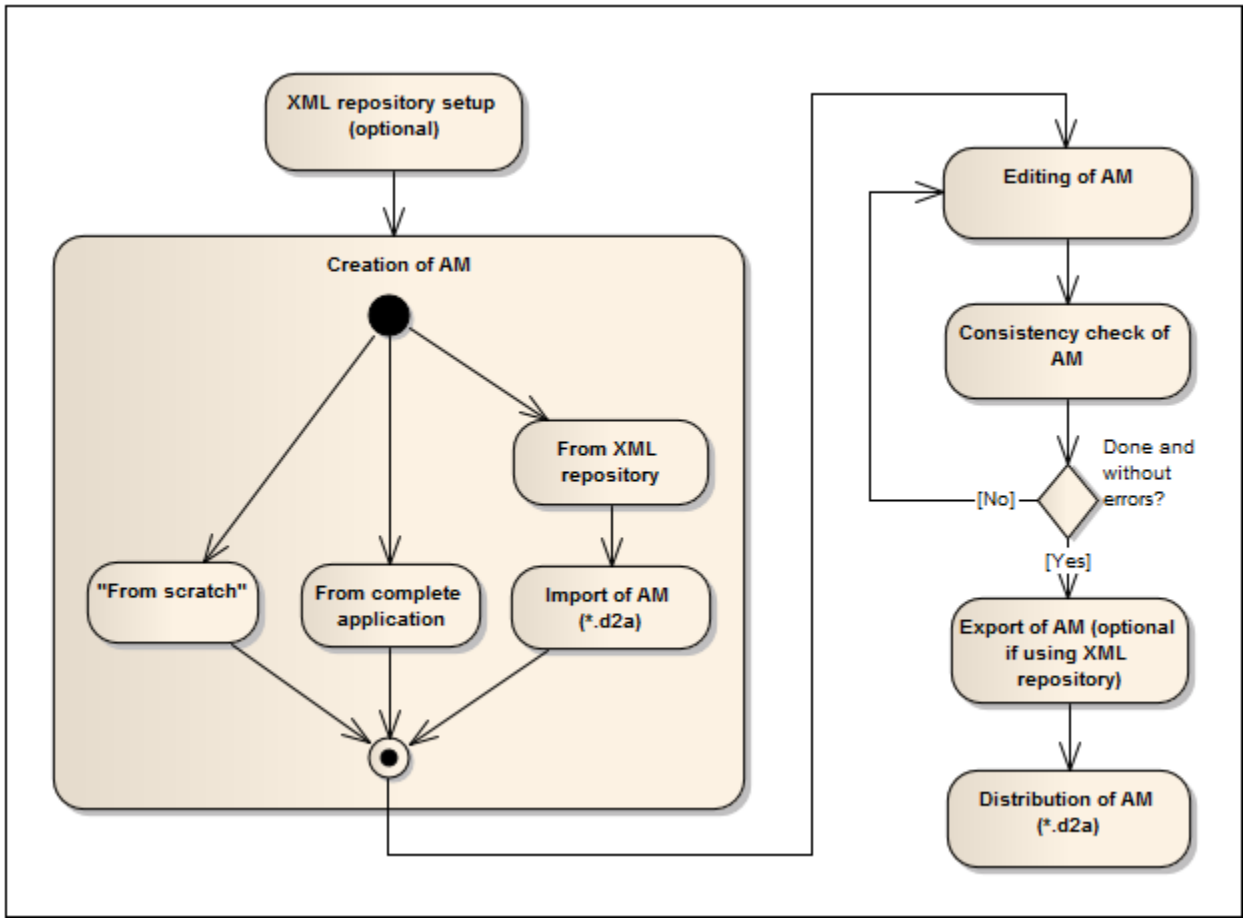
Development cycle

Development cycle of application module

A development cycle of application module consists of six basic steps. Their sequence is displayed on the picture.



1. XML Repository setup (optional)

XML Repository is useful function for the process when both the application module and the application are developed. This repository is specific in its configuration that should be as follows:

Enabled settings:

- To set a file time according to an object modify time.
- To check an object configuration before it is being exported.
- To set ID = 0 for the exported objects.
- To use UTF-8 encoding for the exported files.
- The export with optional membership in the groups.
- The export to "export subdirectory" of the object.
- To change a size of structured variables those size is parameterized.
- To reset the parameterized configuration properties.

Disabled settings:

- The recursive export of the object children.
- The recursive export of the objects that are used by other objects.
- The export of the system objects and variables.
- The export of the logs about "object life".

If the objects are set according to above mentioned a user will always obtain XML files suitable for creating an archive of the application module and distributing of the module. XML files created in this way are not dependent on the current configuration of the parameterized configuration properties.

2. Creation of AM

1. **From scratch**
   Create the application module according to instructions mentioned in this chapter.
2. **From complete application**
   Create the object of *Application module* type and configure its parameters, insert the objects of the application to the application module and parameterize the selected configuration properties.
3. **From XML Repository**
   XML files from XML Repository for application module cannot be imported directly to the application which results from the export setting "**Null parameterized configuration properties**". It can cause that XML file will contain invalid object configuration. Therefore, you have to create the archive of the application module by a D2Archive utility first and then proceed with the step No. 4.
4. **By import of AM archive**
   By the import of AM archive you obtain an operational configuration of application module. The parameter values may be changed at any time through the item of extended actions opened over application module:
   - Show/modify values of module parameters - enables to change the configuration of parameter values of imported module so that the change appears in the member objects.

   You can find an useful information if module of older or current version exists in application - through the following items:
   - Compare module parameters - enables to detect the suspect changes in the module parameters (mainly the change of dependency, removal / adding of some parameter).
   - Compare public members - enables to detect changes in public members of module – adding, removal or change of public member configuration.

3. Editing of AM

Editing of AM means the editing of application module parameters and their usage, and editing of member objects. The information from the following extended actions (activated over the application module) may be useful when object is being edited:

- Module dependencies - informs which objects of another modules are used in the member objects.
- Application of module parameters - informs where and which parameter is used and it alerts to invalid (deleted) parameters.
- Rename module parameters - enables to change the name of the parameters so that the module consistency is kept, i.e. it renames the parameters in each of the configuration properties as well as.
- Preview of the module parameters shows a layout of the dialog box of module configuration parameters when it is being imported.

4. Consistency check of AM

A periodic check of application module during its development helps you to recognize what should be changed in module configuration so that the usable AM archive could be created. The check is called automatically before export of AM.

5. Export of AM (optional)

AM is exported through the item of extended actions opened over the object of *Application module* type. Selected object and all its member objects will be exported according to settings described in the step 1. The consistency check goes automatically before the export. If the module configuration is not consistent it stops export.
The applying of XML Repository enables to export the objects automatically while they are being saved. The consistency is not checked automatically, it must be called manually through the item of extended actions opened over application module. It prevents the inconsistent state of module in XML Repository.

The editing of application module can cause an incompatibility of module with its previous version. This differences you can see in the dialog box opens through the extended actions for AM:

- **Compare module parameters** - compares the parameter configuration of the older version of AM with current one. This dialog box shows parameters that were added, removed or changed their dependency against the other parameter.
- Comparison of public application module members - compares the public members of current and previous version of module (from AM archive). The list gives an information on public members that were added, removed or changed. If some public member was removed you must raise the major version of AM so that the module compatibility could cease to operate.

6. Distribution of AM

It means to create AM archive through D2Archive utility from XML files of exported objects of application module which have been gained from XML Repository module or by the export of AM.

Related pages:

Application module
D2000Archive utility