

Store large texts as container fields

Recently at the dotfmp conference, we talked about storing text vectors for LLMs in FileMaker as container instead of text to save space. But does it? Let's check with current FileMaker Pro.

We can store text in a FileMaker database. The size to store the record is a bit more than the text. Our test table has primary key, two time stamps and the account name for creation and modification. With the overhead for managing the blocks this takes about 270 bytes per record in our test database.

If you store the text in a container field, you basically get the text stored and there is a 12 byte overhead for the container, if you just write text there.

```
Set Field [ Test::ContainerField ; $text ]
```

Alternatively we can package the text as UTF-8 with [Text.WriteToContainer](#) function. That will package the text as a file. This causes an overhead of about 200 bytes as multiple streams of data are managed including a file name. Total size of the container value itself is 36 bytes more than the text length. But by using container instead of a text field, we can save on indexes.

Now we can use [Archive.CompressContainer](#) to compress the container value as a zip archive. This reduces a bigger text easily by 70%. We can skip using the [Text.WriteToContainer](#) function in future with the new [Archive.CompressText](#) function.

```
MBS( "Archive.CompressContainer"; "zip"; "deflate"; "test.zip";  
MBS( "Text.WriteToContainer"; $text & Test::PrimaryKey; "UTF-8";  
"test.txt"); "compression-level=9")
```

Or with [Archive.CompressText](#) function in upcoming plugin:

```
MBS( "Archive.CompressText"; "zip"; "deflate"; "test.zip"; $text;  
"compression-level=9")
```

Storing the text as container with compression with either our [Archive](#) functions or with [Container.Compress](#) may help you to save disk space.

But by storing text in containers, you also enable a few neat features:

- The text in containers is not loaded when the record is loaded, but only when accessed.
- The text in container can be stored externally.
- The text doesn't get indexed, so we save space there.
- Since containers are referenced, duplicating the record just increases the reference count, but doesn't duplicate the data.

Have fun exploring this possibility if you have to store a lot of text with records.