

# EUROPEAN MIDDLEWARE INITIATIVE

## EMI DATA LIBRARY CONSOLIDATION PLAN

### EU TECHNICAL NOTE: DXN.M

---

Document identifier: **EMI\_datalib-consolidation-EMI-tech-doc-v5.docx**

Date: **16/01/2012**

Activity: **JRA1**

Lead Partner: **UiO**

Document status: **DRAFT**

Document link:

---

**Abstract:**

In this document a plan for consolidating the gLite and ARC data libraries into one EMI data library is described.

**Copyright notice:**

Copyright (c) Members of the EMI Collaboration. 2010.

See <http://www.eu-emi.eu/about/Partners/> for details on the copyright holders.

EMI ("European Middleware Initiative") is a project partially funded by the European Commission. For more information on the project, its partners and contributors please see <http://www.eu-emi.eu>.

This document is released under the Open Access license. You are permitted to copy and distribute verbatim copies of this document containing this copyright notice, but modifying this document is not allowed. You are permitted to copy this document in whole or in part into other documents if you attach the following reference to the copied elements: "Copyright (C) 2010. Members of the EMI Collaboration. <http://www.eu-emi.eu>".

The information contained in this document represents the views of EMI as of the date they are published. EMI does not guarantee that any information contained herein is error-free, or up to date.

EMI MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, BY PUBLISHING THIS DOCUMENT.

### Delivery Slip

	Name	Partner / Activity	Date	Signature
<b>From</b>	Jon Kerr Nilsen	UiO/JRA1		
<b>Reviewed by</b>				
<b>Approved by</b>				

### Document Log

Issue	Date	Comment	Author / Partner
1	9/8/2011	First draft	Jon Kerr Nilsen/UiO, Zsombor Nagy/NIIFI
2	17/8/2011	Update based on meeting 15/8/2011	Jon Kerr Nilsen/UiO
3	02/11/2011	Comments from Zsolt and Padova AHM discussion Minor document cleaning Updates/comments for Q&A section	Zsolt Molnar/CERN, Jon Kerr Nilsen/UiO
4	09/01/2012	Updates based on F2F meeting 24-25/11/2011 - added 3 <sup>rd</sup> party transfer library to design - added appendix with gLite/ARC comparison tables - updated Q&A section	Jon Kerr Nilsen/UiO
5	12/01/2012	Comments from Zsolt	Zsolt Molnar/CERN, Jon Kerr Nilsen/UiO
6			
7			
8			

### Document Change Record

Issue	Item	Reason for Change
1		
2		
3		

## TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>5</b>
1.1. PURPOSE .....	5
1.2. ABOUT THIS DOCUMENT .....	5
1.3. REFERENCES .....	5
1.4. DOCUMENT AMENDMENT PROCEDURE .....	5
1.5. TERMINOLOGY .....	5
<b>2. TECHNICAL DESCRIPTION .....</b>	<b>7</b>
2.1. DESCRIPTION OF THE CURRENT DATA LIBRARIES .....	7
2.1.1 <i>gLite Data Library</i> .....	7
2.1.2 <i>ARC Data Library</i> .....	8
2.2. CONSOLIDATION PLAN .....	9
2.2.1 <i>Benefits</i> .....	10
2.2.2 <i>Disadvantages</i> .....	10
2.2.3 <i>Components to be removed</i> .....	10
2.2.4 <i>Components to be created</i> .....	10
2.2.5 <i>Components to be modified</i> .....	11
2.2.6 <i>Estimated amount of work needed</i> .....	11
2.2.7 <i>Estimated SLOC change</i> .....	11
2.3. QUESTIONS AND ANSWERS .....	11
2.3.1 <i>Could GFAL2 use libarcdata2 as a plugin instead?</i> .....	11
2.3.2 <i>How to support the transition period?</i> .....	11
2.3.3 <i>What about platforms outside EMI?</i> .....	11
2.3.4 <i>What about information system support?</i> .....	12
2.3.5 <i>What about naming and packaging?</i> .....	12
2.3.6 <i>Who will use this outside/after EMI?</i> .....	12
<b>3. APPENDIX.....</b>	<b>13</b>
3.1. ARC/GLITE DATA COMPARISON TABLE .....	13
3.1.1 <i>End-user clients</i> .....	13
3.1.2 <i>Libraries</i> .....	13

## 1. INTRODUCTION

### 1.1. PURPOSE

The goal of the EMI Data Library Consolidation activity is to find duplicate functionality within the ARC and gLite data libraries and to merge them into one EMI\_datalib. While reducing the amount of code to maintain is an important aspect, our main goal is to reduce the overall amount of maintenance. This means that, e.g., a clear division of components may be considered more beneficial than shrinking or removing some components.

### 1.2. ABOUT THIS DOCUMENT

This document describes the current gLite and ARC data libraries and the plan on how to merge them.

This document does not provide any details on how to implement or test the planned data library, this will be documented elsewhere.

The document is made by the EMI Data Library Consolidation TF in the EMI Data group, David Cameron (ARC Clients), Zsolt Molnar (CERN Data), Zsombor Nagy (ARC Clients), Jon Kerr Nilsen (ARC Clients) and Michail Salichos (CERN Data).

### 1.3. REFERENCES

<b>R1</b>	
<b>R2</b>	
<b>R3</b>	

### 1.4. DOCUMENT AMENDMENT PROCEDURE

This document can be amended by the authors further to any feedback from other teams or people. Minor changes, such as spelling corrections, content formatting or minor text re-organisation not affecting the content and meaning of the document can be applied by the authors without peer review. Other changes must be submitted to peer review and to the EMI PEB for approval.

When the document is modified for any reason, its version number shall be incremented accordingly. The document version number shall follow the standard EMI conventions for document versioning. The document shall be maintained in the CERN CDS repository and be made accessible through the OpenAIRE portal.

### 1.5. TERMINOLOGY

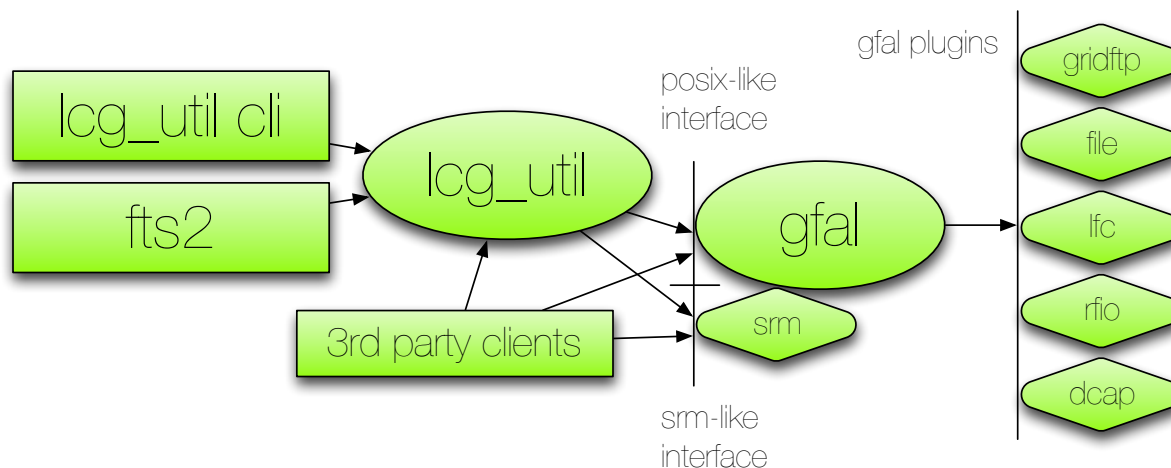
<b>ARC</b>	Advanced Resource Connector
<b>A-REX</b>	ARC Execution Service
<b>DCAP</b>	dCache Access Protocol
<b>FTS</b>	File Transfer Service
<b>GFAL</b>	Grid File Access Library
<b>GridFTP</b>	Grid File Transfer Protocol
<b>HTTP</b>	Hypertext Transfer Protocol
<b>LFC</b>	LCG File Catalogue
<b>POSIX</b>	Portable Operating System Interface for Unix
<b>RFIO</b>	Remote File I/O

<b>SRM</b>	Storage Resource Management
------------	-----------------------------

## 2. TECHNICAL DESCRIPTION

### 2.1. DESCRIPTION OF THE CURRENT DATA LIBRARIES

#### 2.1.1 gLite Data Library



**Figure 1: gLite Data Library Architecture**

The GFAL library provides a posix-like interface for data access. It has a pluggable architecture, where different plugins provide support for different data transfer protocols. It also has a plugin for file catalog (LFC) interaction, which can resolve logical names to transfer URLs.

Besides the posix-like interface, GFAL has an SRM interface which also supports usage of space tokens.

Also part of the gLite data library is the lcg\_util library, which provides higher-level file operations and which uses GFAL for the actual file access. The lcg\_util command line tools are thin wrappers around the lcg\_util library.

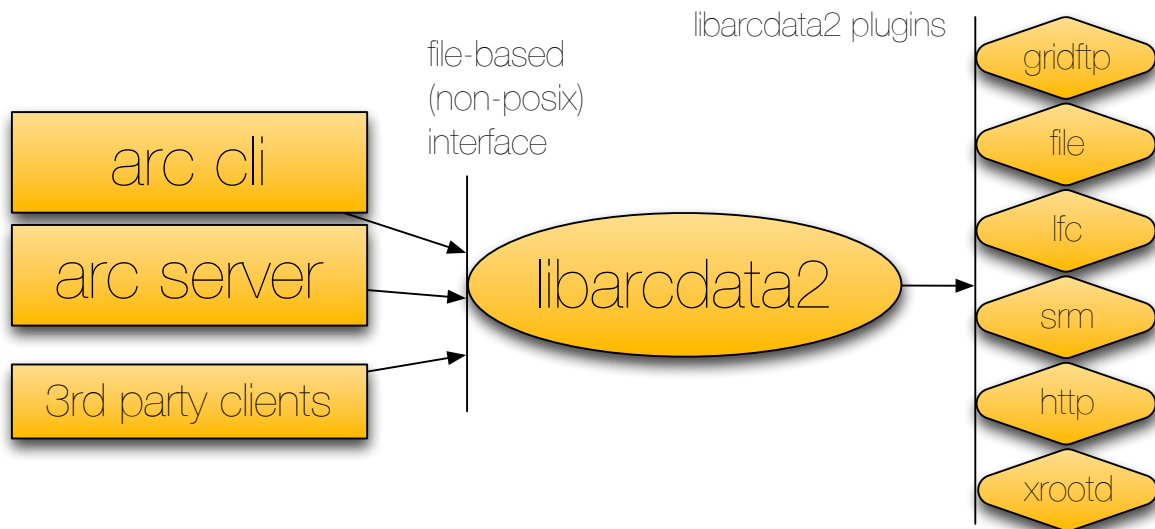
There are several third party clients (integrated into experiments' software) using the GFAL library or the lcg\_util library (some through the python API).

The GFAL and lcg\_util libraries are written in C.

It should be mentioned here that a new version of GFAL, GFAL2, is being implemented. This is a complete reimplementaion of GFAL with new plugins for the different protocols. This means that SRM will be implemented as a regular plugin and not as a part of the GFAL2 interface. For the implementation of the consolidation plan only GFAL2 should be considered.

The planned reimplementaion of FTS, FTS3, is planned to use the GFAL2 library.

## 2.1.2 ARC Data Library



**Figure 2: ARC Data Library Architecture**

The ARC data library called libarcdata2 has a pluggable architecture, different plugins providing support for different data transfer protocols. It is a file-based (non-posix) interface providing higher-level operations to read and write files. It has plugins for LFC and SRM operations (also supports usage of space tokens).

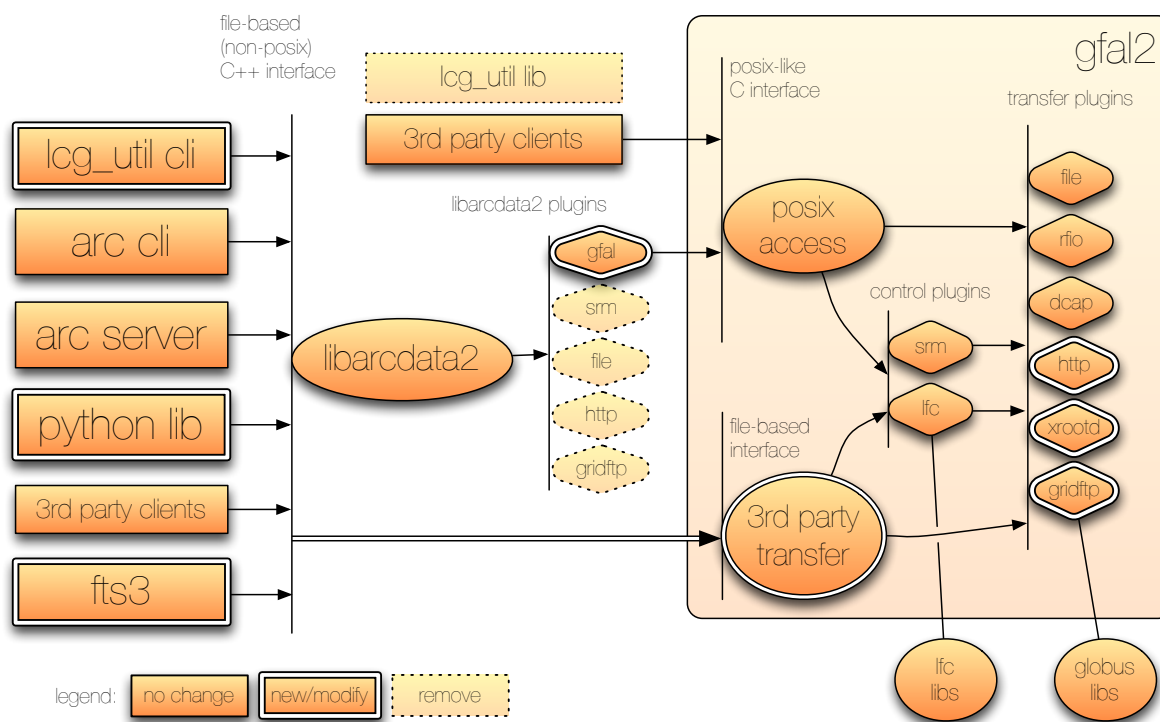
Being a file-based interface, libarcdata2 supports automatic buffer handling and checksumming. With the DataMover libarcdata2 can automatically copy from one location to the other, it can do a configurable number of retries with re-resolving replicas and it handles file caching.

The arc command tools and also the A-REX service on the computing element are using this library to upload and download files. Third-party clients also exist (some of them using the python API).

The libarcdata2 is written in C++.



## 2.2. CONSOLIDATION PLAN



**Figure 3: Proposed architecture of EMI\_datalib**

The proposed architecture (Figure 3) will keep both the POSIX-like interface (provided by GFAL2) and the higher-level file-based interface (provided by the libarcdata2 library). In addition, a transfer interface will be made available in GFAL2 to handle initiation and monitoring of 3<sup>rd</sup> party transfers.

In the proposed architecture, libarcdata2 will use the POSIX-like interface of GFAL2 through a GFAL-plugin. For the platforms where this plugin is available no other plugins will be needed for libarcdata2. While 3<sup>rd</sup> party clients requiring POSIX-based byte-wise data access can use the GFAL2 library directly, file based, data moving clients such as the lcg\_util and ARC CLIs and ARC CE will use the libarcdata2 file-based interface. Additionally, FTS3 and parts of the lcg\_utils and ARC CLIs will use the 3<sup>rd</sup> party transfer library. Hence, the lcg\_util library can be removed. A Python library will be created to replace the needed functionality of the lcg\_util Python API.

Note that the LFC and SRM plugins are put under GFAL2, even though they are (strictly speaking) file based. Having these plugins under GFAL2 allows POSIX access to files without needing to know physical file locations. For the same reasons, even though HTTP and xrootd plugins already exist in libarcdata2, HTTP and xrootd TURLs may be returned from the SRM plugin and the corresponding plugins should therefore be contained within GFAL2.

For the special case of 3<sup>rd</sup> party transfer, needed by FTS3, neither libarcdata2 nor GFAL2 currently have support for this. As this feature is crucial for FTS3 and will require some work to be fully supported in the libarcdata2 – GFAL2 chain, an intermediate 3<sup>rd</sup> party transfer library will be made available. To take advantage of the existing GFAL2 transfer plugins, this library will be closely related to the GFAL2 plugin. The intention is that FTS3 will first use this new library directly, while later libarcdata2 will add support for 3<sup>rd</sup> party transfer using this library so that also FTS3 can use the high-level C++ interface of libarcdata2.

### 2.2.1 Benefits

- One set of file transfer plugins instead of two
- One set of control plugins instead of two
- One file-based library and one posix-like library
- GFAL gets support for http and xrootd transfer
- ARC gets support for third party transfer
- ARC gets support for catalogues and BDII
- No change for existing users
- A clear division of components, with one file-based layer and one underlying posix layer with the transfer plugins should simplify support.

### 2.2.2 Disadvantages

- As development of GFAL will need to focus on the consolidated data library, no new features will be added to the current lcg\_util/GFAL implementation. This means that users need to do more radical changes to get new features.
- Since the GFAL2 API will not be backwards-compatible, changes in third-party clients are needed to move to GFAL2.
- One may argue that the consolidated library does not lead to a huge removal of lines of code when compared to the effort needed.

### 2.2.3 Components to be removed

- lcg\_util library
- lcg\_util clis doing catalog handling
- libarcdata2 plugins
  - file
  - gridftp
  - srm
  - lfc
  - http
  - xrootd

### 2.2.4 Components to be created

- GFAL plugin for ARC
- http plugin for GFAL2
- xrootd plugin for GFAL2
- gridftp plugin for GFAL2
- lfc plugin for GFAL2
- 3<sup>rd</sup> party transfer library

It should be emphasized that GFAL2 is a new implementation, and new plugins would have to be implemented regardless of the consolidation plan.

## 2.2.5 Components to be modified

- FTS3 to use GFAL2 (note that FTS3 is not yet implemented, only plan needs to be modified)
- libarcdata2 and GFAL2 need to provide a way to support requests for third party transfers
- lcg\_util command line tools to use libarcdata2

## 2.2.6 Estimated amount of work needed

gLite:

1. need to finalize GFAL2
  - Work already shows promising results
  - Already more stable than GFAL
  - Will be ready for EMI2 update
2. removing lcg\_util is unproblematic
3. start with FTS3 built on new 3<sup>rd</sup> party transfer library

ARC

1. ARC will start creating a GFAL2 plugin
  - Plugin itself unproblematic
  - Need some work to handle on plugin supporting multiple protocols
2. The removal of plugins should be unproblematic
  - Will require some internal repackaging of ARC components

## 2.2.7 Estimated SLOC change

The modified components are not foreseen to change significantly in terms of SLOC. The http and xrootd plugins removed from ARC will be recreated as GFAL plugins and are not foreseen to change the overall SLOC.

## 2.3. QUESTIONS AND ANSWERS

### 2.3.1 Could GFAL2 use libarcdata2 as a plugin instead?

GFAL2 is written in C and has a POSIX interface, libarcdata2 is written in C++ and has a file-based interface. It is not feasible to transform posix-like C queries into file-based C++ queries.

### 2.3.2 How to support the transition period?

The libarcdata2 library can keep all the plugins until the functionality of the GFAL plugin (in libarcdata2) is at the expected level. If any change to the interface of libarcdata2 is needed, it should be done in a backward compatible way, this way the existing clients of libarcdata2 would function the same way as previously.

The new lcg\_util CLI will provide the same user interface but using the libarcdata2 instead of the lcg\_util library. A new Python API will be provided with the same functionality as the old lcg\_util Python API.

### 2.3.3 What about platforms outside EMI?

Until the platform support (different Linux flavors, Mac, Windows) of GFAL is sufficient to maintain the current level in ARC, libarcdata2 would have to keep the existing plugins, and support them outside EMI.

### **2.3.4 What about information system support?**

Currently GFAL2 is able to query some information from the information system. The libarcdata2 does not have any information system support. GFAL2 needs some parameters for the queries which it gets from environment variables or command line options, so in theory it could keep doing this even if a request came through libarcdata2. But this may involve changing slightly the URL handling of libarcdata2 to let through shortened/invalid URLs which would be resolved by GFAL2 from the information system.

### **2.3.5 What about naming and packaging?**

We think that it is not necessary right now to rename the libraries (e.g. removing the name "arc"). The packaging is not within the scope of this document, but it may be natural that the name EMI\_datalib is introduced as a (meta)package.

### **2.3.6 Who will use this outside/after EMI?**

Naturally, the ARC CE depends on libarcdata2, and will need to move over to the new EMI\_datalib. Since libarcdata2 is already plugin-based, this is not foreseen to have much practical implication but should of course be properly tested. FTS, which now has started a redesign to use GFAL2 for data transfer, will use EMI\_datalib. Additionally, lcg\_util has non-EMI users that will consequently also use EMI\_datalib.

### 3. APPENDIX

#### 3.1. ARC/GLITE DATA COMPARISON TABLE

##### 3.1.1 End-user clients

gLite	ARC	Comments
lcg-cp file:// ...	arcep	-
lcg-cr	arcep lfc://srm/lfc/[metadata] (inc. space tokens)	-
lcg-del	arcrm lfc://srm/lfc/[metadata]	-
lcg-rep	arcep lfc://srm/lfc/[metadata]	-
lcg-gt (getturls) proto	-	-
lcg-sd (set done)	-	-
Functional differences	-	-
Infosys interaction (site name / VO / host to end-point resolution)	-	Reuse gLite or ARC job libraries?
Specifying transfer protocol (by end-user)	Configurable through URL option "transferprotocol"	-
Automatic SURL generation	-	want to have this
- (uses protocol)	chksum on the fly (by internal buffer)	-

##### 3.1.2 Libraries

Comparison of the GFAL2 and libarcdata2 data libraries.

###### 3.1.2.1 Protocols

Available protocols (note, some GFAL2 protocols are just planned, but exist for GFAL1).

GFAL2	libarcdata2	Comments
GridFTP	GridFTP	-
SRM	SRM	-
LFC	LFC	-
file	file	-
rfio	-	-
GSIIdCAP	-	-
-	xrootd	xrootd not in prod. release yet
-	HTTPS	-
-	ARC	-
-	stdio	-

### 3.1.2.2 API's

Comparison of GFAL2 and libarcdata2 API's.

GFAL2	libarcdata2	Comments
POSIX	Functionality based	Limited overlap between gLite and ARC APIs
-	PrepareReading, PrepareWriting	-
-	StartReading, StartWriting	-
-	StopReading, StopWriting	-
-	FinishReading, FinishWriting	-
-	Check	-
-	Remove	-
gfal_stat	Stat	-
-	List	-
-	Methods for locations and registration	-
-	Helper methods	-
-	Set/Get methods	-
gfal_chmod	-	-
gfal_rename	-	-
gfal_lstat	-	-
gfal_access	-	-
gfal_readlink	-	-
gfal_creat	-	-
gfal_open	-	-
gfal_lseek	-	-
gfal_close	-	-
gfal_read	-	-
gfal_write	-	-
gfal_getxattr	-	-
gfal_listxattr	-	-
gfal_mkdir	-	-
gfal_opendir	-	-
gfal_closedir	-	-
gfal_readdir	-	-
gfal_rmdir	-	-

gfal_unlink	-	-
gfal_posix_clear_error	-	-
gfal_posix_release_error	-	-
gfal_posix_strerror_r	-	-
gfal_posix_print_error	-	-
gfal_posix_code_error	-	-
gfal_set_verbose	-	-
gfal_set_vo	-	-
gfal_set_nobdii	-	-
gfal_set_timeout_connect	-	-
gfal_get_timeout_connect	-	-
gfal_set_timeout_sendreceive	-	-
gfal_get_timeout_sendreceive	-	-
gfal_set_timeout_bdii	-	-
gfal_get_timeout_bdii	-	-
gfal_set_timeout_srm	-	-
gfal_get_timeout_srm	-	-
gfal_is_nobdii	-	-