDX package catalogue (XML-based content description file).

Hierarchical directory structures and naming conventions are not hard coded but configurable. Differences between PDX package and target system are described in structure description files (also XML documents) which are used for packaging or unpacking PDX packages.

Specific parts can be packaged or unpackaged using known regular expression syntax like “*.odx-d” or “*.odx-?” etc.

V **Challenges during the project**

The main challenge was to find a concept which is flexible, scalable and open for future requirements. In addition, usability and ease of use was important. For this reason an eclipse RCP plug-in architecture was chosen.

VI **Business benefits**

The PDX-Workspace / PDXlib reduces integration time and avoids failure since no manually interventions are needed. This development leads to many benefits towards exchangeability of data between the OEMs and the suppliers.

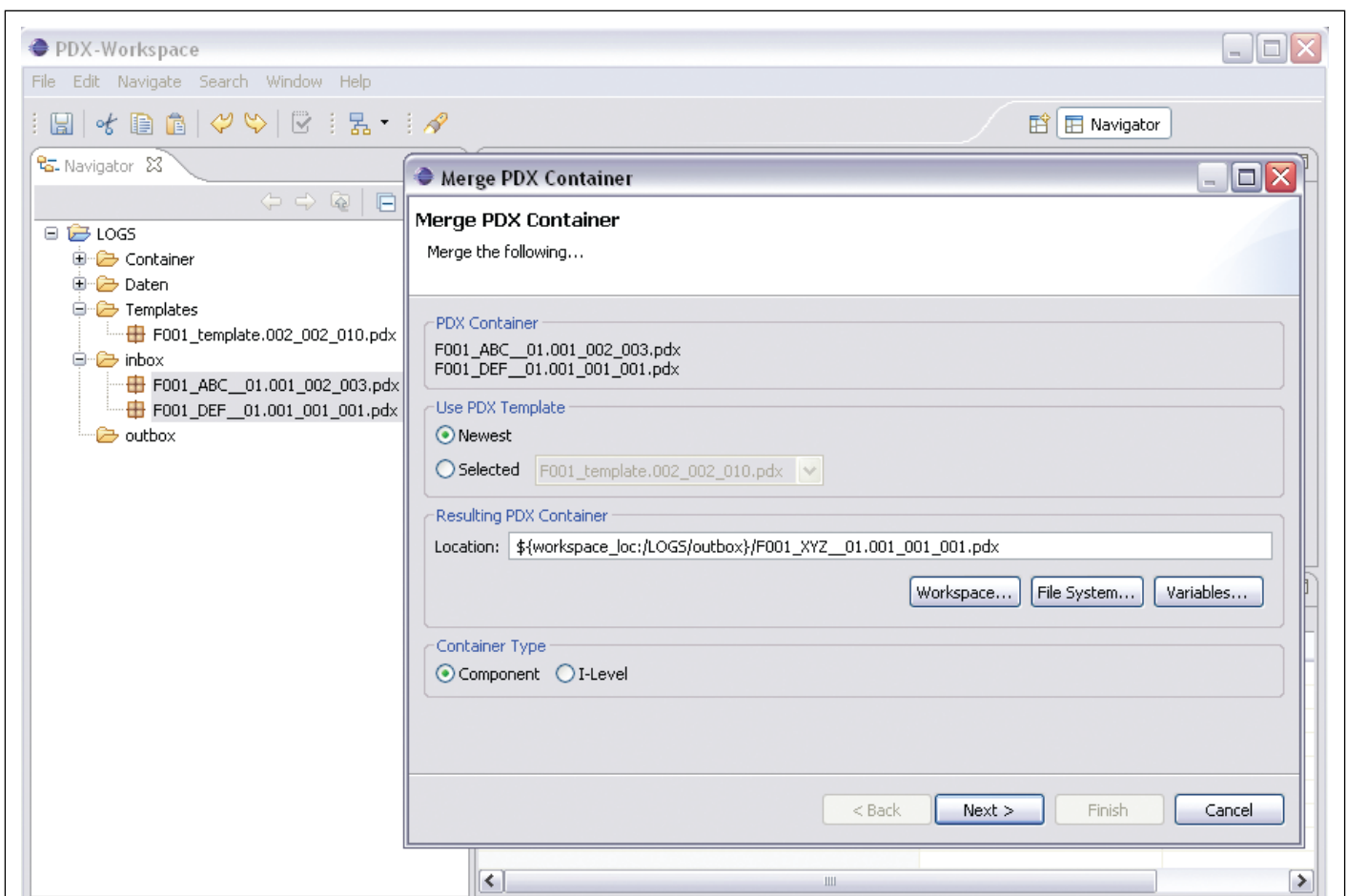


Figure 1: Eclipse rich-client-platform as graphical user interface for the PDXlib functionalities