Database archiving with dbDIPview: a brief overview

Boris Domajnko (Archives of the Republic of Slovenia, Ljubljana)

Boris Domajnko (M.Sc. of Computer Science) has been working at the Department of electronic archives and IT support since 2008. From the beginning, he was intrigued by the problem of preserving databases and making them accessible. He has used his experience in areas of software engineering from his previous work in the private sector and developed a solution that grew into a full production process.

dbDIPview (an acronym for database DIP viewer) is a tool and approach for long-term archiving of databases. It helps us make their information easily accessible for a non-technical user, be it an archivist or a reading room visitor. Domajnko developed it for the first database ingest in 2010 and continues to improve it whenever a new production scenario for ingest or access emerges. The archiving projects have been done with public registries based on dBase, its successors, and also with data from modern DBMSs. The tools are used both for ingest and for access scenarios.

Generally, in AIP, a content information package consists of a data object and representation description information. dbDIPview is used for creating, packing, and using the latter. The data object remains unchanged and can preserve its authenticity. It contains SIARD or CSV files or their combinations.

In the pre-ingest phase, a SIP with a data object is transformed into a preliminary DIP on a dbDIPview server. The archivist now configures the viewer details and examines the content, and finally rejects or accepts the SIP. At this stage, database knowledge is still needed as we try to imitate the search and reporting functionality of the original application by preparing expected use cases in XML. Therefore, the output of old and new applications will be compared.

With this approach, three risks are covered by ensuring the following:

- SIP content is checked before the archive confirms the ingest,
- the access is possible when the pre-ingest is done, and
- the documentation received is adequate.

In the end, the viewer configuration is ready, and the representation information object is built with a single command. This object will accompany the data object. An AIP can be set up and ingested. It is now available for automated deployment.

When an end user requests access to data, the archivists can deploy the database with a few commands in the admin menu, or in an automated way, based on an “order” XML file with needed information, including possible redaction. The end user can then get and use the data ordered – not as a database administrator, but as a user of an application that mimics output of the original application. Nested tree view menus are used for searching and pre-configured hyperlinks between reports are possible for further drilling down on data. Additional on-screen descriptions help to understand the data shown.

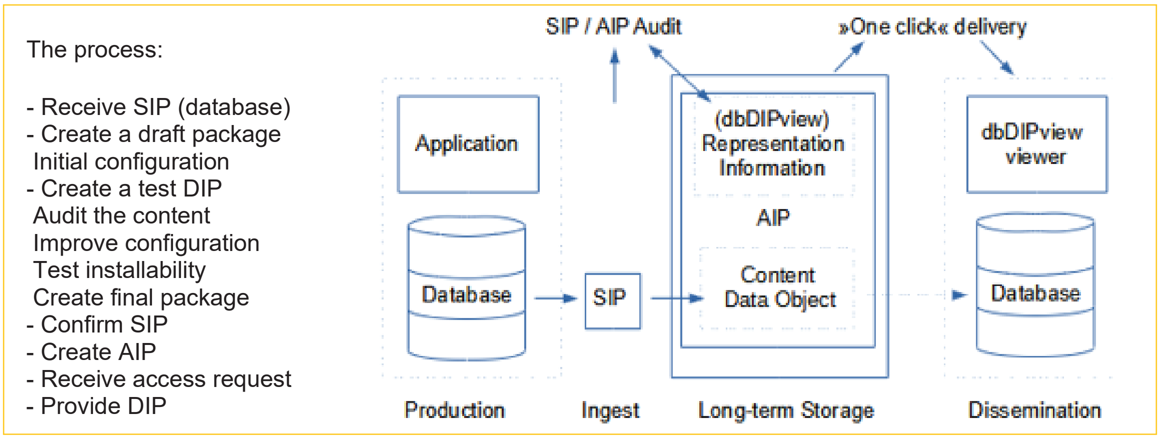
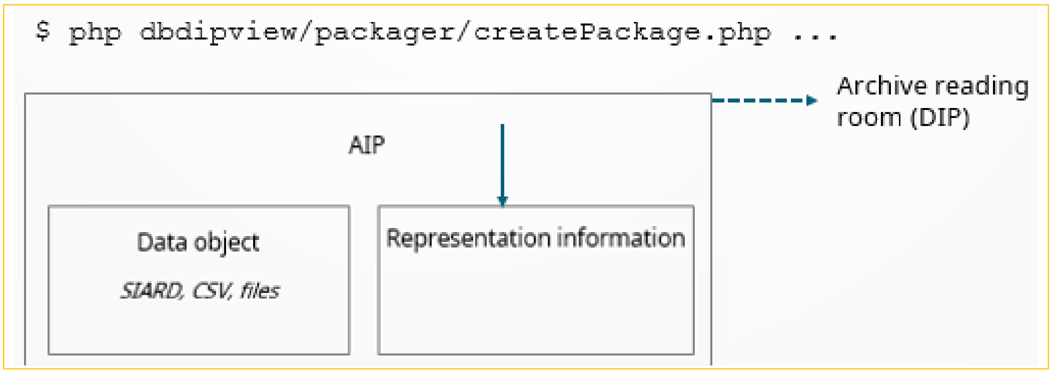


Figure 37: A risk minimizing process steps for DB archiving as proposed by Domajnko

Command line tools are used to create, install and uninstall packages. This example will validate the files and create a package that can contain CSV files or use external SIARD files:

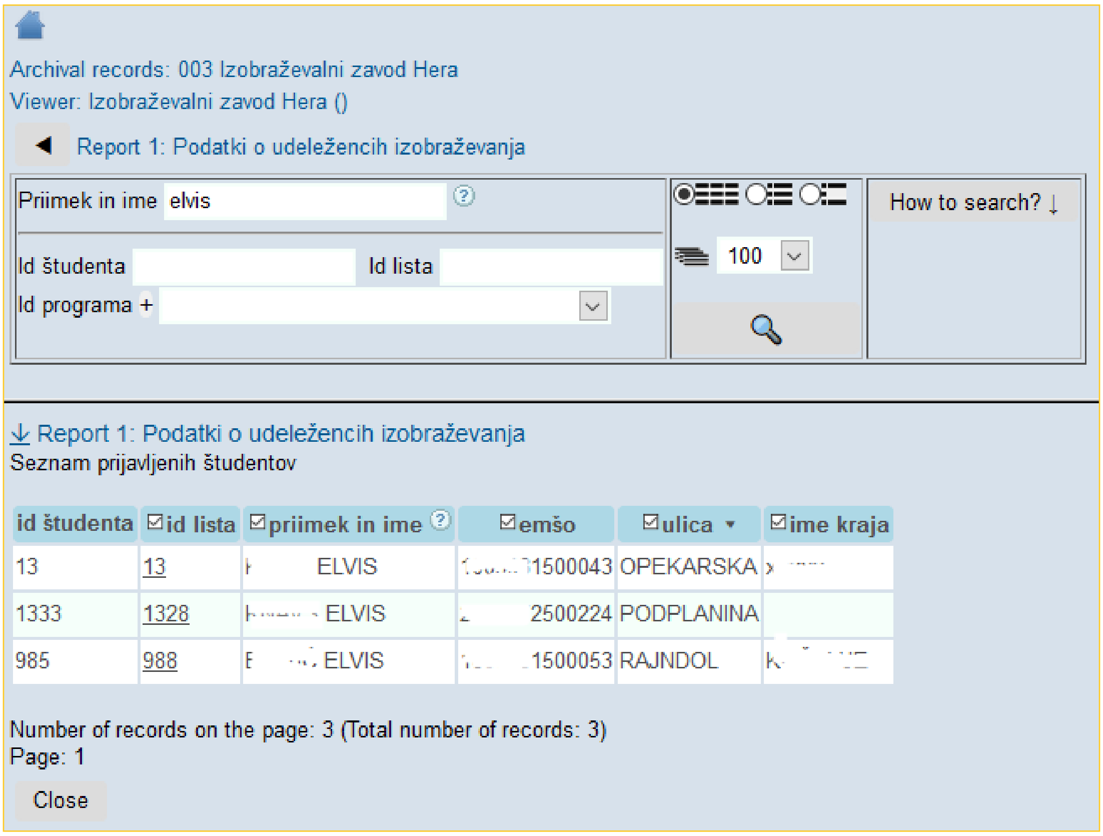


```
$ php dbdipview/packager/createPackage.php ...
```

Figure 38: The dbDIPview command for creating an AIP with representation information

The usability of displayed data can further be increased by allowing for longer text in the headers to explain the details about a certain table or report. External attachments and BLOB content can be shown. The solution is targeting end users who need friendly easy-to-use advanced search functionality. Future researchers could still get direct access to the tables, e.g. for statistical analysis. Nevertheless, the typical use of dbDIPview is getting an unambiguous concrete piece of information.

Every database ingest and use is a bit different. However, all required scenarios have been successfully covered so far.



Archival records: 003 Izobraževalni zavod Hera
Viewer: Izobraževalni zavod Hera ()

Report 1: Podatki o udeležencih izobraževanja

Priimek in ime: elvis
Id študenta:
Id lista:
Id programa +:

100

Report 1: Podatki o udeležencih izobraževanja
Seznam prijavljenih študentov

id študenta	☑id lista	☑priimek in ime	☑emšo	☑ulica ▾	☑ime kraja
13	13	ELVIS	1500043	OPEKARSKA	
1333	1328	ELVIS	2500224	PODPLANINA	
985	988	ELVIS	1500053	RAJNDOL	

Number of records on the page: 3 (Total number of records: 3)
Page: 1

Figure 39: A typical dbDIPview user interface screen with hyperlinks

Questions and discussion

- Is dbDIPview open source? – Yes, it is on GitHub: https://github.com/dbdipview/dbdipview/wiki, and our institution can also be contacted in case any additional information is needed.
- Raja Appuswamy asked if a SIARD reader is the right tool – couldn't you move the SIARD file to a common database format instead of using the dbDIPview framework? This would allow use of all kinds of analysis tools that work on databases. Boris Domajnko replied that authenticity is also important. The typical archival use case is someone addressing the archive and asking for a document. The archivist must be able to use the authentic DB data and find the right information at once, without extra time for tackling the tables. People need a clear output from the tool. Of course, researchers who want to make in-depth studies can get the SIARD package and perform their analyses on any platform they desire.
- Torbjørn Aasen noted that it is important to work out what should be kept and what should be thrown away. It should be the aim to describe as much as feasible so users can later analyse it.
- James Doig remarked that resourcing is an issue. A normalisation approach such as SIARD standardises format makes life much easier for an archive going forward.