

T_EX, BLUes, and Jazz—a search for values

Kees van der Laan

Abstract

How to use T_EX for a database oriented application is shown by the example of a collection of Jazz CDs. The request was to provide (part) of the title of a tune and to let T_EX response with all the CDs which contain that tune. Only a few lines of code was all that was needed to fulfill the purpose. The example might serve as a user-oriented stepping stone for understanding how in BLUe's format T_EX as database engine has been used.

Keywords: BLUe, database, jazz, plain T_EX

Of late I was interested in a way of keeping track of my Jazz collection. Given a tune I wanted to know who had performed it, well ... on the CDs (, records and (vcr) tapes) I own.

A simple application of the database idea, and why not use T_EX for it?¹

1 T_EX as a database engine

On various occasions T_EX has been mentioned as a database tool. Jurriëns has used T_EX as a mail-merge engine. Lenstra and friends launched $\delta\alpha$ T_EX, to handle general data. I myself use T_EX as database ubiquitously. And undoubtedly many others do.

My aim is to share my 'little language,' well a few macro lines, which will achieve the modest request, a very rudimentary database-like 'query.'

At the heart lies BLUe's format `\search` macro, which for this application has been simplified even more, and is called `\logsearch`, because it writes in the log file only.

```
\def\logsearch#1{\immediate
  \write0{Searched for: #1}
  \def\lst##1##2{\loc{#1}{##2}\iffound
    \immediate\write0{\noexpand##1}\fi}
  \input jazz.dat
}
%with auxiliaries
\newcount\cnt \newif\iffound
\def\loc#1#2{\def\locate##1#1##2\end
  {\ifx\empty##2\empty\foundfalse
    \else\foundtrue\fi}\expandafter
  \locate#2.#1\end }
```

`\logsearch` matches its argument with the entries in the data file and it returns the name(s) of the matched entries in the logfile. The reporting is preceded by the search string.

Basically, it comes down to appropriately defining the list element tag, `\lst`, which is the first of 3 elements to characterize each database entry.

```
\lst<name>{<entryproper>}
```

I used as `<entryproper>` the titles of the tunes, each on a line, and as `<name>` I used the main performer, the group-name or any other appropriate identification, with a sequence letter added. To simplify matters further I used lower case characters only and excluded the use of special characters like `&`.

Example (*Search for performers of Summertime*)

The following

```
\newif\iffound \escapechar'055
\logsearch{summertime} \bye
```

yields

```
(jazz.search
Searched for: summertime
(jazz.dat
-blakeyarte
-davisma
-gershwinga
-getzsa
-holidayba
-parkercb
-rossda
-vaughansd
```

¹In fact this application is simple and can be considered as a pedagogical stepping stone to understand the use of T_EX as database engine to handle formats, tools, pictures, addresses, literature items and ilks, as I did in `blue.tex`. As you can see I use few, as few as feasible, structuring tags in the data. (Of course when you provide all parts of an item with a markup tag then you can do anything.) Typing the data remains tedious, alas, and is error-prone. No fancy database-like data entry as yet, alas. But, ... for me an editor and ASCII are sufficient. The big advantage is that I'm *not* dependent on vendors nor versions of database software releases. I can exchange the files easily and reuse them. Crucial is that I have modest wishes.

```
-vaughansg
-vaghansj
)
```

So it is performed by Art Blakey, Miles Davis, on the Gershwin tribute CD, Stan Getz, Billy Holiday, Charlie Parker, Diana Ross, and on various CDs by Sarah Vaughan.

Writing in the logfile is convenient, also when I use it remotely. More than one search can be done in one run.

2 A print of the total collection

Given the names it is easy to spot the entries, be it physically or in the total printout. The latter can be generated by the following job.²

```
\newcount\cnt
\def\{\{\hfil\break}
\def\lst#1{\advance\cnt1
  \\\the\cnt\string#1\quad}
\escapechar'055\obeylines
\input jazz.dat
%Total number of entries at the end
\vfill
```

```
Entries: \the\cnt
\ject\end
```

Each item will be printed and preceded by its sequence number, with the names of the tune on separate lines. The frightening escape character is printed as the friendly hyphen.³

The application is robust against variations in spelling—or typos—because any substring can be specified as search string. The discipline in using comes down to specifying the most relevant part only. Handle data entry carefully and accurately (sigh, what a pleonasm). To verify the relevant parts of the titles is the very least.

3 Conclusion

With a few macro lines a little of the search and report database functionalities have been added to plain T_EX, well ... it can be added to any flavour of T_EX. The problem size is 300 at the moment, the number of my CDs and records in my collection.

Have fun, and all the best.

²With blue.tex it is default in the more economical 2-column.

³Agreed, a very rudimentary realization of a database-like report generation, but for me it is sufficient.