# Conversion Tools - Feature #7169

## drive conversion order using user-specified dependencies

03/03/2023 02:09 PM - Greg Shah

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Greg Shah | **% Done:** | 0% |
| **Category:** | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | |
| **billable:** | No | **vendor_id:** | GCD |

**Description**

**Related issues:**

| | |
|---|---|
| Related to Conversion Tools - Feature #6256: improved profile support | **WIP** |
| Related to Conversion Tools - Feature #6253: add file-set processing into p2j... | **Closed** |
| Related to Conversion Tools - Feature #7180: create a centralized manager for... | **New** |
| Related to Conversion Tools - Feature #7179: rework the conversion process to... | **New** |
| Related to Conversion Tools - Bug #6082: automatically add to conversion list... | **New** |

### History

#### #2 - 03/03/2023 02:30 PM - Greg Shah

*- Related to Feature #6256: improved profile support added*

#### #3 - 03/03/2023 04:04 PM - Greg Shah

*- Related to Feature #6253: add file-set processing into p2j.cfg.xml added*

#### #4 - 03/03/2023 04:09 PM - Greg Shah

The profile approach (see #6256) provides the ability to segment a larger code base into named profiles. An important objective of doing this is to be able to convert minimum sets of functionality rather than requiring the entire application to be converted at once. To make this work properly, we must be able to specify dependencies between profiles and the priority or order of conversion within profiles. This is important because converting an arbitrary profile may depend on files in one or more other profiles. If we cannot find the conversion artifacts for a dependency, then we cannot properly convert. This also matters within a profile. When all files are converting in a single batch, it is possible to convert procedures in a simple alphabetical order. When some files are converted separately, we may need control over the ordering such that we can get correct results.

#### #5 - 03/03/2023 04:10 PM - Greg Shah

Constantin suggested the following:

> As I see it, you need a custom comparator to sort the file list according to some rules.
>
> Greg: for this, a custom comparator could be configured in p2j.cfg.xml, and if that parameter exists (and if the class can be loaded), then sort the program list using that. It's just a matter to write the comparator in Java, add it to a jar and to the conversion classpath.
>
> But I don't think this can work with the .cls files, as those need some special processing in FWD, only with the plain .p programs.

#### #6 - 03/03/2023 04:12 PM - Greg Shah

The comparator is an interesting idea.  It certainly would provide a great deal of customizable control.  However, it brings with it:

- Extra complexity because now we have to have extra code that is compiled and found at conversion time.
- Dependencies are hidden inside whatever Java code is written, rather than being made explicit in configuration.

For this last reason especially, I prefer to avoid the comparator approach.

#### #7 - 03/03/2023 04:18 PM - Greg Shah

For dependencies between profiles, I think we should a add a depends element which encodes one or more profiles on which this profil depends.  In other words, that list of profiles must be converted first.  A key idea here is this must be an acyclic directed graph.  There must be no direct or indirect cycles (profiles which depend upon themselves directly or indirectly).

For ordering within a given profile, I think we can just allow the order of the file-set to define the order of conversion.  This may mean that the file-set will no longer be a simple directory spec, but I think that is an acceptable trade off.  Breaking out the entries and ordering them explicitly makes the result obvious since it is encoded explicitly in the configuration.

Today we just sort alphabetically, but the only real advantage to that is that it is easy for a human to see where in the overall code the processing is occurring.  That is useful but not critical.  The idea of dependencies must break that alphabetical ordering anyway, so there is no extra loss here.

For OO we must process the objects in their object graph order.  We already do this, but we will need changes to handle it across profiles.

#### #8 - 03/06/2023 02:36 AM - Constantin Asofiei

Greg, if I understand correctly, your suggestion is to order the non-OO file list using the include/exclude filters from the file-set in p2j.cfg.xml, for whatever profile gets converted.  A problem I see here, is we use the include/exclude filters for example to exclude an entire folder and just cherry-pick some files from the excluded folder, with include filters, after that.  Better said: there is already a semantics for the order of include and exclude filters in a file-set.  In any case, it should not be a problem to enable sorting using the order of the include filters in the file-set.

#### #9 - 03/06/2023 03:02 AM - Stefan Brands

Are profiles already supported today ? And if so, is there an explanation on how to use them ?

#### #10 - 03/06/2023 06:41 AM - Greg Shah

> Greg, if I understand correctly, your suggestion is to order the non-OO file list using the include/exclude filters from the file-set in p2j.cfg.xml, for whatever profile gets converted.  A problem I see here, is we use the include/exclude filters for example to exclude an entire folder and just cherry-pick some files from the excluded folder, with include filters, after that.  Better said: there is already a semantics for the order of include and exclude filters in a file-set.  In any case, it should not be a problem to enable sorting using the order of the include filters in the file-set.

I'm more focused on the directory and filename directives than the include and exclude filters.  The excludes we don't have to worry about because those files are never converted so the order of non-conversion doesn't matter.  I agree that there might be cases when we want a different order than can be expressed with a minimal file-set definition.  A brute force solution to that problem is to more explicitly list directory or filename entries that otherwise would not be needed.

If we find that to be a common case, we can consider adding other syntax but for now I prefer if we can solve this using the existing approach.

#### #11 - 03/06/2023 06:43 AM - Greg Shah

> Are profiles already supported today ? And if so, is there an explanation on how to use them ?

Yes, these are supported today.  Unfortunately, we have not yet documented the features.  You can see the discussions including some configuration syntax in #6256 and #6253.

#### #12 - 03/06/2023 03:57 PM - Stefan Brands

These profiles might be something that can help me. A couple of questions :

- Can the propath include directories outside the file sets inside the profile ?
- How are these profiles processed during a conversion ? Is it in the order they are specified in the config file ? So that it has the same effect as what I did by naming the folders with number prefixes so that they are alphabetically in the correct dependency order ? ( projects with least dependencies first )

To explain this a bit, now I have folders like this :

a005zzz/src/api
a006zzz/src/impl
a007bbb/src/api
a008bbb/src/impl

The above guaranties that we can convert a007bbb/src/api separately ( as long as we also include a006 and a005 and everything above it ) and then use the result in a full conversion environment. This only works if we don't make temp table changes that conflict with projects depending on a007. This is already a big help for us, but the a unique consistent naming of DMOs would help us much more of course.

#### #13 - 03/06/2023 04:08 PM - Stefan Brands

btw, for now we could live with a schema shared between the profiles.
Would the profiles name the DMOs with a namespace ? That could make the DMO names more unique ?

#### #14 - 03/06/2023 04:18 PM - Stefan Brands

Temp table DMOs I mean.

I don't know why, but I can't modify my own comments in this ticket...

#### #15 - 03/06/2023 04:28 PM - Greg Shah

> Can the propath include directories outside the file sets inside the profile ?

Yes, but in practical terms this is mostly useful for include files. Our OO approach still requires all referenced OO .cls files to be converted in a single batch.

> How are these profiles processed during a conversion ? Is it in the order they are specified in the config file ?

No, today the code builds up an overall list of files to process and alphabetizes them all into a single list. My idea is to specify the dependencies between profiles and ensure that when you convert profile C which depends on B (and B depends on A), that we:

1. Have access to the converted code for B and A such that we can read any needed conversion details from the artifacts in the jar. OR
2. Convert the missing profiles in the proper order. If B is missing but A exists, then we convert B before C. If A is missing, then we convert A before whatever we do for B or C. And so forth...

> So that it has the same effect as what I did by naming the folders with number prefixes so that they are alphabetically in the correct dependency order ? ( projects with least dependencies first )

For now it is a way to specify an app-level configuration but it doesn't cleanly resolve the various other issues of converting app by app.

**#16 - 03/06/2023 04:29 PM - Greg Shah**

> Would the profiles name the DMOs with a namespace ?

It does not do this today.

> That could make the DMO names more unique ?

Eric?

#### #17 - 03/07/2023 03:54 AM - Stefan Brands

Greg Shah wrote:

> No, today the code builds up an overall list of files to process and alphabetizes them all into a single list.  My idea is to specify the dependencies between profiles and ensure that when you convert profile C which depends on B (and B depends on A), that we:
>
> 1. Have access to the converted code for B and A such that we can read any needed conversion details from the artifacts in the jar. OR
> 2. Convert the missing profiles in the proper order.  If B is missing but A exists, then we convert B before C.  If A is missing, then we convert A before whatever we do for B or C.  And so forth...

This would indeed make it possible to do a conversion of a single app. Especially if you combine this with putting DMOs in the namespace of the app wher it is defined in. I think this is the right solution.

#### #18 - 03/07/2023 08:18 PM - Greg Shah

*- Related to Feature #7180: create a centralized manager for the conversion list and per-file/project status and logging added*

#### #19 - 03/07/2023 08:20 PM - Greg Shah

*- Related to Feature #7179: rework the conversion process to ensure that it can finish an entire run even when there are failures after parsing added*

#### #20 - 03/07/2023 08:29 PM - Greg Shah

We should implement a dependency manager to track dependencies and allow cascading adds and deletes from the conversion list (see ##7180 and #7179).

#### #21 - 03/07/2023 08:30 PM - Greg Shah

The OO processing today does part of this job, for the object graph.  It would be moved into this external dependency manager.

#### #22 - 05/02/2023 09:49 AM - Greg Shah

*- Related to Bug #6082: automatically add to conversion list all non-skeleton .cls dependencies added*

#### #23 - 05/06/2023 09:42 AM - Greg Shah

*- Assignee set to Greg Shah*