Guildford Protocol

Current Maintainer: Thomas Krichel
This document contains contributions by José Manuel Barrueco Cruz, Sune Karlsson and Ivan Y. Kurmanov

1999–12–27

This version

```
http://openlib.org/acmes/root/docu/guilp\_1999-12-27.html
http://openlib.org/acmes/root/docu/.papers/guilp\_1999-12-27.a4.pdf
http://openlib.org/acmes/root/docu/.papers/guilp\_1999-12-27.letter.pdf
```

Last stable version

```
http://openlib.org/acmes/root/docu/guilp.html
http://openlib.org/acmes/root/docu/.papers/guilp.a4.pdf
http://openlib.org/acmes/root/docu/.papers/guilp.letter.pdf
```

1 Introduction

This document is the Guildford protocol. It is named after the town where its very first version was written. The protocol provides a set of rules for the publication and exchange of documents on the Internet. It could be implemented in any group that wishes to distribute documents on the Internet. It is particularly suitable if the metadata is encoded in ReDIF.

The idea behind the protocol goes back to a statement by William L. Goffe. On 15 July 1995, he wrote on the (now defunct) NetEc-admin list:

> What I would suggest is this: a distributed system with any number of sites, each mirroring each other. It would have extensive bibliographic functions (cross-referencing, etc.), and my favorite, digital timestamps for when the papers were put up. For archives outside it, papers could be listed, but no cross-referencing. But, such archives could "join" the system (say it was written in perl so could run on NT as well as Unix). Then you'd have the best of both worlds: distributed, anybody could join, extensive cross-referencing, the whole works. Such a system could easily grow with the profession's use of the net. Such a system would GREATLY benefit the profession.

The way to achieve this "global and local" archive is through a comprehensive distribution system that is based on a set of archives. An archive is a machine that makes data available. It is a place where original data enters the system. The data are then distributed to any number of sites. A site is a collection of archives on the same computer system. It usually consists of a local archive augmented by frequently updated copies of remote archives. The local archive is maintained on the local computer, whereas the remote archives are maintained on other computers. We call a frequently updated copy of one archive on a remote site a "mirror". There is no need for every site to mirror the complete contents of every archive in the system. Some sites may only mirror bibliographic information rather than the documents. Others may mirror all the files of an archive. Others will mirror only parts of a few archives.

All archives may hold documents and metadata about documents, as well as software that is useful to maintain archives. Everything contained in an archive may be mirrored. For example, if the full text of a paper is in the archive, it may be mirrored. If the archive does not wish the full text to be mirrored, it can store the documents outside the archive.

The Guildford protocol aims to find a set of minimal restrictions on archives such that a global and local system will work. A second key aim of this document is to provide a set of rules such that if they are followed locally, require almost no central effort. However a small amount of work has to be provided by a central archive. This archive is called the ALL archive. It contains ReDIF templates that describe all known archives in the system.

A limitation of this protocol is that it does not deal with limiting access to metadata. As far as the protocol is concerned, the "metadata" is everything that is encoded in ReDIF. The protocol assumes public reading access to files that contain ReDIF. Limiting the access to the resources that are described by the metadata is possible but remains outside of the scope of the protocol.

A second limitation is that the protocol does not deal with archiving and preservation issues. A key feature of the protocol is that each document has exactly one home site. If the home site withdraws the document it is withdrawn (after a short delay) on all sites that maintain copies of the document.

The last limitation is that the protocol is not concerned with providing end user services. For example the protocol does not provide any ideas on how to present documents on a web server, how documents should be indexed etc. However one of the key features of the protocol is that software used to perform these task can be written by a community of contributors and distributed among sites for the benefits of everybody. There may be a variety of software tools written to render the data in all sorts of ways, to prepare all sorts of indexes, fire up a number of search tools etc.

2 Definitions and conventions

The "authority" is a group of people that have come together to implement the Guildford protocol on a set of documents and metadata. For example "RePEc" is an authority that supports the creation and deployment of ReDIF data in economics according to the Guildford protocol. For the rest of the document, we will use the string *authority* to refer to the name of the authority.

Authorities share a common space called the "root" level. The root is based at `ftp://openlib.org/acmes/root`.

"ReDIF (Research Document Information Format)" is a set of data structures to encode information about documents, series of documents, persons, institutions and other aspects of reality that academics are interested in. In ReDIF version 1, these data structures take the form of template types. These templates types are discipline-independent and independent of the organisational structure that supports their creation and deployment.

A "series" is a collection of documents that are kept together.

An "archive" is a directory on a computer that is open to access by ftp or http. It holds a collection of series of documents or a collection of data about documents held elsewhere.

A "*authority* archive" carries data formated in ReDIF that use handles controlled by the authority *authority*.

A "site" is a collection of (normally one) local archive plus any number of mirrored archives. For the purpose of identification, sites and archives are treated identical. Normally each site runs on archive and mirrors several others.

The "*authority* ALL archive" for an authority *authority* is a single machine on which a limited number of important files are kept. These include the documentation of the protocol, including the regulations of the decision making process, brief descriptions of all the software contributed to the organisation and the core templates (see the ReDIF documentation) of all participating archives. The contents of the ALL site is made available on all other sites.

A "service" is a site which offers end-user facilities.

"mirroring" is a process by which copies of series of documents are made from one site to another, such that the contents of the archive is the same on all sites except for a short delay that is inevitable.

3 The structure of an archive

An archive contains a set of files. All names of files are case-insensitive. Each archive is identified by a three letter identifier. All series identifiers have six letters. The reserved file names have four letters. Archive identifiers are awarded by the authority. The series identifiers are fixed by the archives in consultation with the authority, and the reserved words are those mentioned in the protocol. All files ending with the extension `.rdf` (case-insensitive), pronounced "ReDIF" are files that contain ReDIF templates.

If an archive runs on a multi-user machine, it is recommended to create a special account *archive_account* for the archive. The files that make up the archive can then live in the home directory of the user *archive_account*, or better, in a subdirectory *authority* of this account that points to a space in the files system that is accessible via anonymous ftp or http. In the following we assume that we are in this subdirectory and we are looking at the files and directories that it contains. We use the Unix convention to call the current directory `.` and to separate the

`./`*`archive_id`*`/`*`archive_id`*`arch.rdf` a file describing the archive using a single ReDIF archive template. (mandatory)

`./`*`archive_id`*`/`*`archive_id`*`seri.rdf` a file describing the series in the archive using a sequence of ReDIF series templates. All series in the archive are described in this file, one template for each series. (mandatory)

`./`*`archive_id`*` /`*`archive_id`*` mirr.rdf` a file describing the mirroring arrangement of the site, using the ReDIF-mirror template.

`./`*`archive_id`*` /`*`series_id`*` /` a directory for papers and metadata for the series that is identified by *series_id* are stored. All files that that pertain to the series *series_id* must be stored in that directory. Files that contain ReDIF data are called ReDIF files. Their names must end in `.rdf`, but otherwise the structure of the directory is free. You may put all templates for all documents in the series in one file or you may put each template in a different file, just do as you like. It is good practice to start each ReDIF file in the directory with the *series_id*.

`./`*`archive_id`*` /pers/` a directory for person templates, only to be created if such templates are to be supplied locally, which is possible but not recommended.

`./`*`archive_id`*` /inst/` a directory for institution templates, only to be created if such templates are to be supplied locally, which is possible but not recommended.

`./`*`archive_id`*` /soft/` location software that is written locally. For example, an archive may wish to write a specific procedure by which its ReDIF template is translated into html.

`./`*`archive_id`*` /conf/` location of the configuration files for software. It does not matter whether the software is supplied by the local archive or a remote archive.

For any archive, the collection of `./`*`archive_id`*` /`*`archive_id????`*` .rdf` where *????* is either `arch`, `seri` or `mirr` are called its core templates. They are mirrored on the ALL archive.

If you are only interested in opening an archive to provide data to an authority, there is no need for you to read any further. However if you wish to provide software to be shared across other users, please consider Section 7.

4 The structure of a site

A site is at least one local archive augmented by possibly several remote archives. Each site may mirror a number of series from any number of archives. If any site mirrors any series for an archive, it may mirror the complete subdirectories of the series or all ReDIF files (ending with `.rdf`) of the series. If the maintainers of an archive do not wish the documents to be mirrored, then they will store them outside the archive.

The structure of the site is the same as the structure of the local archive. The only element that is added is a location for the remote archives.

`./remo/`*`archive_id`*` /` This is a strong suggestion (in the sense that some software may not work if you to not follow it) where to put data from a remote archive *archive_id*. A site does not need to mirror all the files from a remote archive. It must mirror the archive templates of the remote archive. It may then select series to mirror, or only mirror the ReDIF files.

5 The structure of the ALL archive

For each authority, there is a central archive that stores the core templates that describe all the archives that are identified by the authority. This archive is the ALL archive.

In the case of RePEc, the ALL archive operates at ftp://netec.mcc.ac.uk/pub/RePEc/all. Its archive code is ALL. On the root directory of the ALL archive, you will find *???* `arch.rdf`, *???* `seri.rdf` and *???* `mirr.rdf` files for every archive that is known to the authority. In addition there are the following directories.

`./all/soft/` location of software that is provided by the ALL archive.

`./all/docu/` location of any documentation that is provided by the ALL archive.

`./all/root/` a mirrored copy of the root directory from the cross-authority root level http://openlib.org/acmes/root. The contents of that directory is described in the next section.

6 The structure of `ftp://openlib.org/acmes/root`

This site hosts the cross authority root level.

`ftp://openlib.org/acmes/root/auto/`

the directory that contains the authority templates of each authority.

`ftp://openlib.org/acmes/root/docu` the directory where the documentation is stored that is shared by all authorities. This includes the Guildford protocol and the ReDIF documentation.

`ftp://openlib.org/acmes/root/soft` the directory where the software is stored that is shared by all authorities. This includes the Guildford protocol compatible mirror script and the ReDIF parsing software.

`ftp://openlib.org/acmes/root/conf` the directory that contains the specification of ReDIF. It should not contain any other files.

7 Providing software

A software packages is a set of software files that together perform a particular function. Archives may provide software for various sites. Services are encouraged to provide software that allows the construction of identical services on many sites.
Currently all our software is written in Perl. All scripts assume that Perl lives in `/usr/bin/perl` to allow direct use. Thus all scripts start with `#!/usr/bin/perl`.
For the sake of simplicity of exposition, let us assume that there is an archive "giv" that gives (provides) a software package called "bambi". This software is provided in the `./giv/soft` directory. Software written at the local site will be found in the `./giv/soft` directory.
The file `./giv/soft/bambi` is the main executable. Names of other files start with 'bambi' (e.g. `bambi.README`) or should be placed in a directory `./giv/soft/bambi` in order to distinguish which file belongs to which packages. Scripts used by various programmes are placed in `./giv/soft/all`.
All scripts at start perform some general procedures to ensure their portability between different sites and environments. They first check for command-line options `-rdir` *`directory_name`* and `-conf` *`filename`*) to allow users to set the archive directory (which ends with archive identifier) and to chose a programme configuration file name (instead of the default), respectively. The latter is not necessary if software does not use a configuration file.
If the `-rdir` option is not given, then the programme checks the environment variable `REDIFDIR` for the archive home directory. The programme checks the environment variable `AUTHORITY` to find the authority it is working for. If it is not set, then it assumes that current directory is the archive directory. Keep in mind that some Perl functions in your software may require the absolute path of the archive directory, thus a simple `./` may be too simple.
If the archive home directory was received through a command-line option or as environment variable, it should be checked whether such a directory exists at all, and whether it ends with a three letter sequence, which is supposed to be the local archive identifier. Under Unix the software should be aware of the optional trailing backslash.
In ftp://openlib.org/acmes/root/soft/guilp_use_perl.eg we show an example of Perl code to implement this algorithm. It was written by Ivan Kurmanov. You are welcome to adapt it in your programmes. Ivan welcomes comments and suggestions.