

ZipBurst™ Tech Backgrounder

For more information, contact David Dantowitz at (628) 400-Code or via email at david@dantowitz.com.

ZipBurst is a multi-threaded NoSQL database engine that accepts queries from Apache and other apps. Queries are direct to one or more data tables and a set of result records are selected. The records are sorted using information from the query, and then merged into a template file and returned as any type of data (e.g., XML, HTML, CSV, etc.) You can also generate a file to be post-processed by an external plug-in or app (e.g., PHP).

ZipBurst, written in C, includes a custom database search engine and two interpreted programming languages for developers to customize results.

Languages

ZB-Tag Language (ZBTL) is a template or tag-based language that permits arbitrary embedding and is merged with data from the search engine to provide a query result. ZBTL is a complete and capable language, able to perform any type of computation. Using ZBTL enables customization of results and even recursive calls to the database engine. This has resulted in the design of some elegant, self-contained web-solutions.

The second language is used primarily within the database engine for embedded computation but may also be used with ZBTL. This embedded language, akin to microcode is named ZipBurst Microcode Language, or ZBML. ZBML is a stack-based language with procedural support and late binding. Using ZBML you can insert global and local computations at 10 distinct points within a search query. This solution was developed to address a client's request for daily changes within the database kernel as they fine-tuned how they wished to search data.

Of note is that both languages can call the other, giving ZBML access to ZBTL primitives and giving ZBTL access to the finesse of ZBML.

Searching & Sorting

Searching is as you would expect: each web query includes one or more search directives, essentially conditionals (equal, not equal, less than, greater than, contains, starts with, ends, with etc.) and supports data types you would expect: integers, floating point, dates, strings and so on. Once data is selected, the results are then sorted, based on information from the web query.

Merging Results with a Template file

After a search is complete and the records sorted, the app reads a template file containing tags to be processed and merged with the data. A simple example might look like this:

```
<p>
[ZB-ResultCount] records were found.

Locations:
[ZB-Result]
  [ZB-Field name=City]<br>
[/ZB-Result]
```

Such template could yield

```
4 records were found.

Locations:
Boston
New York
Palo Alto
Seattle
```

ZipBurst processes the template file and interprets the “tags” which have the form:

```
[ZB-tagname ... ]
```

resulting in actions, text, or both.

The tags above are simple in nature:

- 1) Report how many results were found: **[ZB-ResultCount]**,
- 2) Start and end a result sub-section, bounded with **[ZB-Result]** and **[/ZB-Result]**, noting that the content within is repeated for each resulting record found by the query (you can cap the number records returned in a single query).
- 3) Retrieve specific data from the current matched record: **[ZB-Field name=fieldName]**

More Advanced Tags

If/Then/Else

Loops, with the ability to jump to the start of the loop as well as exit (i.e., continue and break in C)

Variable Support for strings, ints, floats, dates, and booleans

Nested queries within the report template

Relational queries across multiple data tables

An http call to another server (e.g., get the current temperature) and others ...

Nesting

Tags can also be nested arbitrarily. For example nesting a second If/Then/Else within the Then or Else body of an enclosing If/Then/Else.

A single If/Then/Else

```
[ZB-If arg1=... arg2=... op=...]  
  True content  
[ZB-Else]  
  False content  
[/ZB-If]
```

A nested If/Then/Else

```
[ZB-If arg1=... arg2=... op=...]  
  True content  
    [ZB-If arg1=... arg2=... op=...]  
      Double True content  
    [ZB-Else]  
      True-False content  
  [/ZB-If]  
[ZB-Else]  
  False content  
[/ZB-If]
```

In addition to tags being nested, the value for any arguments within a tag may embed any tag:

```
[ZB-If arg1="[ZB-Field name=city]" arg2=... op=...]  
  True content  
[ZB-Else]  
  False content  
[/ZB-If]
```

Even an IF tag could be used within an argument. As expected, Loops may be nested, and support early restart and exit of the current loop.

As the template files' tags provide a complete programming language, you can deliver both simple and complex results with ease.

Query arguments passed in via Apache can request a highly varied mix of criteria for a search, e.g., match or contain a string, distance from a location, a numeric value with a specific relationship, and these may be combined via AND or OR to select records in the database. Varied sorting fine tuning is also possible. You can also use ZBML (described a bit more below) for computation of arbitrary search and sorting criteria.

ZipBurst Microcode Language (ZBML)

If the query you have in mind requires more than one or more simple, “one dimensional” criteria, ZipBurst also contains a microcode-like language that is interpreted during the query and can be used in place of any argument. So, if you wish to search based on an average of two values in a record, but don't wish to create a new table or column, you can do that with ZipBurst Microcode Language (ZBML) on-the-fly.

ZBML can be executed at 10 distinct points within a query, enabling you to have great flexibility. ZBML can also be used as an argument for search criteria and as a value within any tag in a template file (there is a tag to execute ZBML and that tag may be used in an argument within another tag).

ZBML is a stack-based language similar to Forth and PostScript. It supports the data types: Int, Float, String, Lists, Literals and Procedures, stack-based operations, as well as an option for early vs. late binding.

Other Details

Matching: ZipBurst also offers adaptive spelling options for queries where the data source and the query aren't exact matches.

Web Form Argument Hiding: In order to simplify URLs and hide details of queries, you can store static elements of an HTTP query in an argument file. This results in cleaner URLs and enables you to hide some internal database details from the web.

Types of Data: Database files may be CSV or TAB delimited static files, exported or published from other systems and deposited exclusively for searching. Another option is available for live, editable data. The live data option is used identically to static data files and has a secondary interface to access and modify data over via the web.

Data Time-Spanning: When using live database files, ZipBurst enables queries to access data at specific time slices, dubbed “Data Time-Spanning” or DTS. One use for this is if you begin a set of queries, and the database is changed during a set of queries, you have the ability to continue accessing the database as it was when you began your

queries. So a group of queries will yield a self-consistent set of results despite on-going changes. A “plus” feature for researchers is that DTS enables databases to be accessed across arbitrary times in the life of the database. Of note: the DTS concept is a side product of the implementation to enable sustained read and write operations with minimal latency.