# Kepler - Bug #3575

## A representation of COMAD collections on the file-system

10/27/2008 02:33 PM - Daniel Zinn

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 10/27/2008 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Timothy McPhillips | **% Done:** | 0% |
| **Category:** | actors | **Estimated time:** | 0.00 hour |
| **Target version:** | 3.X.Y | **Spent time:** | 0.00 hour |
| **Bugzilla-Id:** | 3575 | | |

**Description**

It would be useful if there were a representation of Collections in the file system. In particular, I could imagine using directories to represent collections (say named with a running number and the name of the collection). Then we could use files to represent data items and mata-data (for collections and data); For each data Token we could name the file the same as the Type of the Token (with a leading running id), and store a serialized version of the data token in it (which would be easy for strings, for example). We could use the same file-name with a suffix of, say, .METADATA:owner to store the metadata with the key owner inside (possibly also with the type of the metadata in the string of the filename).

This would not only make collections browse-able via standard file-system tools, but since there exist distributed filesystems (such as the hadoop filesystem) the size of these collections can easily scale up to TBs of data.

This is somewhat similar to request #3573, but aims more towards a general 'storage'-backend for COMAD 2 collections.

I am not proposing that some intermediary result should be represented as directories (though my request does not exclude this either). I am just requesting that besides the XML-representation of COMAD collections (that we currently have, right?) it would be good to have a representation that is file-system based. Similar to the XML representation, which is not materialized within a workflow run (ie, during the actors), the directory-representation need not to be materialized inside the workflow. Instead it should be a user-friendly way of browsing (and even creating) COMAD collections with ordinary file-manipulation tools. You can then copy or move content from one collection (=directory) to some other collection. This representation can then be used as inputs for workflows and can be an output format (to have a closed-loop system).

Besides gaining the power of simple file(system)-manipulation tools back for COMAD collections, this representation can be stored on a distributed file-system (ie hadoop fs) and the collection with the data can so easily hold terabytes of data.

What I am proposing are two actors, one that reads a (special) directory into a COMAD collection and one that can save any comad collection (stream) into a specially formated directory.

### History

**#1 - 03/27/2013 02:23 PM - Redmine Admin**

Original Bugzilla ID was 3575