Sanity checks in Aleph

(Filtering Bibliographic and Item Records in Aleph with Cinege)

Nagy, Elemér Károly
eknagy@omikk.bme.hu
Krauthausen, Leon
krauthausen@ub.fu-berlin.de

IGELU 2006 Stockholm

1 minute

Definitions

Wikipedia (http://www.wikipedia.org):

- ***Aleph:*** *Aleph, an integrated library system from Ex Libris LTD used to manage library operations ...*
- ***Cinege:*** *Cinege is a GPL-ed (free), platform independent package of tools ...*
- ***Filter:*** *A filter is a computer program to process a data stream.*
- ***Bibliographic record:*** *no page with that title exists*
- ***Item record:*** *no page with that title exists*
- *So, I will talk about processing ???s that are originated from a commercial ILS (LMS), with some free and platform independent tools.*

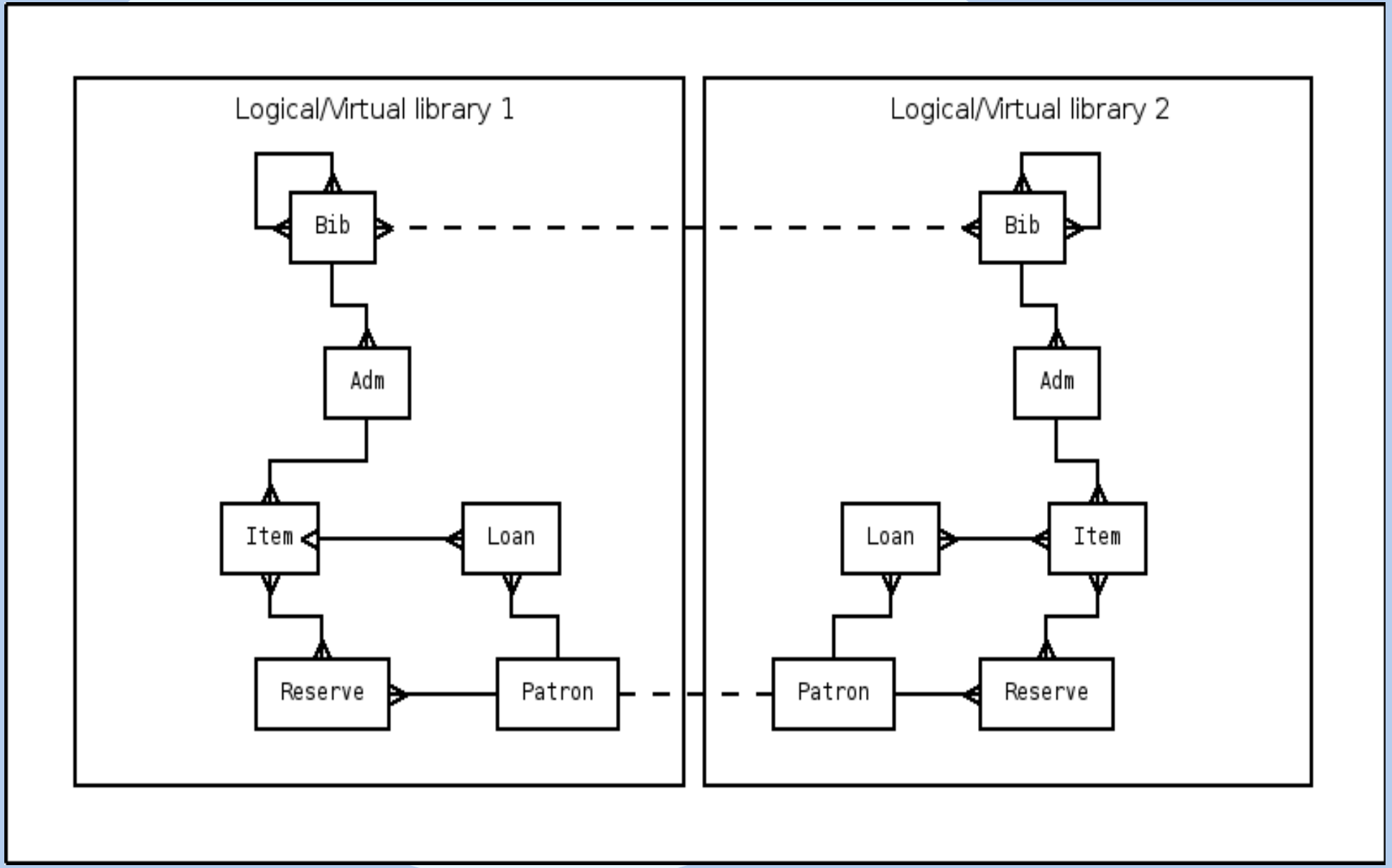
1 minute

Comments on my person

- ***But who I am to talk about such serious issues?***
- ***I am the developer of Cinege and doing my PhD in IT - please feel free to ask deep technical questions.***
- ***I have been working with Aleph since 2002 and produced a few hundred error lists with different scripts/programs.***
- ***I converted all our Bib/Item records for various purposes a few times - sometimes heavily modifying them.***
- ***But even if I am working in a library, I am not a librarian - please don't be angry if I use "news" instead of "serial".***
- ***I will talk about filtering Bib, Adm and Item in general and will show Cinege examples from BUTE and FUB.***

0.5 minute

These ???s we talk about



0.5 minute

Bibliographic records (Z00)

- ***Contains most of the information that is common about groups of items (title, author, location, ...)***
- ***Needs a very flexible format (Example: a book with 13 authors, with 3 primary authors, 4 secondary authors, 3 editors, 2 deputy editors and 1 translator)***
- ***There are special cases (multi-volume books, conference proceedings, electronic books, holding records, authority records, audiovisual books, books with CDs, ...)***
- ***Cross-references between records, libraries, different types of records***
- ***Different practice in different libraries lead to the need of different tools to handle records***

0.5 minute

Item records (Z30)

- ***Contains most of the information that may be different it different physical items (barcode, call number, location, protection mark, volume, ...)***
- ***A pretty inflexible format with a fixed number of fields and a fixed format of the fields hard-coded***
- ***On the other hand, the values of the fields are locally defined just as the connection between the fields***
- ***Because of the fixed format, some fields are used to store something else than it was designed for, e.g., call_no is translated to „Shelf code" and call_no_2 is translated to „Call number"***
- ***Different practice in different libraries lead to the need of different tools to handle records***

0.5 minute

Administrative records (Z103)

- ***Links Item and Bib records***
- ***May contain the information that is common about a smaller groups of items (volume, location, protection mark) - but not in Aleph.***
- ***Needs a flexible format***
- ***Some Bib records have no Item records***
- ***Some Bib records have no Adm records***
- ***Z103 is (re)generated from the BIB records - so it is only a cache - altough very useful***

1 minute

Main problems with these records

- Some information (location, protection mark, volume) can be stored in different places (Bib or Item), thus we once more see different practices.
- Some rules may be implemented in local regulations only, without software implementation (if *z30_collection* is ***'Closed stack'*** then *z30_item_status* should not be '01'). In some cases, Aleph implementation of such rules is practically impossible.
- Some (circular and one-to-many) links may be very hard to handle forth and back (if there is an item with a *status* of *92*, then amongst the items that are linked to the same Bib record, there can be no item with a *status* of *23*).

1 minute

Error sources I.

- Personnel errors:
  - in data entry
  - in data modification
  - in data deletion
- Import errors:
  - character set errors
  - missing fields
  - extra fields
  - fields with invalid format
  - fields with invalid value

1 minute

Error sources II.

- Data is imported:
  - from external sources
  - when upgrading Aleph
  - when upgrading Oracle
  - when upgrading OS
- Data format is possibly changed:
  - by new standards
  - by new media types
  - by new softwares
  - by new personnel
  - by new technologies

1 minute

Life of a database

- Let us assume we:
  - had an empty dBASE / ASCII database in 1975
  - migrated 500K records to FoxPro / Latin2 in 1985
  - migrated 1000K records to Aleph300 / Latin2 in 1995
  - migrated 1500K records to Aleph500 / Latin2 in 2000
  - migrated 2000K records to Aleph500 / UTF-8 in 2005

- So, if we hadn't upgraded (or changed) the DBMS, the OS or the ILS, except when migrating, then 25% of our records are converted 4 times, 25% is converted 3 times, etc. ...

0.5 minute

Life of a database

- Let us assume we:
  - had an empty dBASE / ASCII database in 1975
  - migrated 500K records to FoxPro / Latin2 in 1985
  - migrated 1000K records to Aleph300 / Latin2 in 1995
  - migrated 1500K records to Aleph500 / Latin2 in 2000
  - migrated 2000K records to Aleph500 / UTF-8 in 2005

- So, if we hadn't upgraded (or changed) the DBMS, the OS or the ILS, except when migrating, then 25% of our records are converted 4 times, 25% is converted 3 times, etc. ...

- But we did, didn't we?

1.5 minute

Problems with migration

- When we migrate data, the data format is changed. (Anybody has any counter-examples?)
- When the data format is changed, data must be transformed to fit into the new format.
- If the input records is **flawless**, **and** the conversion is perfect, then the output records are good enough.
- If the conversion is not perfect, then even in the best case, the output record will be faulty.
- If the input record is not flawless, then even in the best case, the output record will be faulty.
- If the input record is not flawless and the conversion is not perfect (reality), we are in serious trouble.

0 minute

Records with errors

- So we have 3% (0.1% to 10%) records with errors. So?

1 minute

Records with errors

- So we have 3% (0.1% to 10%) records with errors. So?
- **A record with error might not be found.**
- Example: if query ends with *language is 'hun'*, you won't find records with *language being 'hu' or 'hum' or 'magyar' or 'hungarian' or 'enggehun'*)
- A less precise query (“language like '%hun%') might find it, but is much slower and may find records with *language* being 'zahune' (written in 'zah' and 'une' languages).
- When we convert 'hu', the output record may contain the same error, may be without a language code, or may be a record wreck (if conversion expected it to be 3 characters long) or may fail (on that record or entirelly).

1 minute

Sidenote: Offensive and Defensive programming

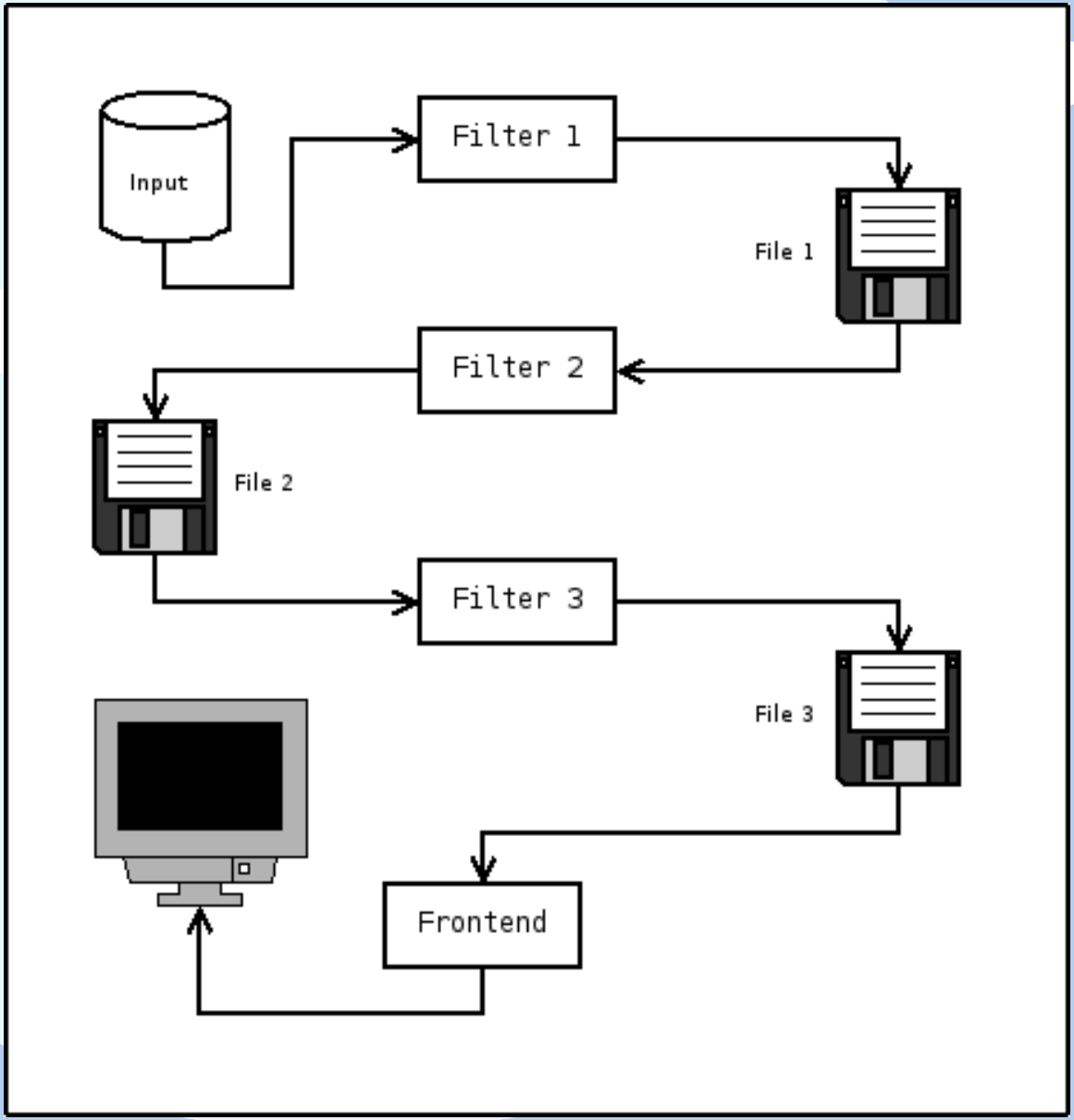
- Defensive programming (defensive error handling) tries to work-around or repair errors, concealing it as much as possible. This may be great in video conference software.
- Agressive programming (agressive error handling) tries to produce the most serious possible error. This may be great in accounting software.
- More defensive error handling typically means more user friendly, more convenient and a more marketable product.
- More agressive error handling typically means more error messages, more user problems and a cleaner database.
- In this view, defensive is a short-term approach and aggressive is a long-term approach.
- IMHO, converters and filters should be agressive.

Converters and filters

- Converters are programs that modify some or all of the records, treating all of them the same way.
- Filters are programs that drop some or all of the records, thus treating records in different ways.
- The simplest way to differentiate them is to refer to their function - a program that is used to filter records is a filter, even if it was created as a converter but omits records with errors. A program that was created as a filter, but adds fields as a side-effect can be used as a converter.
- It is very useful, if a group of converters and filters can be pipelined because the output of some of these programs is a suitable input for some of these programs.

1 minute

Pipeline diagram



1 minute

Sidenote: Programming languages I

- Script languages (AWK, VB, PERL) are very fast to develop - but often referrer to as „write only” languages, modifications are hard to implement and there are often problems with typical plain-text input/output files. Ideal for „run exactly once” and „never to be modified” programs.
- Traditional programming languages (C, Fortran, Cobol) are much slower to develop but provide a very rich set of libraries and programs written in them are easier to maintain. Branching of code and lost source code are still problems.
- Object-oriented programming languages (C++, Java, Phyton) are slower to learn but are much easier to maintain, as only some components needs to be changed/replaced. Ideal for complex and/or slowly (but steadily) changing programs.

1 minute

Sidenote: Programming languages II

- Programming languages may be compiled, interpreted or hybrid languages (categories may interlap).
- Interpreted languages are slower, more platform-independent and the source code typically can be regenerated from the running program relatively easily (companies don't like this).
- Compiled languages are faster by magnitudes and source code is typically much harder to regenerate, once source code became unavailable (lost code = practically unmodifiable program).
- Hybrid languages include PERL and Java, both having a compilation to a binary form which is then interpreted, being efficient enough and keeping source code in reach.

1 minute

Sidenote: Regular expressions

- PERL and Java both support regular expressions.
- Regular expressions are very dense pattern definitions that can be used to formulate complex search criteria.
- Built-in parsers and matchers are well tested and fast - no need to reinvent the wheel.
- Not easy to read
- Examples:

^.*$	anything
^.*this.*$	anything containing “this”
^this$	this (without anything behind or after it)
^.*[a-z]{3,6}$	anything that ends with 3 to 6 of the 24 latin1 lc. letters
^[^].*$	not starting with space
^.*[^0-9]$	not ending with a digit
^[0-9]{4,4}.+$	four digits and at least one character after it
^(19[5-9][0-9])|(200[0-6])$	any four-digit number between 1950 and 2006

1 minute

Cinege I.

- A GPL-ed librarian data filtering command-line tool package written in Java, that uses Mysql.
- GPL-ed means it is free (free as in free speech and also free as in free beer)
- Java means it will run on Windows, Linux, Aix, Solaris, ...
- Command-line means it can be run in scripts at night, used in web applications or on headless machines (mainframes)
- Using MySql means that it can use a local database or use a remote database or use several remote database or use combinations of the above on almost any platform (more than 16 operating systems supported).

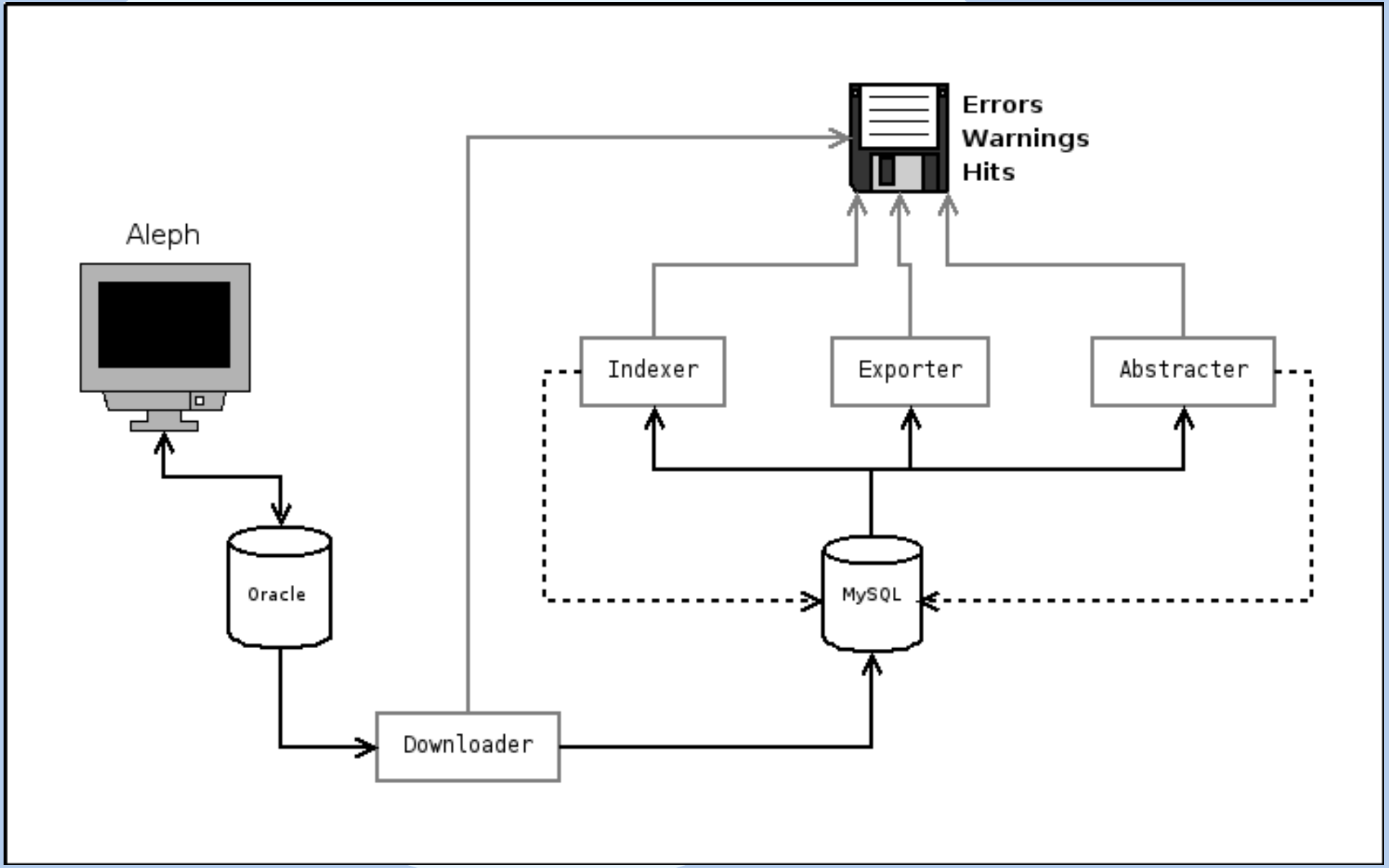
1 minute

Cinege I.

- A GPL-ed librarian data filtering command-line tool package written in Java, that uses Mysql.
- GPL-ed means it is free (free as in free speech and also free as in free beer)
- Java means it will run on Windows, Linux, Aix, Solaris, ...
- Command-line means it can be run in scripts at night, used in web applications or on headless machines (mainframes)
- Using MySql means that it can use a local database or use a remote database or use several remote database or use combinations of the above on almost any platform (more than 16 operating systems supported).
- I have a Cinege with complete snapshots of BUTE Adm, Bib and Item data - on this laptop.

1 minute

Cinege II.



2 minute

Lists that can be made with Cinege

- Sanity checks, containing records that match or does not match given criteria.

  Example: Find books with two or more primary titles.

- Working lists used in different reorganizations, containing all kinds of Item and/or Bib information.

  Example: Find books in English, German or French to support creation of a new EU reading room.

- Daily statistics containing agglomerated data.

  Example: Give loaning statistics (min, max, avg) on books in EU reading room and give title of most popular items.

- Decision support lists.

  Example: Create list of patrons who lost at least two books with a value of at least 10 Euros and have at least 20 Euros total fine.

1 minute

On-line vs. off-line

- On-line queries are very useful, for example, in a case when someone is repairing the records and want to see the current records with error by pressing a force reload.
- Some people like to have error lists in batches, for example, as auto-mailed excel files in the morning.
- Off-line queries are more close to the data warehouse approach and often called reports - very handy in controlling staff and efficiency, not to mention upgrade preparation or upgrade testing.
- In some cases, some queries are so slow, that on-line is not an option (Bibliographic sanity checks).
- In some cases, we can chose which one to use - in some cases, we can use off-line only.

Online sanity checks with formgen

- Online sanity checks can be useful in all those cases where they can be achieved by simple, fast running SQL queries.
- In many cases, specially regarding checks on item records, only staff members (of a specific sublibrary) can decide what fields and what dependencies need to be checked.
- Embedding SQL scripts in PHP using the formgen package provides staff members (with no direct access to the server) with the ability to execute sanity check queries on their own.

Online sanity checks with formgen

- The output of the queries can be produced as HTML or Excel lists or as CSV files.
- Macro programming can provide a easy way to search and load a specific record in an Aleph module by just highlighting a barcode or an ID and pressing a hotkey.

Example of an online sanity check

- Check collection codes for invalid item status.
- Find item records with a code for a closed stack collection "MAG-G" and an (invalid) item status of "01" for open shelf loan.
- Use of an AutoHotkey macro to load records in the GUI-module.

Example of an online sanity check

Step 1: List collection and item status

SQL Reports

List of item collection codes

ADM:	FUB51
Sublibrary:	808
Evaluate item status?	Yes
Output:	HTML
	Starten

SQL Reports

List of item collection codes

Collection	Item Status	Record count
		2
	03	147
	05	68
	06	1
	07	5

[...]

MAG-G	01	20
MAG-G	02	43158
MAG-G	03	21396
MAG-G	04	55
MAG-G	05	104

20 records need to be corrected

Example of an online sanity check

Step 2: Find records with collection code "MAG-G" and item status "01".

SQL Reports

Item records

ADM:	FUB51
Sublibrary:	808
Call number:	
Material:	
Item status:	01
Item process status:	
Collection:	MAG-G
All note fields:	
Sort:	Call number
Encoding:	LATIN1
Output format:	HTML
	Start

Call number	Barcode	Title	OPAC Note	Circulation note	Internal note	Material type	Collection	Item status
CE 410/77 <6>	ZZ-015884100010	Kapellmann, Klaus D. : Einführung in die VOB, B.- Düsseldorf: 1997				BOOK	MAG-G	01
KL 420/63	ZZ-017190800020	Böhmer, Friederike : Die Ermächtigung zu militärischer Gewaltanwendung durch den Sicherheitsrat.- Baden-Baden: 1997				BOOK	MAG-G	01
LA 678/18	546584188	Voigt, Christian : Die Haftungsfolgen im Staatshaftungsrecht im Vergleich zu den Haftungsfolgen im Zivilrecht.- : 2001				BOOK	MAG-G	01
M III a 450	ZZ-011909500010	Ockenfels, Wolfgang :				BOOK	MAG-G	01

With the help of a special macro (in this case an AutoHotkey macro), the highlighted barcode can be used to search for the record in GUI module.

Example of an online sanity check

- If the number of records that need to be corrected is too large to do so manually, the record IDs of the query result can be saved automaticaly in a file in the ../alephe/scratch directory.
- This file can than be used as an input file for the p-manage-62 service supplied with Aleph.

Example of an online sanity check

Find records with any call number that have an "in process" status (GG)

SQL Reports

Item records

ADM:	FUB51
Sublibrary:	808
Call number:	%
Material:	
Item status:	
Item process status:	GG
Collection:	
All note fields:	
Sort:	Call number
Encoding:	LATIN1
Output format:	SYSID

Start

SQL Reports

Item records

Call number: %...
Item process status: GG

Z30_REC_KEY
002908142000020
000034702000130
002835408000010
002846631000010

[...]

002775059000010
002839945000020
002842275000010
002904650000030
000096372000040

Record keys were saved in:
/exlibris/aleph/u16_1/alephe/scratch/fub51_808_0901.lst.

1 minutes

Sanity check example I.

- Specification: Because a new EU reading room is to be opened soon, we need to get a list of candidate items.
- All items should be written in English, German or French.
- The language codes are stored in the Bib record, in field *041*, in subfield *a and are stored in three-letter language codes (**ISO 3166-1 alpha-3**).*
- *The list should contain the bibliographic system numbers, the titles (245/a), the city if publication (260/a) and the language codes (041/a).*
- *Later, more specialised lists with item number, number of loans, etc. will be required.*

0.5 minutes

Sanity check example II.

- *Trivial solution to get Bib numbers:*

```
java Exporter DB2BIB:b1.csv:IFANYMATCHES#\
NO_FILTER#041#a#^.*((eng)|(ger)|(fre)).*$
```

- *This founds all the records we need, right?*

1 minute

Sanity check example II.

- *Trivial solution to get Bib numbers:*

```
java Exporter DB2BIB:b1.csv:IFANYMATCHES#\
NO_FILTER#041#a#^.*((eng)|(ger)|(fre)).*$
```

- *This founds all the records we need, right?*
- ***Wrong. It does not find any records with errors:***
  - ***No 041/a field in the bib records.***
  - ***041/a field contains uppercase letters.***
  - ***041/a field contains whitespaces, commas, etc.***
  - ***041/a field contains two-letter codes (ISO 3166-1 alpha-2)***
  - ***041/a field contains language names ('english')***

3 minutes

Sanity check example III.

- *To get records with no 041/a:*

```
java Exporter DB2BIB:w1.csv:\
IFMULTIPLICITYCHECKSUCCEEDS#041#.*#a#false#0#0
```

- *To get records with at least one 041/a:*

```
java Exporter DB2BIB:c1.csv:\
IFMULTIPLICITYCHECKFAILS#041#.*#a#false#0#0
```

- *To get records with valid 041/a:*

```
java Exporter DB2BIB:w1.csv:IFNOTALLMATCHES\
#NO_FILTER#041#a#^(([a-zA-Z]){3,3})+$
```

- *To get records with language codes mentioned:*

```
java Exporter DB2BIB:g1.csv:IFANYMATCHES#\
POST_ALPHANUMERIC#041#a#\
^.*((eng)|(ger)|(fre)|(ENG)|(GER)|(FRE)).*$
```

2 minutes

Sanity check example IV.

- *To get the first titles, publishing cities and language codes for the BibSysNos, we can:*

```
java Exporter BIB2LIST:g1.csv:\
FIRST#245#a:FIRST#260#a:FIRST#041#a
```

- *To get the first titles and publishing cities and all the language codes for the BibSysNos we can:*

```
java Exporter BIB2LIST:g1.csv:\
FIRST#245#a:FIRST#260#a:ALL#041#a
```

- ***There are plenty (9 directives, 14 subdirectives) of other possibilities - they are evaluated in Exporter.pdf in the data/cinege/docs/ directory.***
- ***These filters are designed to be used in pipeline, like DB2BIB => BIB2ADM => Adm2LoanNumber***

Sanity check example V.

- *So, the final, ultimate solutions that also generates two error lists as a side effect:*
- `DB2BIB:w1.csv:IFMULTIPLICITYCHECKSUCCEEDS#041#.*#a#false#0#0`
  `BIB2LIST:w1.csv:FIRST#245#a:FIRST#260#a:ALL#041#`
- `DB2BIB:w2.csv:IFMULTIPLICITYCHECKFAILS#041#.*#a#false#0#0:\`
`IFNOTALLMATCHES#NO_FILTER#041#a#^(([a-zA-Z]){3,3})+$`
  `BIB2LIST:w2.csv:FIRST#245#a:FIRST#260#a:ALL#041#a`
- `DB2BIB:g1.csv:IFMULTIPLICITYCHECKFAILS#041#.*#a#false#0#0:\`
`IFALLMATCHES#NO_FILTER#041#a#^(([a-zA-Z]){3,3})+$:\`
`IFANYMATCHES#POST_ALPHANUMERIC#041#a#\`
`^.*((eng)|(ger)|(fre)|(ENG)|(GER)|(FRE)).*$`
  `BIB2LIST:g1.csv:FIRST#245#a:FIRST#260#a:ALL#041#a`
- *It might look frightening at first.*

Sanity check example V.

- *So, the final, ultimate solutions that also generates two error lists as a side effect:*
- `DB2BIB:w1.csv:IFMULTIPLICITYCHECKSUCCEEDS#041#.*#a#false#0#0`
  `BIB2LIST:w1.csv:FIRST#245#a:FIRST#260#a:ALL#041#`
- `DB2BIB:w2.csv:IFMULTIPLICITYCHECKFAILS#041#.*#a#false#0#0:\`
`IFNOTALLMATCHES#NO_FILTER#041#a#^([a-zA-Z]){3,3})+$`
  `BIB2LIST:w2.csv:FIRST#245#a:FIRST#260#a:ALL#041#a`
- `DB2BIB:g1.csv:IFMULTIPLICITYCHECKFAILS#041#.*#a#false#0#0:\`
`IFALLMATCHES#NO_FILTER#041#a#^([a-zA-Z]){3,3})+$:\`
`IFANYMATCHES#POST_ALPHANUMERIC#041#a#\`
`^.*((eng)|(ger)|(fre)|(ENG)|(GER)|(FRE)).*$`
  `BIB2LIST:g1.csv:FIRST#245#a:FIRST#260#a:ALL#041#a`
- *It might look frightening at first, but it is easy to get used to it - there are repeating patterns inside.*
- *Commercial support is also available on decent prices, if you want to remain frightened.*

1 minutes

Sanity check example VI.

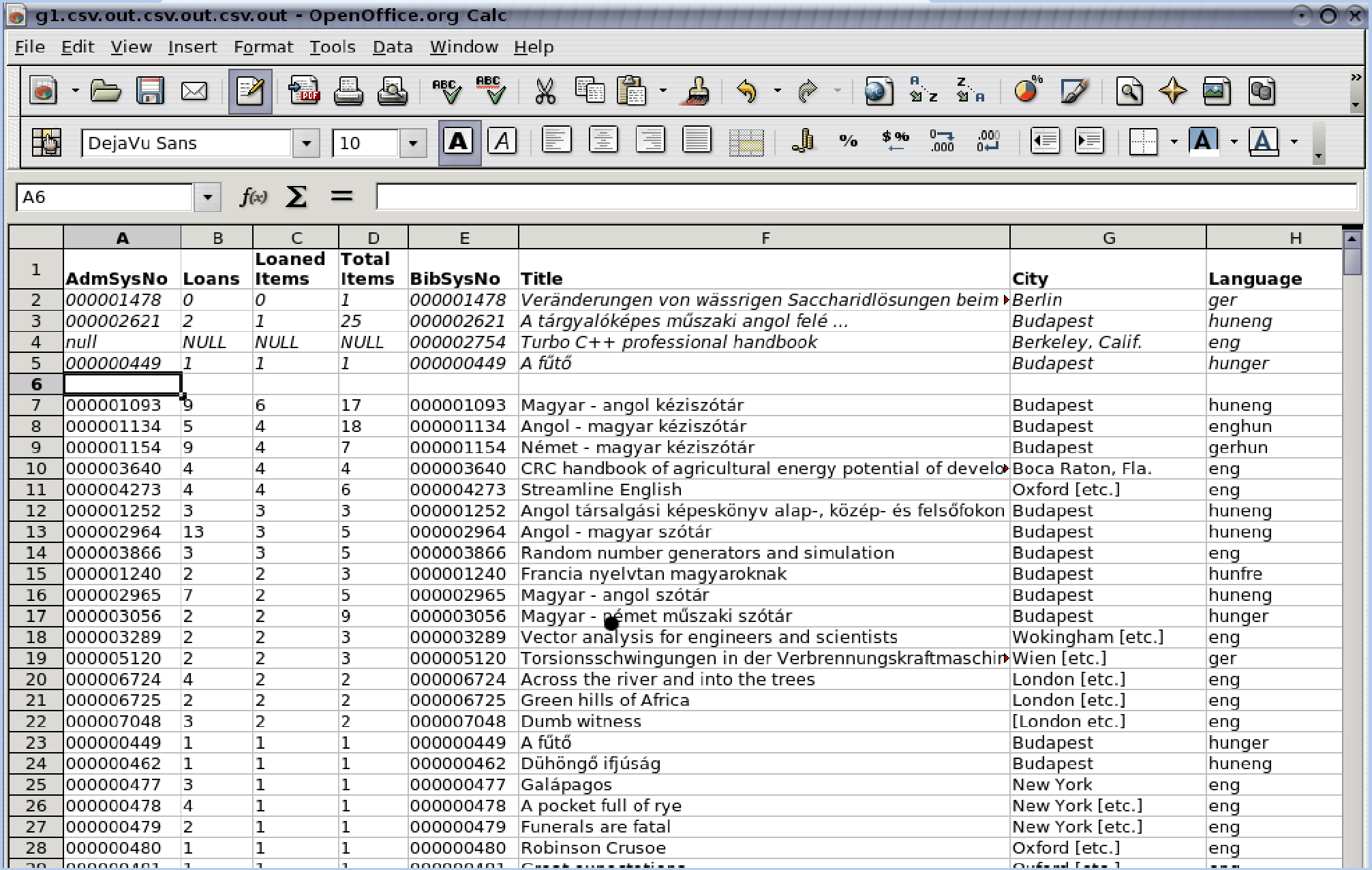
- Later, management decides that we only put books that have been loaned to this new reading room. So, how can we get the number of loans?
- Simple solution: Z36, Z36H, group by, count
- Problem: If we have 10 items and 22 loans, we do not know if all 10 items were loaned - maybe only 5 items were loaned. So, the SQL query gets complicated (two tables to merge, *having clause*, ...).
- Alternatively, we do it with cinege:

```
BIB2ADM:g1.csv.out.csv
ADM2LoanNumber:g1.csv.out.csv.out.csv:200501010000:200512312400
```

- *That's all. It's done. Output list can be loaded to Excel.*
- *Many other directives to use in the documentation.*

1 minutes

Sanity check example VII.



	A	B	C	D	E	F	G	H
1	**AdmSysNo**	**Loans**	**Loaned Items**	**Total Items**	**BibSysNo**	**Title**	**City**	**Language**
2	*000001478*	*0*	*0*	*1*	*000001478*	*Veränderungen von wässrigen Saccharidlösungen beim*	*Berlin*	*ger*
3	*000002621*	*2*	*1*	*25*	*000002621*	*A tárgyalóképes műszaki angol felé ...*	*Budapest*	*huneng*
4	*null*	*NULL*	*NULL*	*NULL*	*000002754*	*Turbo C++ professional handbook*	*Berkeley, Calif.*	*eng*
5	*000000449*	*1*	*1*	*1*	*000000449*	*A fűtő*	*Budapest*	*hunger*
6								
7	000001093	9	6	17	000001093	Magyar - angol kéziszótár	Budapest	huneng
8	000001134	5	4	18	000001134	Angol - magyar kéziszótár	Budapest	enghun
9	000001154	9	4	7	000001154	Német - magyar kéziszótár	Budapest	gerhun
10	000003640	4	4	4	000003640	CRC handbook of agricultural energy potential of develo	Boca Raton, Fla.	eng
11	000004273	4	4	6	000004273	Streamline English	Oxford [etc.]	eng
12	000001252	3	3	3	000001252	Angol társalgási képeskönyv alap-, közép- és felsőfokon	Budapest	huneng
13	000002964	13	3	5	000002964	Angol - magyar szótár	Budapest	huneng
14	000003866	3	3	5	000003866	Random number generators and simulation	Budapest	eng
15	000001240	2	2	3	000001240	Francia nyelvtan magyaroknak	Budapest	hunfre
16	000002965	7	2	5	000002965	Magyar - angol szótár	Budapest	huneng
17	000003056	2	2	9	000003056	Magyar - német műszaki szótár	Budapest	hunger
18	000003289	2	2	3	000003289	Vector analysis for engineers and scientists	Wokingham [etc.]	eng
19	000005120	2	2	3	000005120	Torsionsschwingungen in der Verbrennungskraftmaschin	Wien [etc.]	ger
20	000006724	4	2	2	000006724	Across the river and into the trees	London [etc.]	eng
21	000006725	2	2	2	000006725	Green hills of Africa	London [etc.]	eng
22	000007048	3	2	2	000007048	Dumb witness	[London etc.]	eng
23	000000449	1	1	1	000000449	A fűtő	Budapest	hunger
24	000000462	1	1	1	000000462	Dühöngő ifjúság	Budapest	huneng
25	000000477	3	1	1	000000477	Galápagos	New York	eng
26	000000478	4	1	1	000000478	A pocket full of rye	New York [etc.]	eng
27	000000479	2	1	1	000000479	Funerals are fatal	New York [etc.]	eng
28	000000480	1	1	1	000000480	Robinson Crusoe	Oxford [etc.]	eng

1 minute

Off-line: Cinege / Purple example

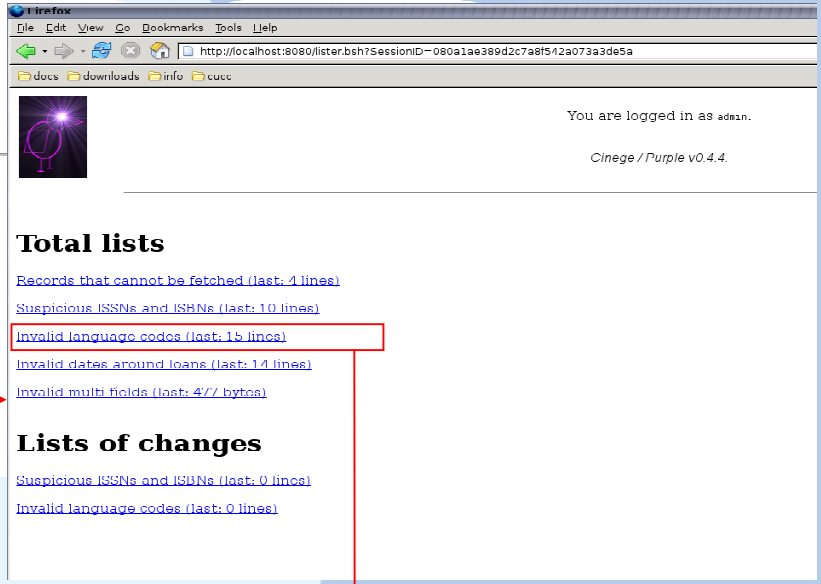
Start using Cinege/Purple.

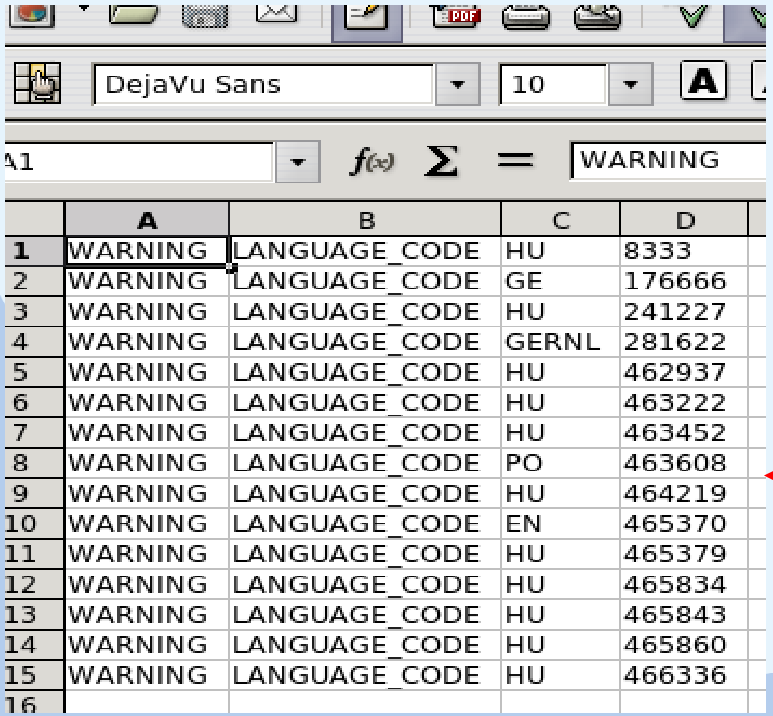
A Cinege/Lila használatának megkezdése.

Login name: admin

Password: ******

Submit





	A	B	C	D
1	WARNING	LANGUAGE_CODE	HU	8333
2	WARNING	LANGUAGE_CODE	GE	176666
3	WARNING	LANGUAGE_CODE	HU	241227
4	WARNING	LANGUAGE_CODE	GERNL	281622
5	WARNING	LANGUAGE_CODE	HU	462937
6	WARNING	LANGUAGE_CODE	HU	463222
7	WARNING	LANGUAGE_CODE	HU	463452
8	WARNING	LANGUAGE_CODE	PO	463608
9	WARNING	LANGUAGE_CODE	HU	464219
10	WARNING	LANGUAGE_CODE	EN	465370
11	WARNING	LANGUAGE_CODE	HU	465379
12	WARNING	LANGUAGE_CODE	HU	465834
13	WARNING	LANGUAGE_CODE	HU	465843
14	WARNING	LANGUAGE_CODE	HU	465860
15	WARNING	LANGUAGE_CODE	HU	466336
16				

There are 7 items of "Invalid language codes" here.

2006/09/29 (15 lines)
2006/09/27 (15 lines)
2006/09/26 (15 lines)
2006/09/25 (15 lines)
2006/09/24 (15 lines)
2006/09/23 (15 lines)
2006/09/22 (16 lines)

1 minute

Off-line: Cinege / Purple example

◇ lila.properties (/home/eknagy/current/java/data/cinege/l10n/)

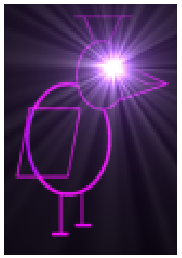
```
en START_CP         = Start using Cinege/Purple.
hu START_CP         = A Cinege/Lila használatának megkezdése.

hu LOGIN            = Felhasználónév
en LOGIN            = Login name
hu PASSWORD         = Jelszó
en PASSWORD         = Password

hu GO               = Mehet
en GO               = Submit

hu LOG_OUT          =Kijelentkezés
en LOG_OUT          =Log out

hu LOGIN_FAILED     = Sikertelen bejelentkezési kísérlet
en LOGIN_FAILED     = Login failed
hu TRY_AGAIN        = Próbálja újra
en TRY_AGAIN        = Try again
```



lila.properties

- Translation file
- Easy-to-use format
- Unicode
- Unlimited # languages

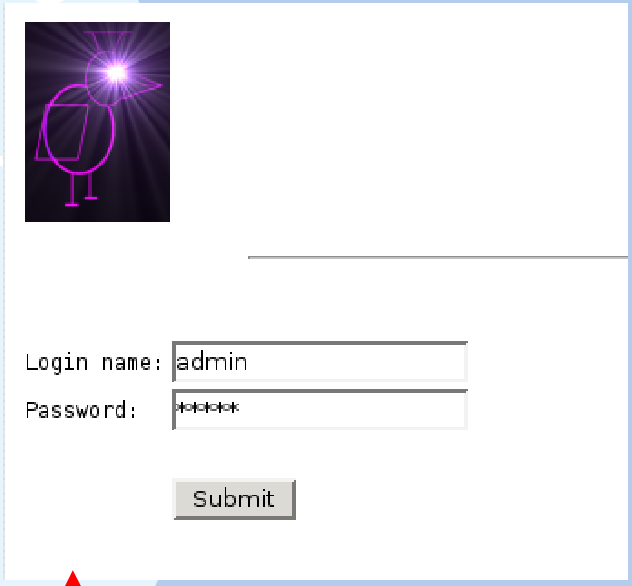
Start using Cinege/Purple.

A Cinege/Lila használatának megkezdése.

Off-line: Cinege / Purple example

purple.properties

- Security configuration file
- Basic authentication:
  - local file database
  - simple to maintain
- Patron authentication:
  - Can share files with patrons
  - Requires a Java process on the Aleph server
- Staff menu authentication:
  - Seamless integration
  - Prototype works finely



purple.properties (/home/eknagy/current/java/data/cinege/web/)

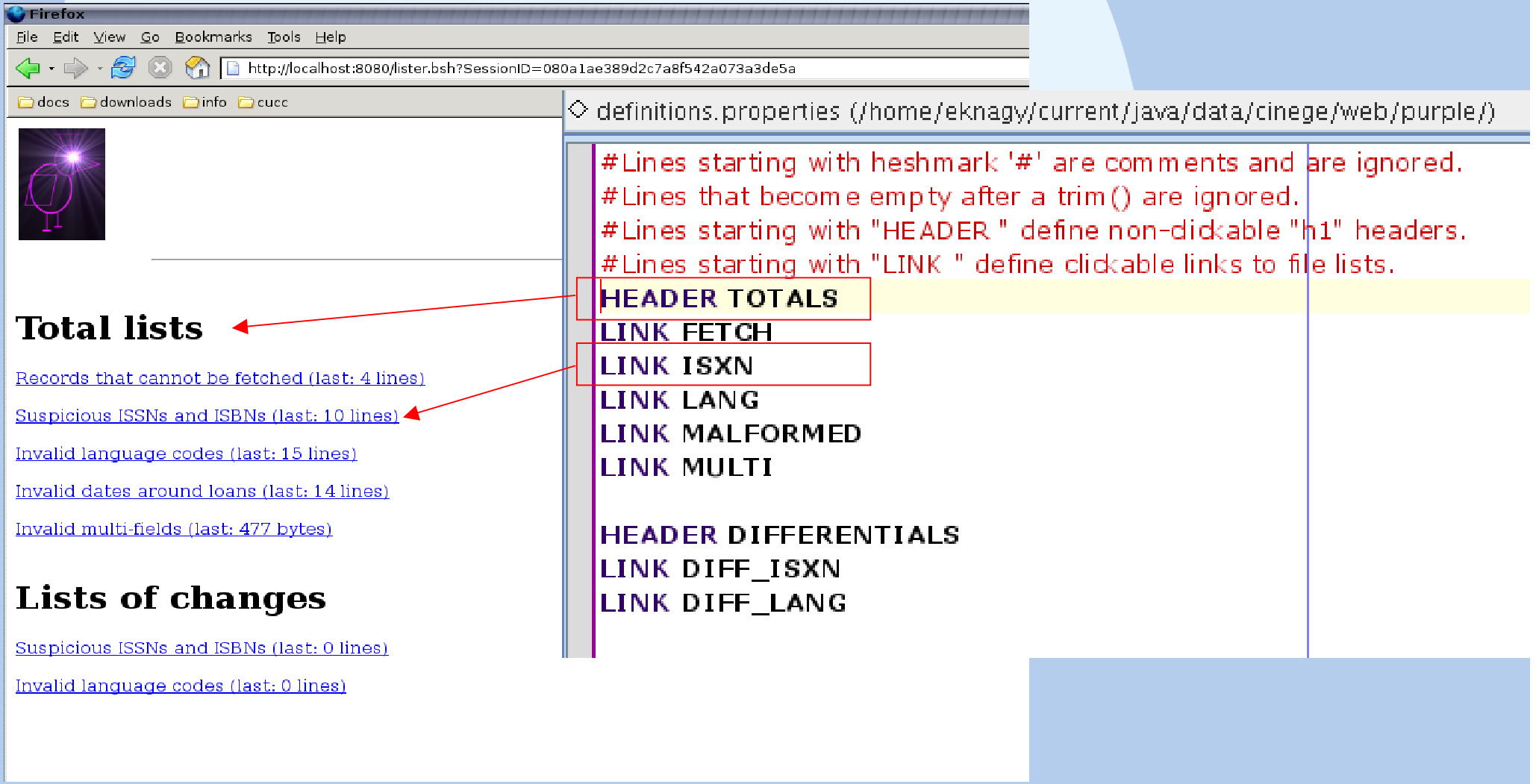
```
all Authentication=Basic
all Password:ADMIN = a1:a3:af:a9:fa:d7:25:27:c3:09:ca:8e:ca:00:9f
```

1 minute

Off-line: Cinege / Purple example

definitions.properties

- Defines headers and files to show

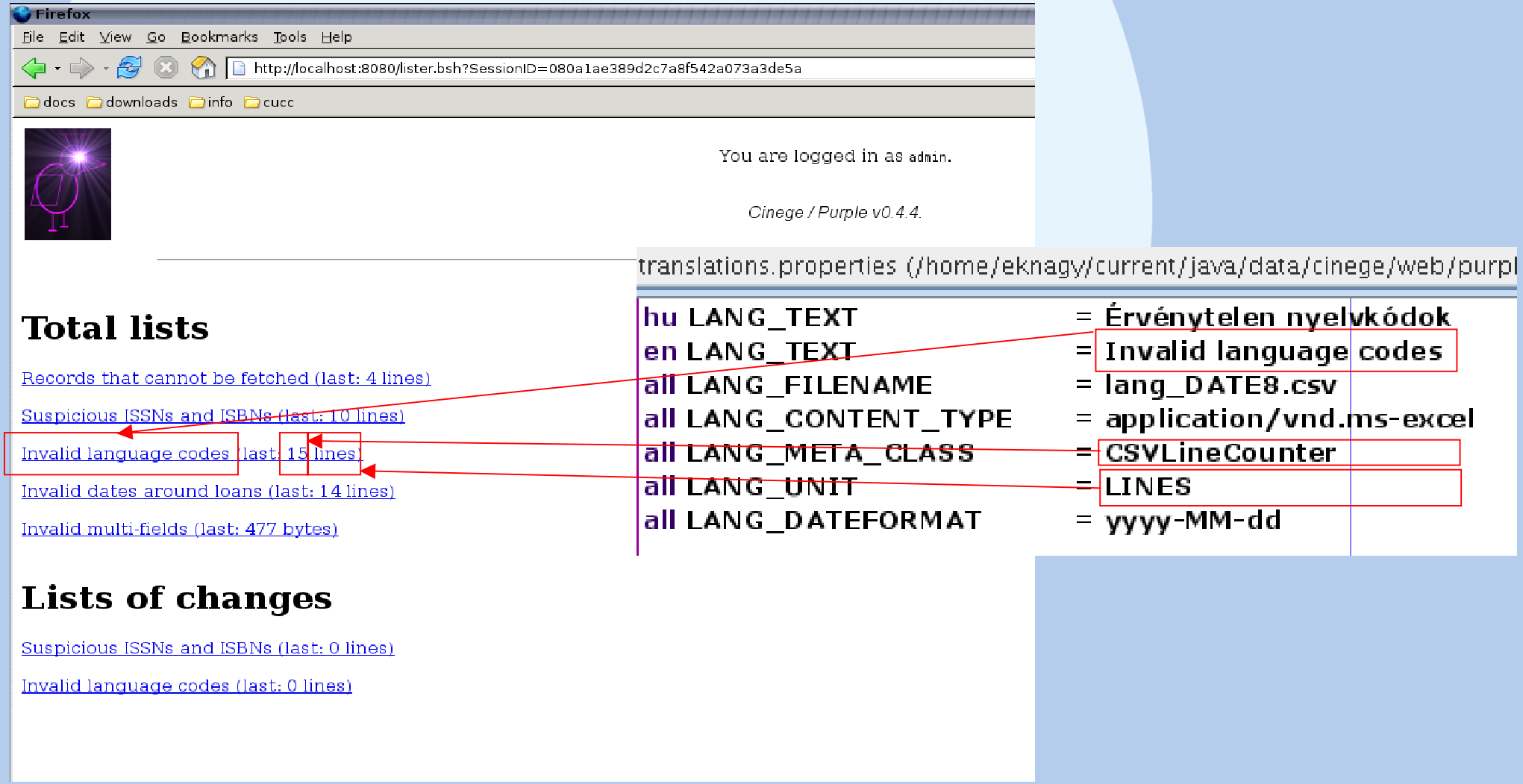


1 minute

Off-line: Cinege / Purple example

translations.properties

- Defines lines metadat



Off-line: Cinege / Purple example

There are 7 items of "Invalid language codes" here.

2006/09/29 (15 lines)
2006/09/27 (15 lines)
2006/09/26 (15 lines)
2006/09/25 (15 lines)
2006/09/24 (15 lines)
2006/09/23 (15 lines)
2006/09/22 (16 lines)

```
data/cinege/web/purpledata$ ls -lt lang*
1 2006-09-02 22:48 lang_20060829.csv
1 2006-08-28 18:37 lang_20060827.csv
1 2006-08-27 14:18 lang_20060826.csv
1 2006-08-25 17:15 lang_20060825.csv
1 2006-08-24 20:17 lang_20060824.csv
2 2006-08-24 17:31 lang_20060822.csv
1 2006-08-24 17:31 lang_20060823.csv
```

translations.properties (/home/eknagy/current/java/data/cinege/web/purpl

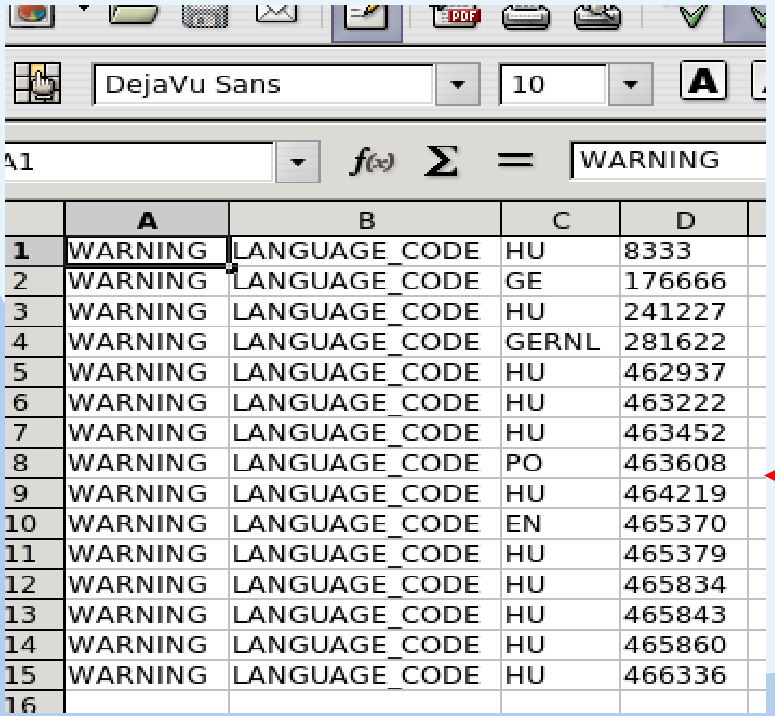
```
hu LANG_TEXT          = Érvénytelen nyelvkódok
en LANG_TEXT          = Invalid language codes
all LANG_FILENAME     = lang_DATE8.csv
all LANG_CONTENT_TYPE = application/vnd.ms-excel
all LANG_META_CLASS   = CSVLineCounter
all LANG_UNIT         = LINES
all LANG_DATEFORMAT   = yyyy-MM-dd
```

1 minute

Off-line: Cinege / Purple example

translations.properties (/home/eknagy/current/java/data/cinege/web/purpl

```
hu LANG_TEXT          = Érvénytelen nyelvkódok
en LANG_TEXT          = Invalid language codes
all LANG_FILENAME     = lang_DATE8.csv
all LANG_CONTENT_TYPE = application/vnd.ms-excel
all LANG_META_CLASS   = CSVLineCounter
all LANG_UNIT         = LINES
all LANG_DATEFORMAT   = yyyy-MM-dd
```



	A	B	C	D
1	WARNING	LANGUAGE_CODE	HU	8333
2	WARNING	LANGUAGE_CODE	GE	176666
3	WARNING	LANGUAGE_CODE	HU	241227
4	WARNING	LANGUAGE_CODE	GERNL	281622
5	WARNING	LANGUAGE_CODE	HU	462937
6	WARNING	LANGUAGE_CODE	HU	463222
7	WARNING	LANGUAGE_CODE	HU	463452
8	WARNING	LANGUAGE_CODE	PO	463608
9	WARNING	LANGUAGE_CODE	HU	464219
10	WARNING	LANGUAGE_CODE	EN	465370
11	WARNING	LANGUAGE_CODE	HU	465379
12	WARNING	LANGUAGE_CODE	HU	465834
13	WARNING	LANGUAGE_CODE	HU	465843
14	WARNING	LANGUAGE_CODE	HU	465860
15	WARNING	LANGUAGE_CODE	HU	466336
16				

There are **7** items of "Invalid language codes" here.

2006/09/29 (15 lines)
2006/09/27 (15 lines)
2006/09/26 (15 lines)
2006/09/25 (15 lines)
2006/09/24 (15 lines)
2006/09/23 (15 lines)
2006/09/22 (16 lines)

Thank you very much!

Any questions?

(if there is any time left)

Nagy, Elemér Károly

eknagy@omikk.bme.hu

Krauthausen, Leon

krauthausen@ub.fu-berlin.de

IGELU 2006 Stockholm