

Claris FileMaker

Under the Hood with Clay

Clay Maeckel

Draco Software Architect



What is going to be covered

- Draco Data object and JSON improvements
- Patching with Claris FileMaker Upgrade Tool
- Other changes
 - Open Source library updates
 - Future of OData
 - Recent vRev releases



Draco Data object and JSON



What is Draco

The Database Engine of FileMaker

- C++ Engine
- Broken down into three different layers
 - Support
 - DBEngine
 - FMEngine
- Runs on both the clients and the servers



What is DBData

- Basic C++ object that stores content for a field, parameter, or variable
- Support all the FileMaker types: text, number, date, time, timestamp, container
- Everything but containers are stored as text on disk with Draco
- While in memory there are native binary representations if needed
- Handles conversion between different data types, locales, and font usage
- Locale and font information comes from the file the DBData is associated with

What is DBData

The Plug-In SDK has C++ header files that are a subset of the full object definition

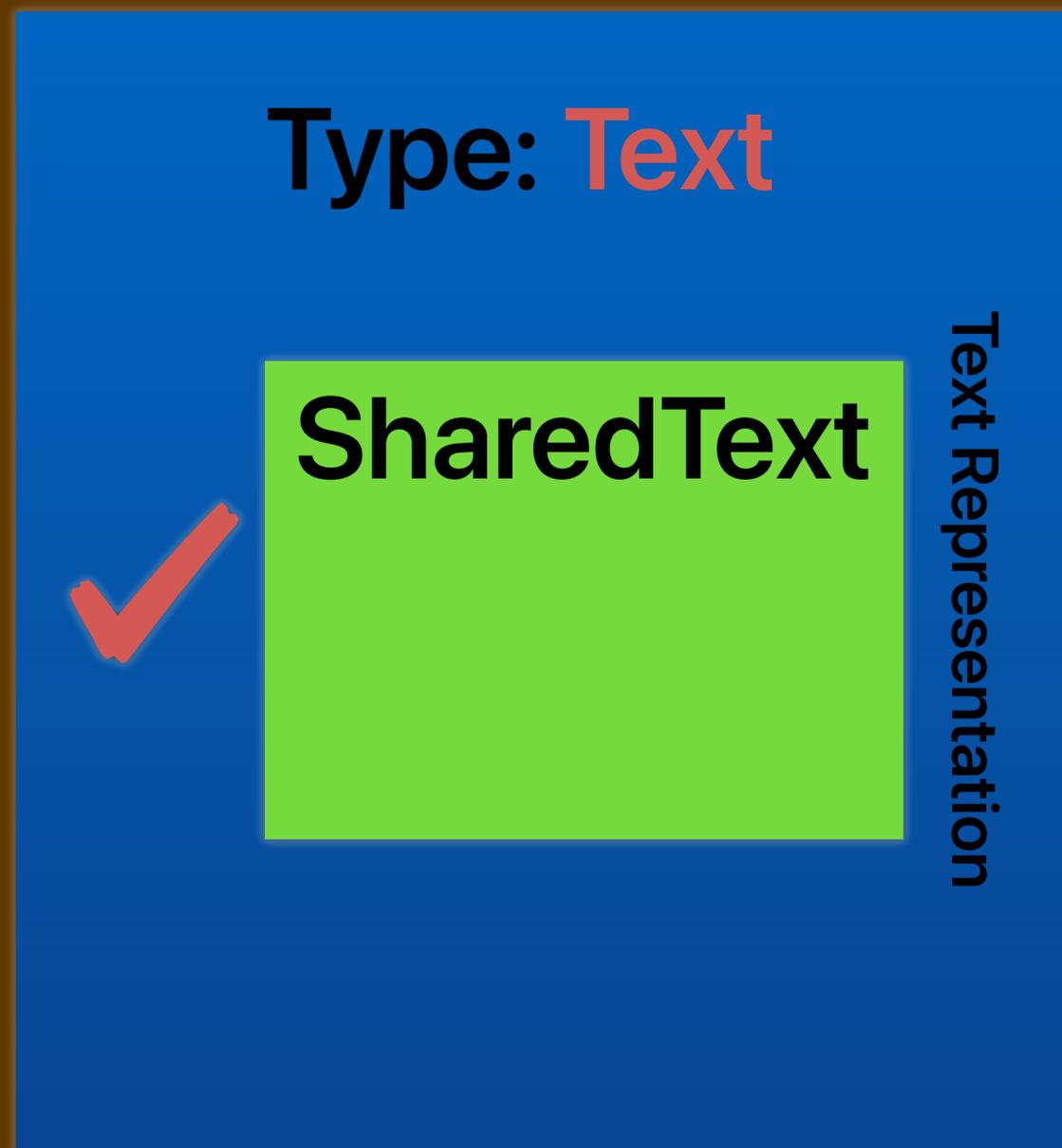
```
class Data
{
public:

    enum DataType
    {
        kDTInvalid, kDTText, kDTNumber,
        kDTDate, kDTTime, kDTTimeStamp,
        kDTBinary, kDTBoolean
    };

    bool IsEmpty () const;
    bool IsValid () const;
    bool IsFindRequest () const;
    void Clear ( DataType newNativeType = kDTInvalid );
    const Text &GetAsText () const;
    const FixPt &GetAsNumber () const;
    const DateTime &GetAsDate () const;
    const DateTime &GetAsTime () const;
    const DateTime &GetAsTimeStamp () const;
    bool GetAsBoolean () const;
    const BinaryData &GetBinaryData () const;
    const Locale &GetLocale () const;
    const DataType GetNativeType () const;
    void ConvertData ( DataType nativeType );
    errcode SetAsText ( const Text &textData, const Locale &sourceLocale, const DataType nativeType = kDTText );
    void SetAsNumber ( const FixPt &numericData, const DataType nativeType = kDTNumber );
    void SetAsDate ( const DateTime &dateData, const DataType nativeType = kDTDate );
    void SetAsTime ( const DateTime &timeData, const DataType nativeType = kDTTime );
    void SetAsTimeStamp ( const DateTime &timeStampData, const DataType nativeType = kDTTimeStamp );
    void SetDateTime ( const DateTime &dateTimeData, const DataType dateTimeType, const DataType nativeType );
    void SetBinaryData ( const BinaryData &binaryData, bool forceBinaryNativeType = true );
};
```

JSON parsing version 1

JSON was always stored as text and was parsed into a binary structure when needed and converted back to text when done



```
JSONGetElement( $json ; "idx" )
```

Parse

Json

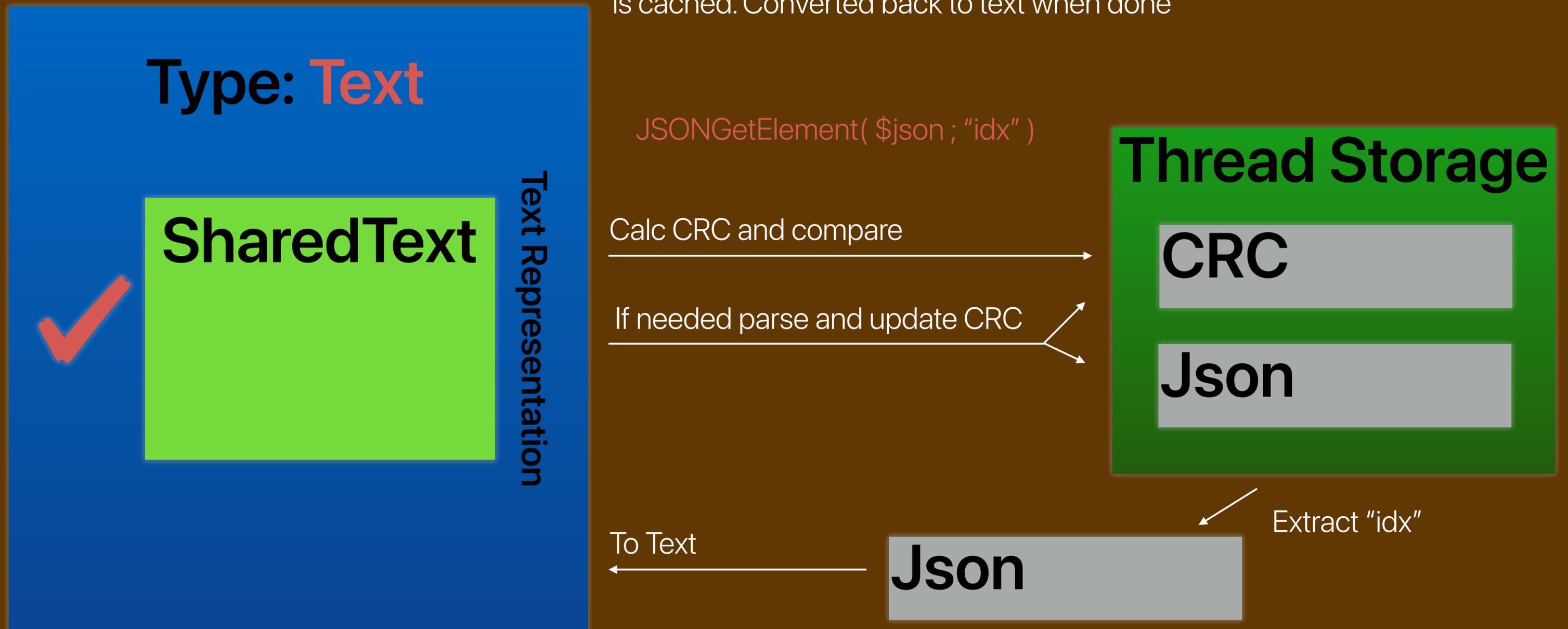
Extract "idx"

Json

To Text

JSON parsing version 2

Thread based caching of the last parsing operation added in 18. Used a CRC to determine what is cached. Converted back to text when done

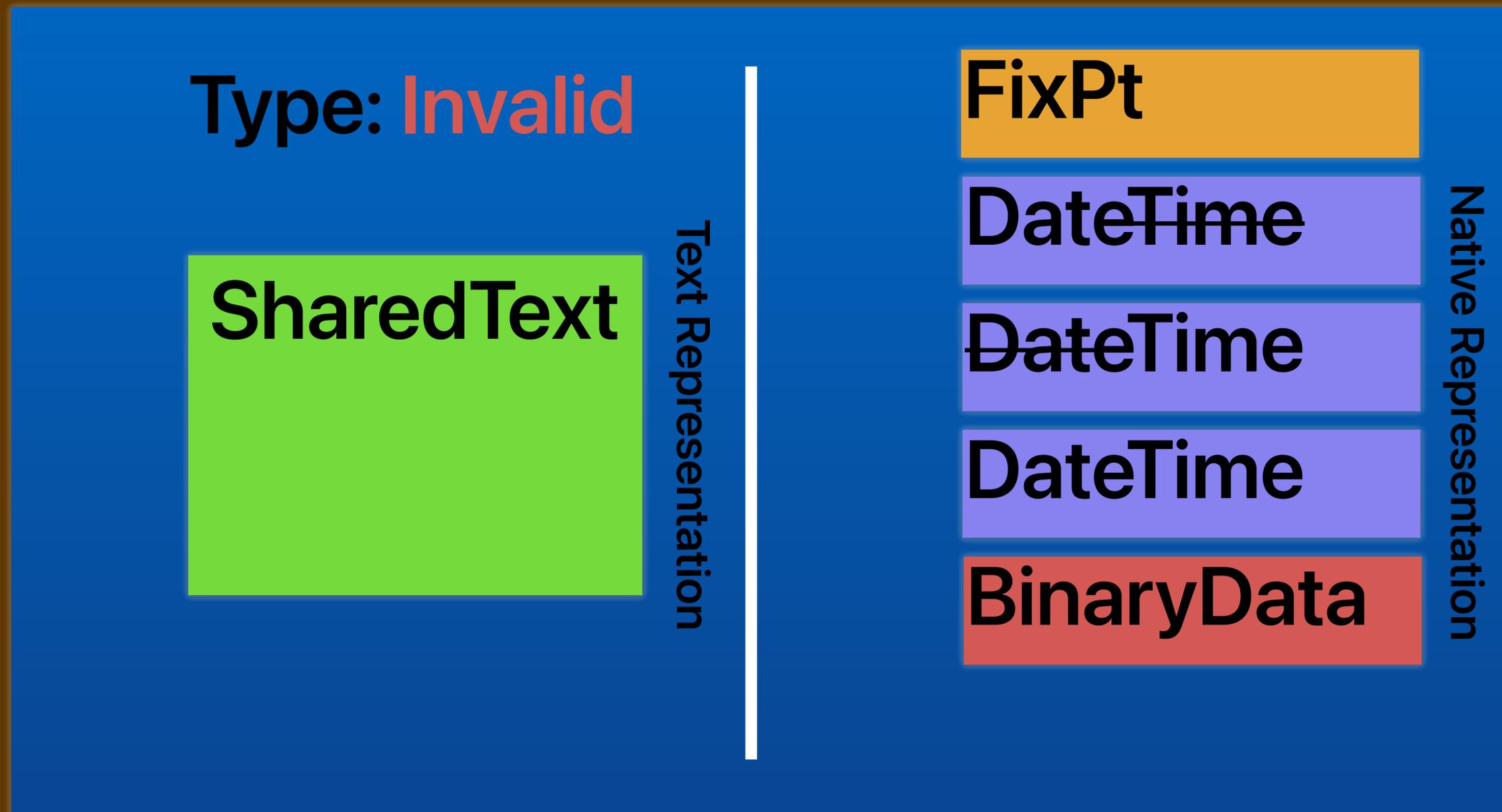


Getting deeper



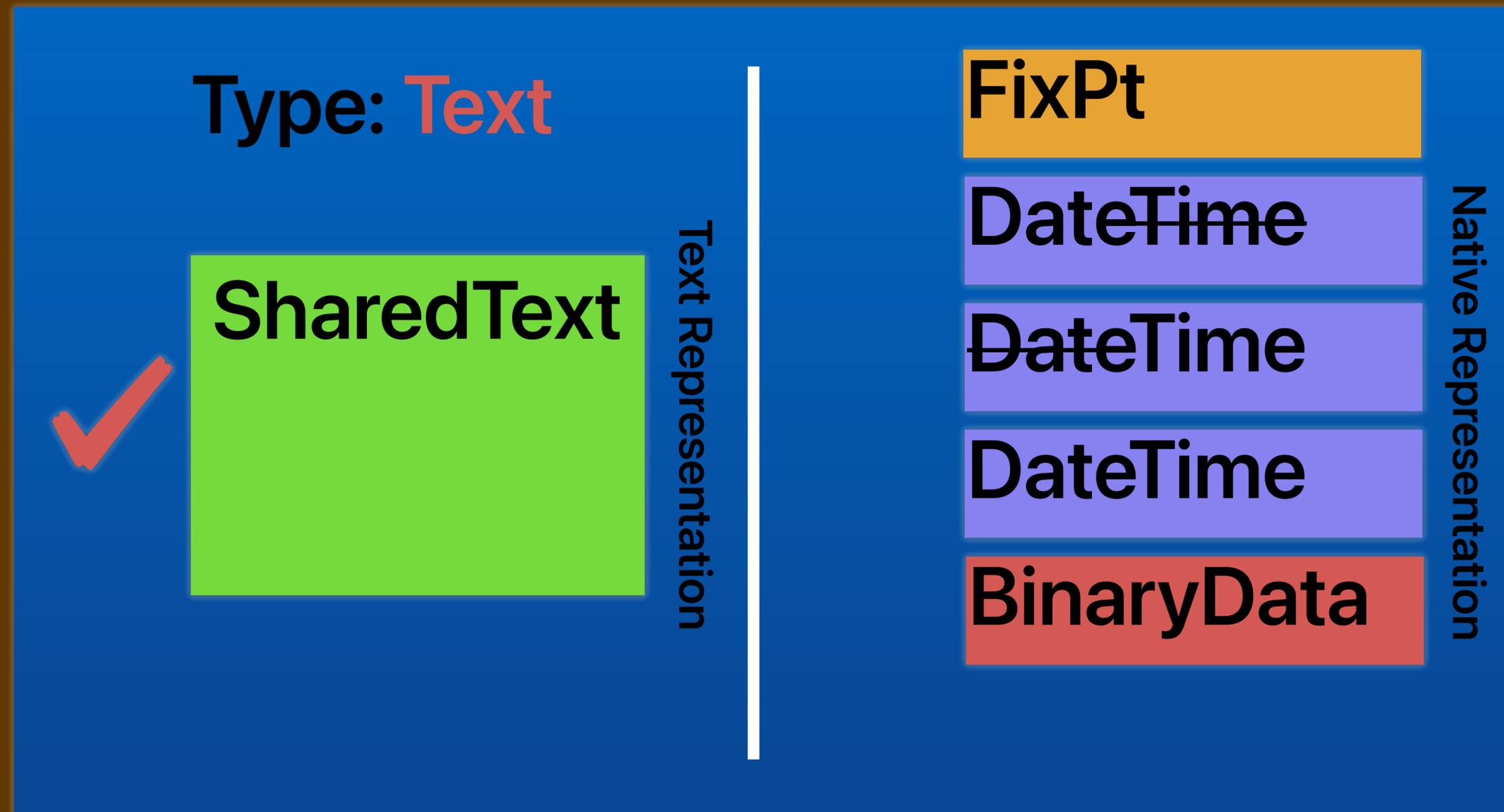
\$var = txtField + 1

Starting state, empty DBData



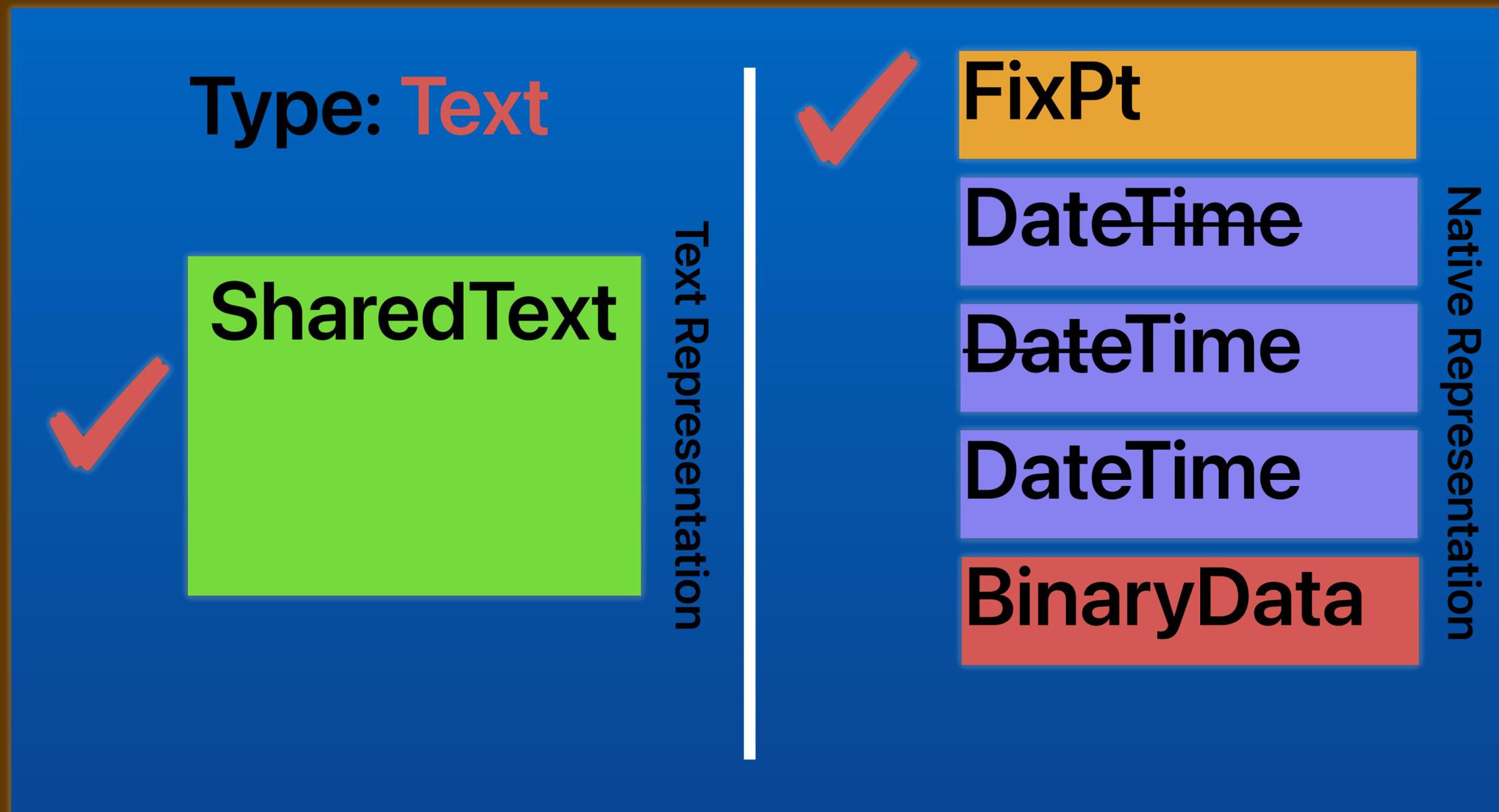
\$var = txtField + 1

Load DBData with contents of txtField



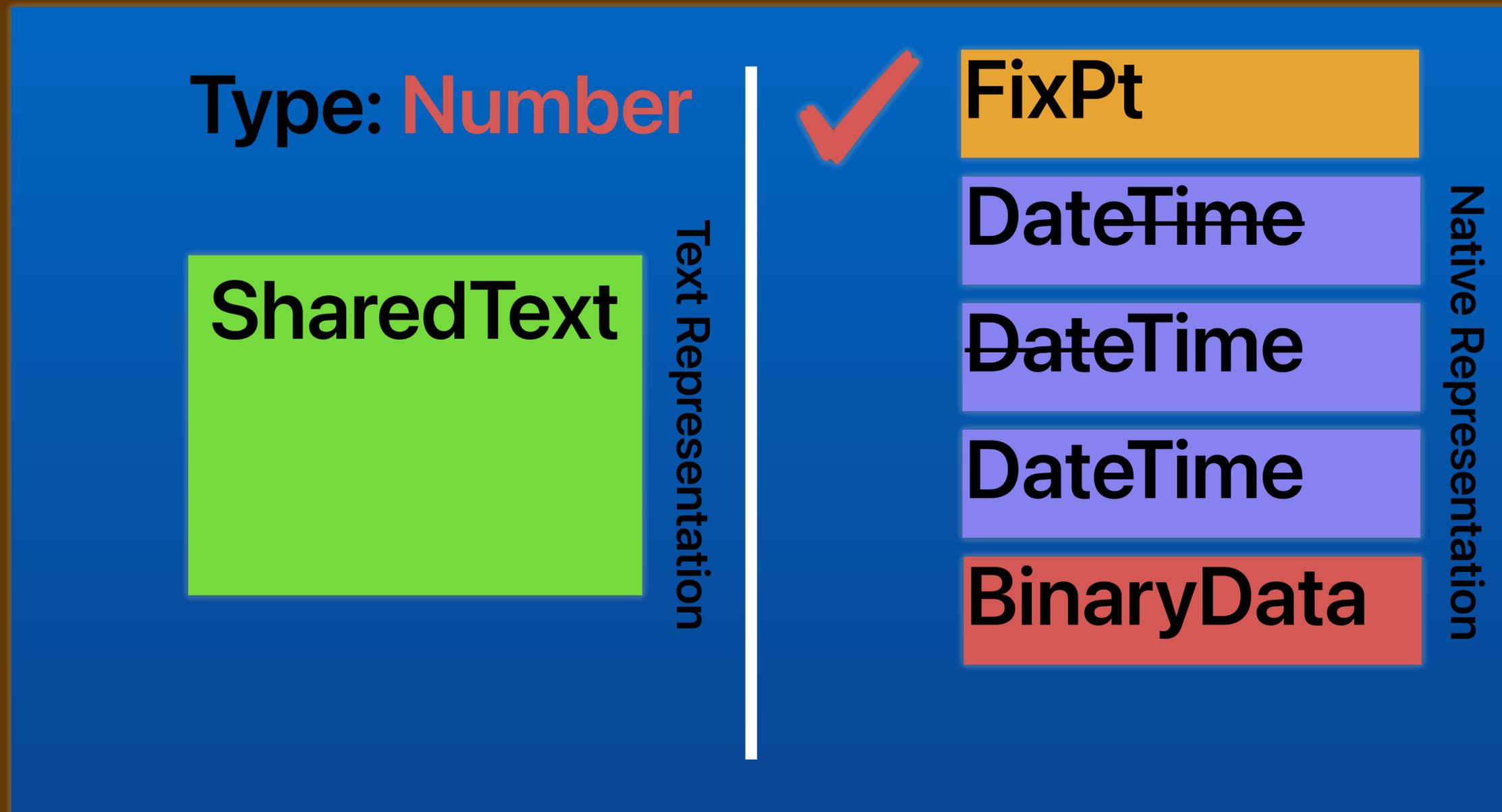
\$var = txtField + 1

Adding a number to this requires getting the numerical value of the text



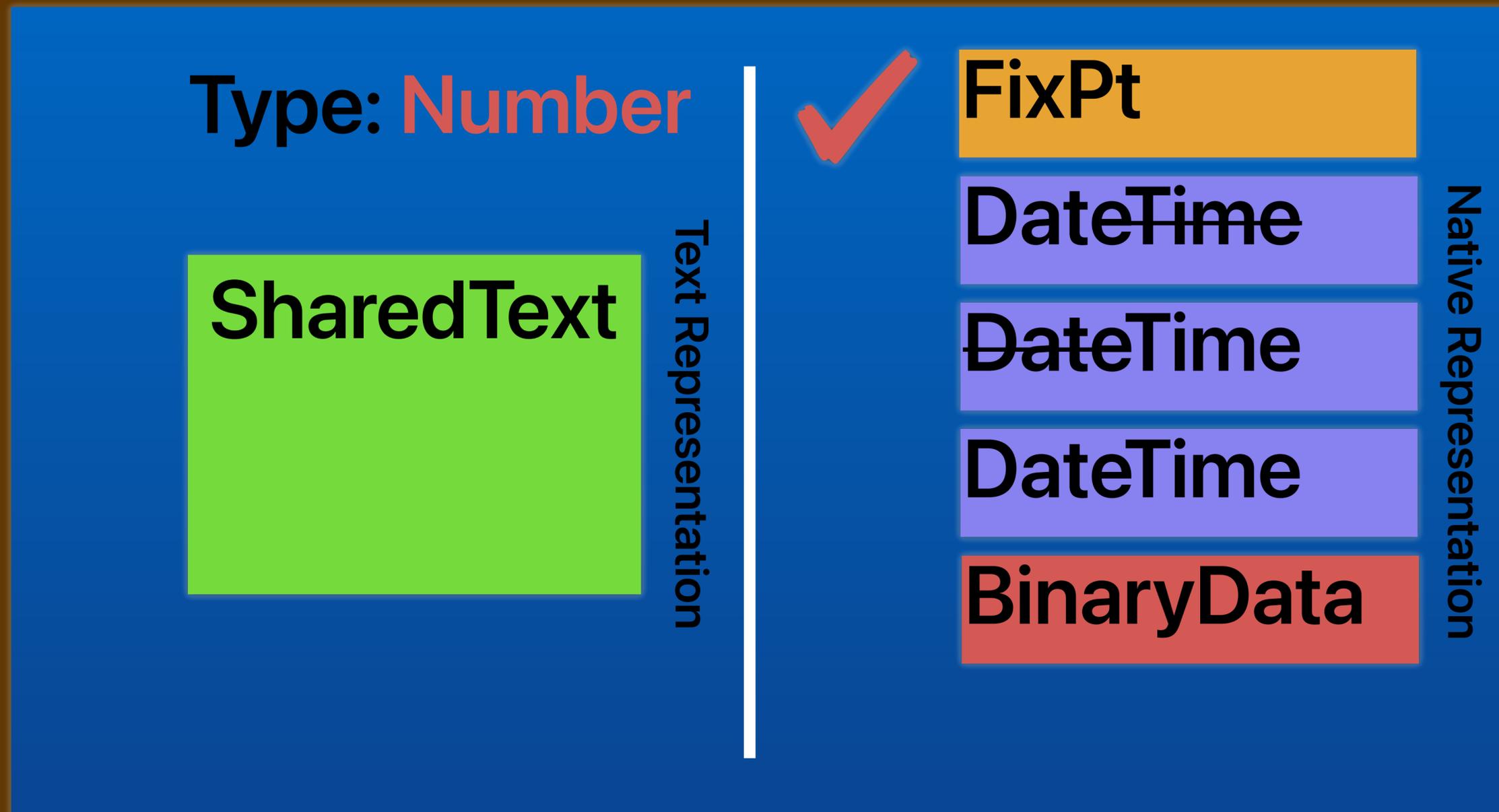
\$var = txtField + 1

Adding one to this DBData



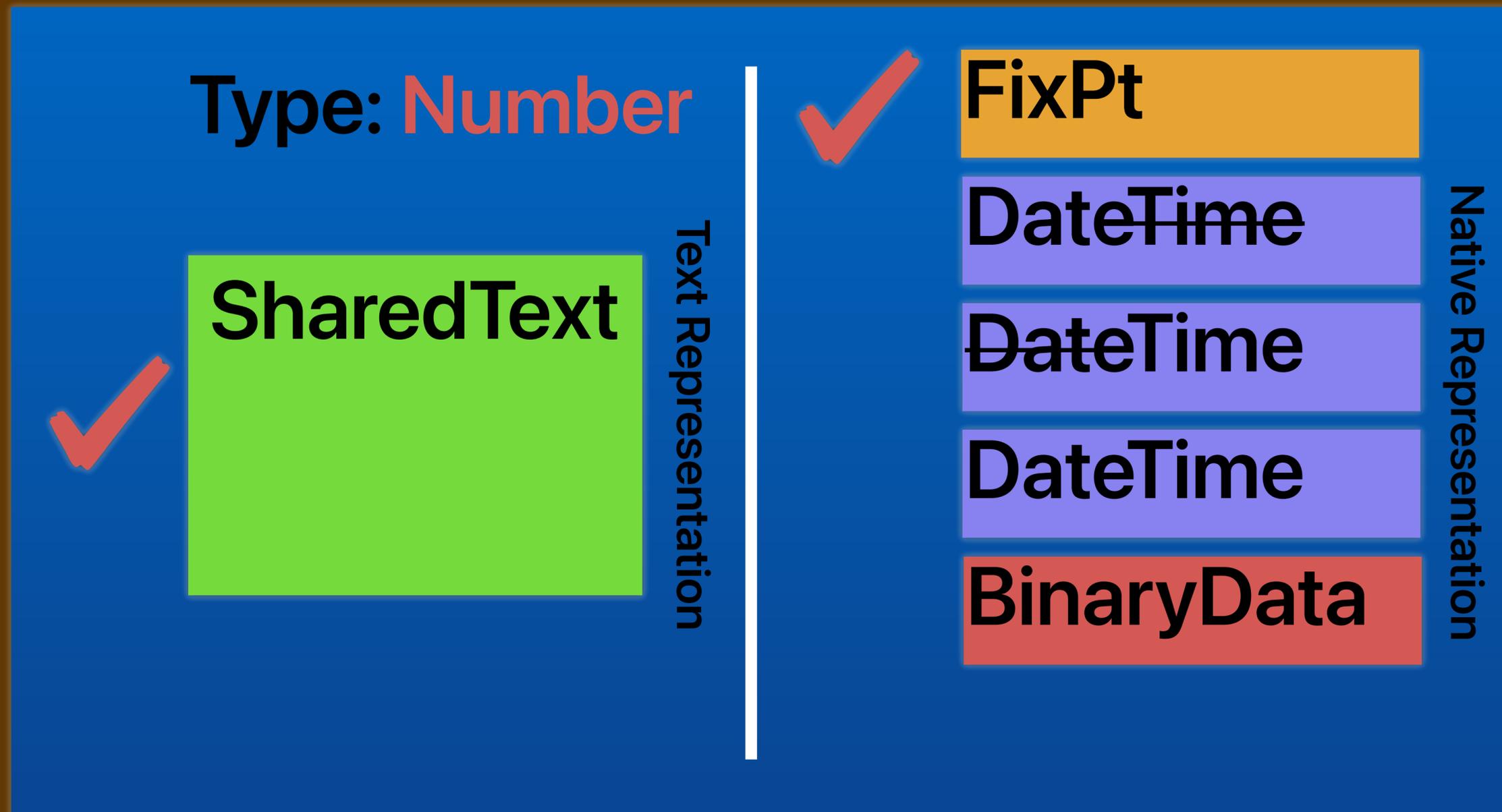
\$var = txtField + 1

Saving the results in the variable



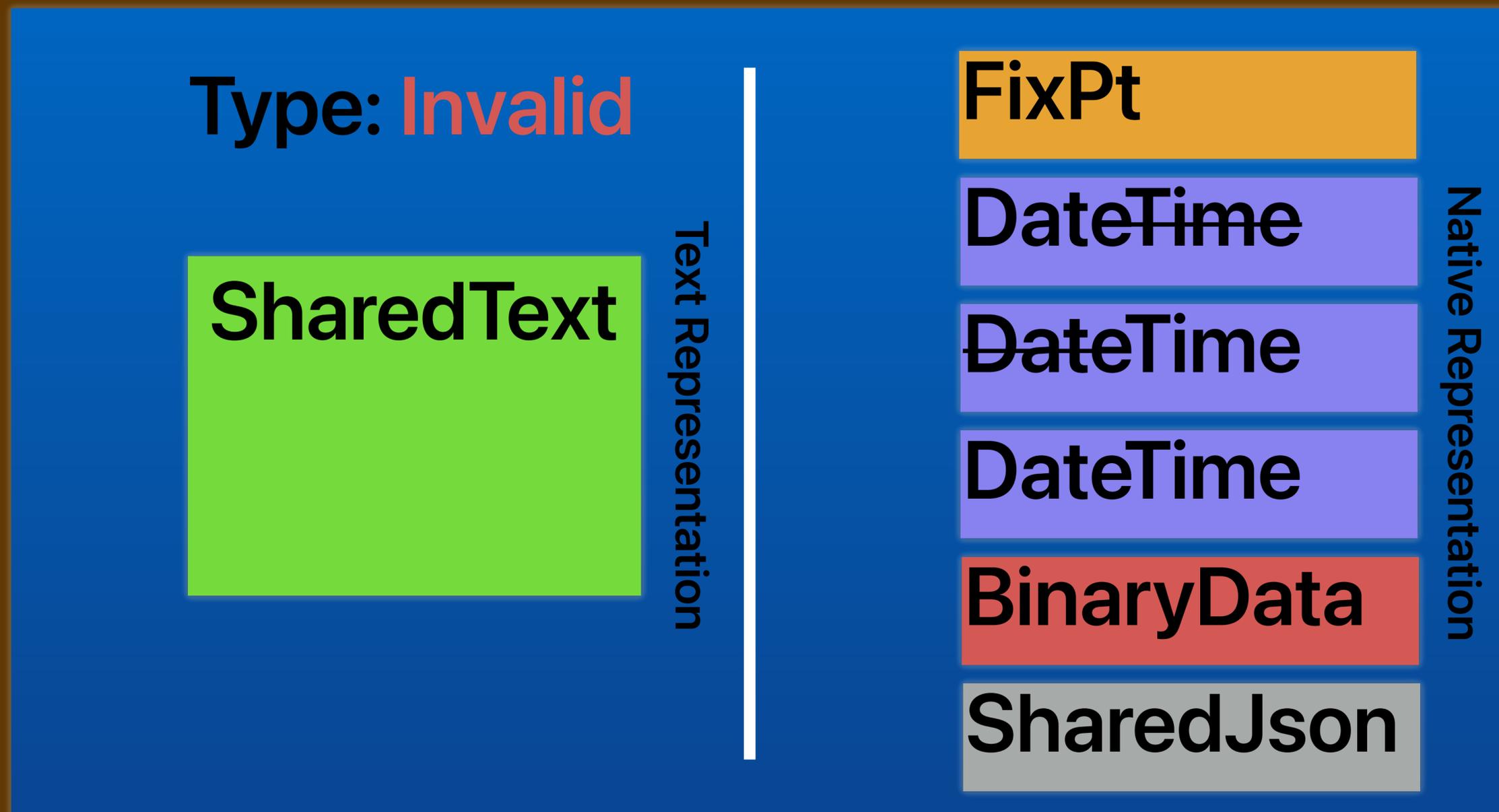
GetAsText(\$var)

Convert the native representation to text, no need to throw away the FixPt yet



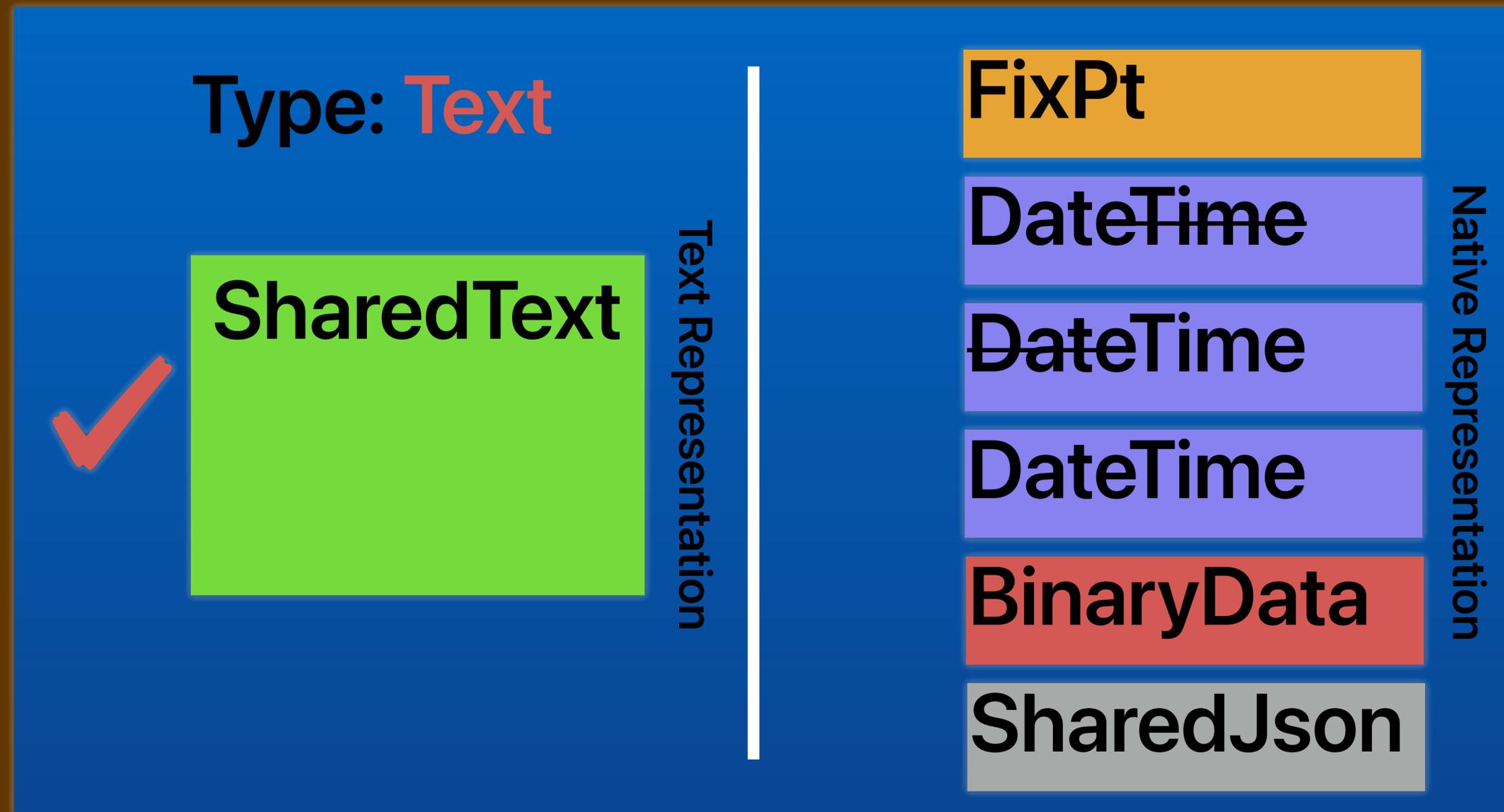
JSON parsing version 3

With the new version of FileMaker in 22, DBData has a native JSON representation



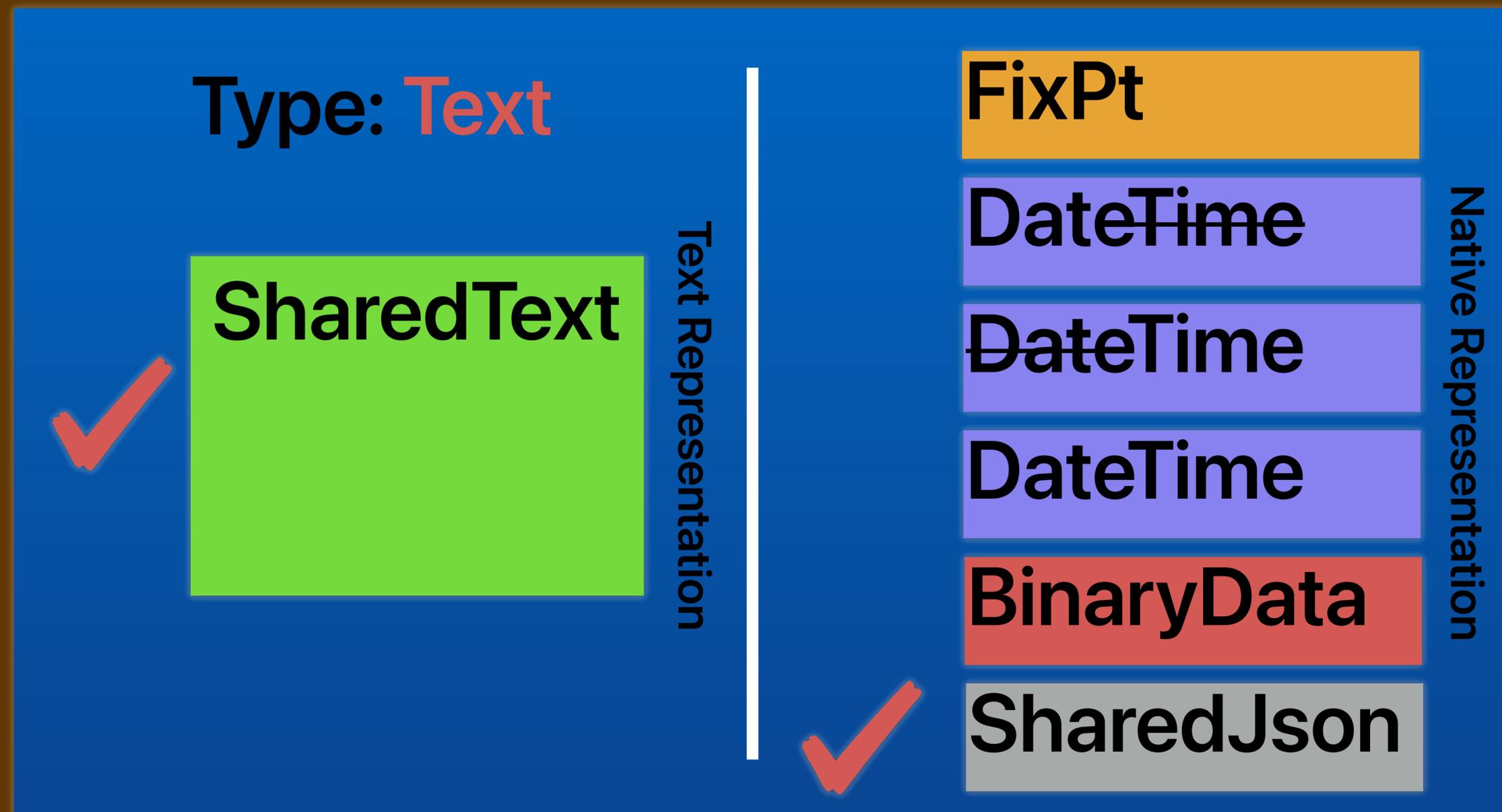
```
$var = JSONGetElement( txt ; "idx" )
```

Load DBData with contents of txt field



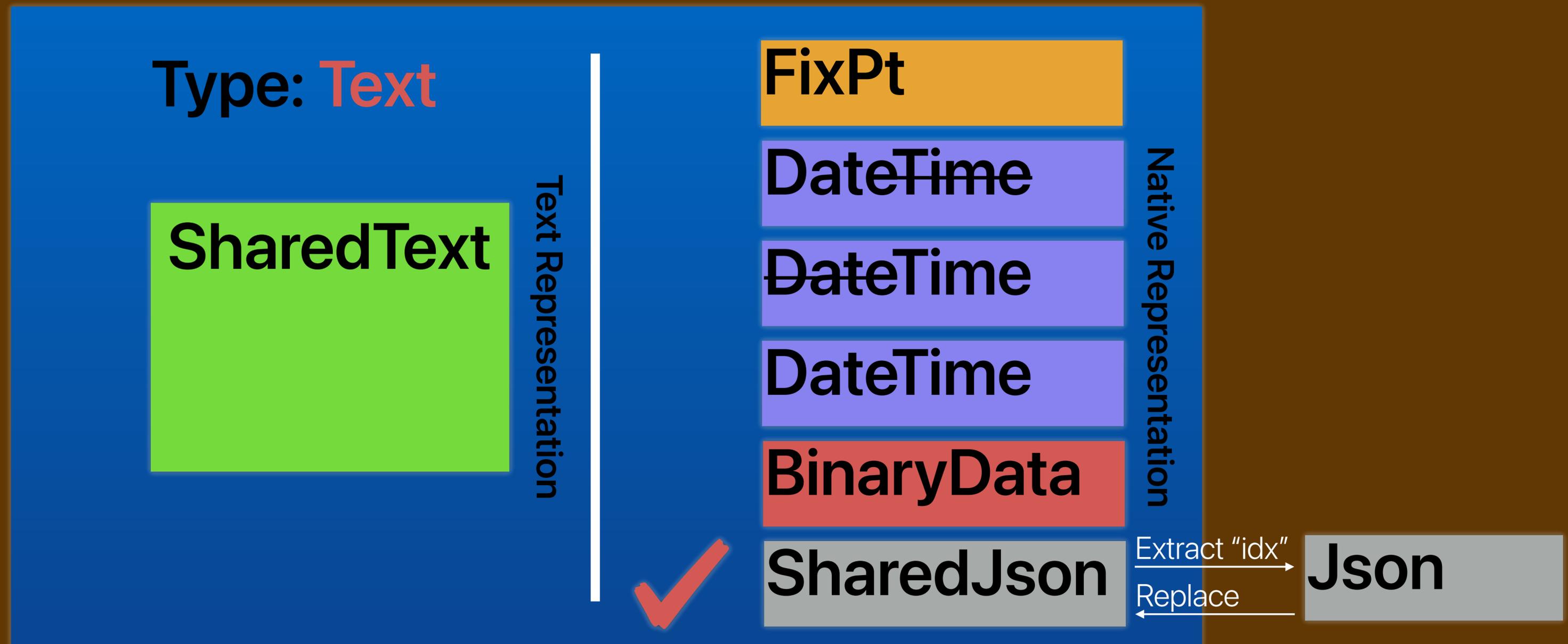
```
$var = JSONGetElement( txt ; "idx" )
```

JSON function requires getting the parsed JSON of the text



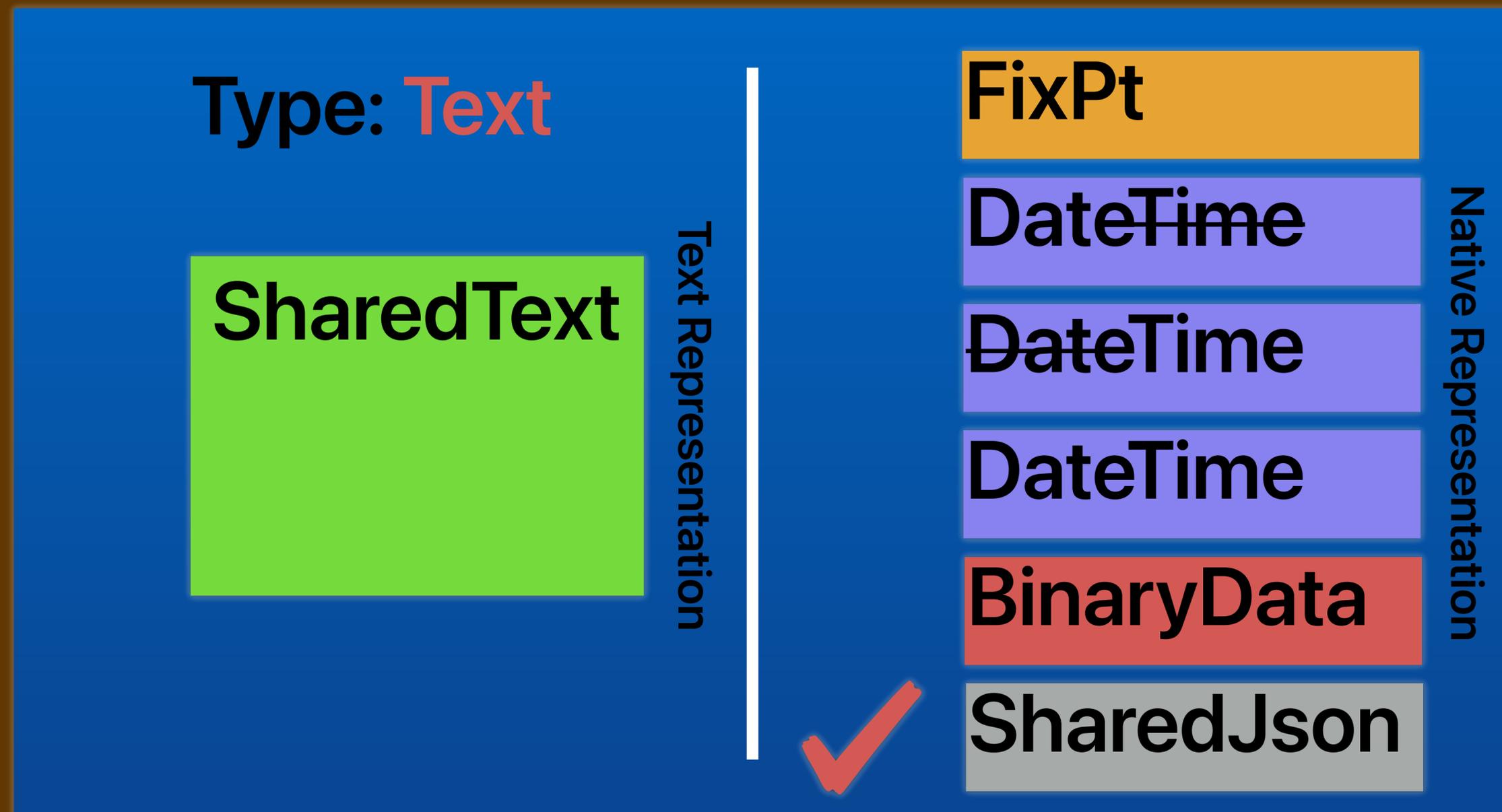
```
$var = JSONGetElement( txt ; "idx" )
```

Extract the "idx" portion of the JSON put it into this DBData, text is dropped



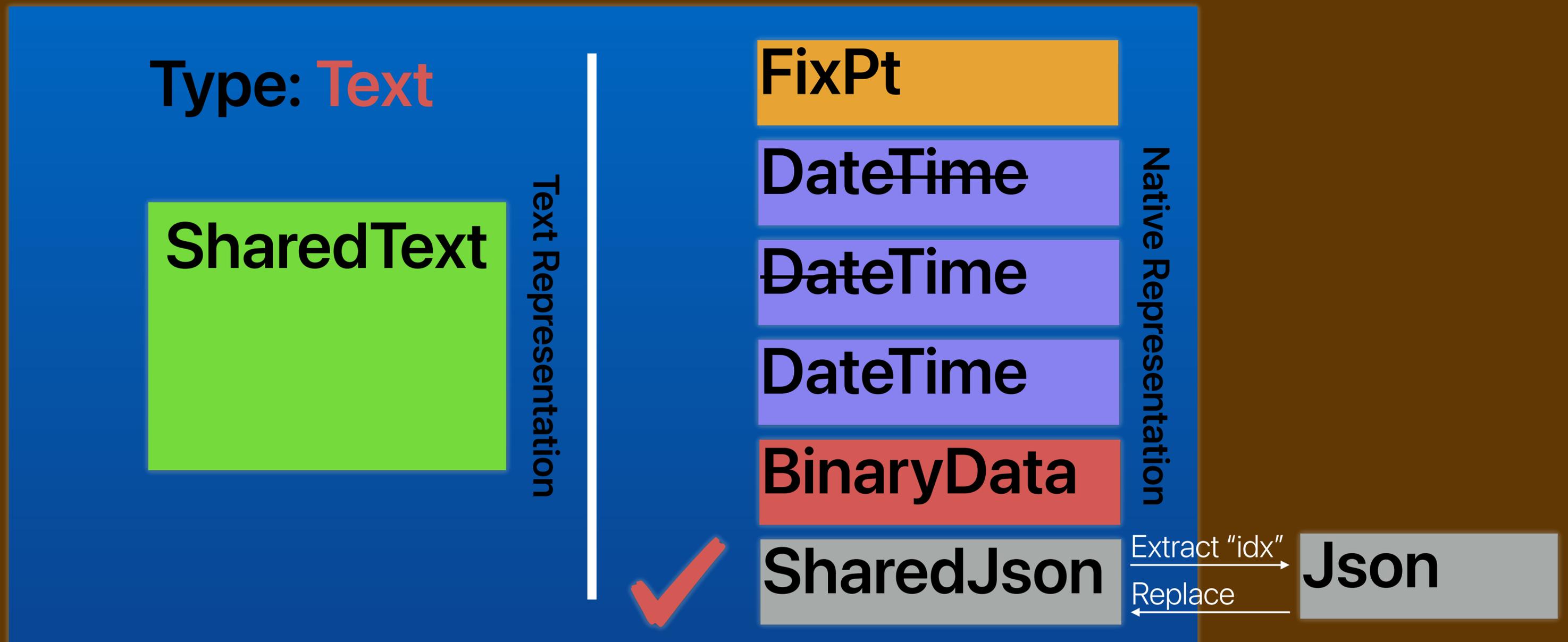
```
$var = JSONGetElement( txt ; "idx" )
```

Saving the results in the variable



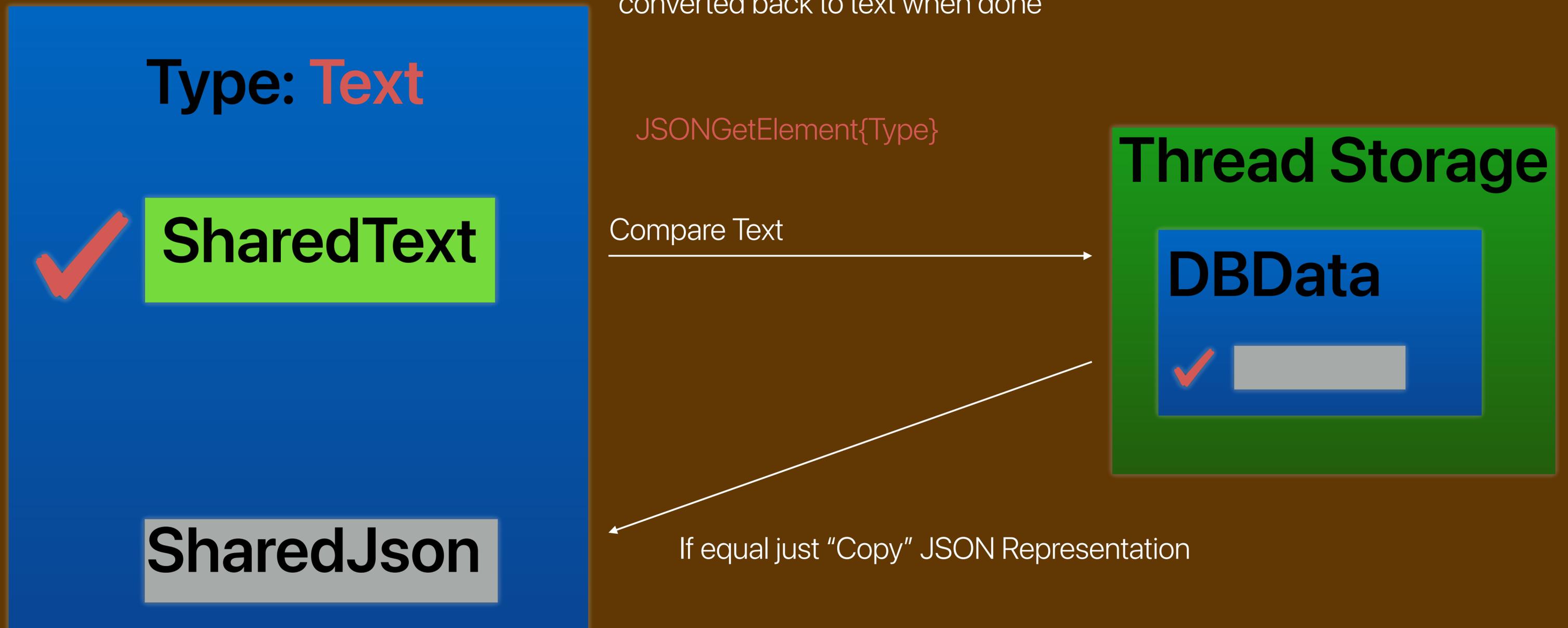
JSONGetElement(\$var ; "age")

Saving the results in the variable



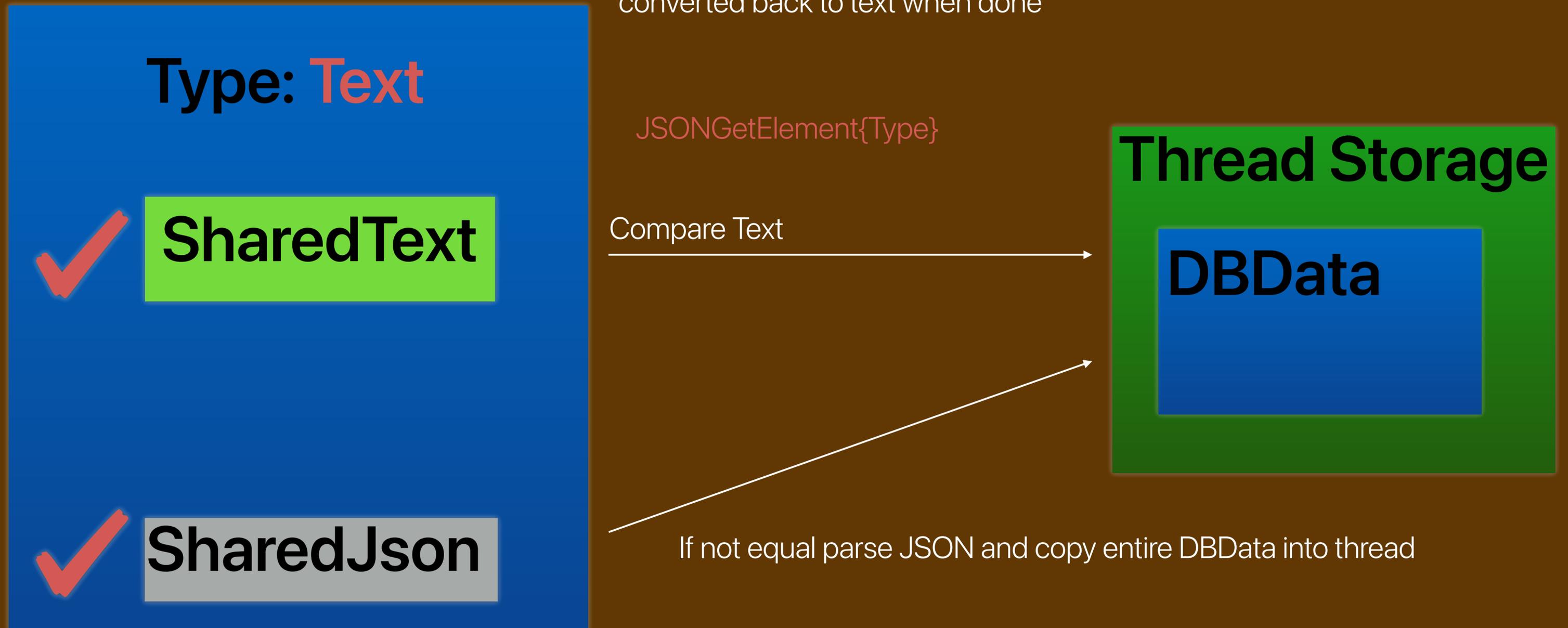
Version 3 Thread Caching

For JSONGetElement and JSONGetElementType we still have a thread cache that is checked.
converted back to text when done



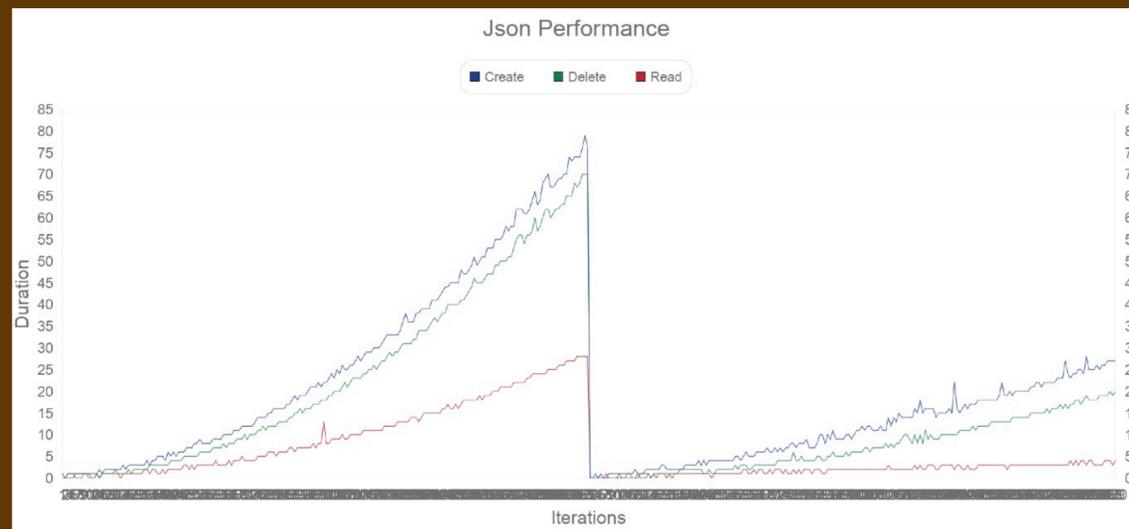
Version 3 Thread Caching

For JSONGetElement and JSONGetElementType we still have a thread cache that is checked.
converted back to text when done



Passive Performance improvements

- Without changing anything there will be performance improvements
 - BenchTest from 24U Software
 - json-speed-regression from Josh Willing
 - json loop test speed from Rob Hopkins
 - MBS JSON Benchmark from MonkeyBread
 - Json Performance from Pro 18 test file



Building JSON Array in local script variable	2225	4816	2.16 x faster
Accessing JSON Array Elements in local script variable	15739	155614	9.89 x faster
Building JSON Array in global variable	2222	4814	2.17 x faster
Accessing JSON Array Elements in global variable	15744	155644	9.89 x faster

JSON Speed Regression

Run Runtime in microseconds: 285,876
fmp_version: Pro 21.1.1

Run Runtime in microseconds: 38,581
fmp_version: Pro 22.0.1

7.4 x faster

json loop test speed v2	21.1.1	22.0.1	faster
parse using GetJSONElement	2669	167	16x
parse using JSONDeleteElement	2976	739	4.x
parse using GetValue	196	177	1.1x

MonkeyBread	21.1.1	22.0.1	faster
Build JSON Array	487	192	2.5x
Get Values from Array	229	35	6.5x

Active Performance improvements

- **JSONParse function**
 - Parses the input parameter returning the same text if valid or the JSON error string if not
 - Key thing it does is generate the native JSON representation in the DBData
 - Saving this result in a variable (not a field!) and reusing this variable will help a lot
- **JSONParsedState function**
 - Looks at the input parameter to see if it has native JSON as part of it. If it returns 0 it has not been parsed, -1 means it was parsed but is not valid JSON, other numbers match the values return by JSONGetElementType function for the root item

Active Performance improvements

Demo - Bad

```
1 Set Variable [ $name1 ; Value: JSONGetElement ( Demo::JSON1 ; "name" ) ]
2 Set Variable [ $name2 ; Value: JSONGetElement ( Demo::JSON2 ; "name" ) ]
3 Set Variable [ $state1 ; Value: JSONGetElement ( Demo::JSON1 ; "state" ) ]
4 Set Variable [ $state2 ; Value: JSONGetElement ( Demo::JSON2 ; "state" ) ]
```

No help from DBData or Cache

Demo - Better 

```
1 Set Variable [ $name1 ; Value: JSONGetElement ( Demo::JSON1 ; "name" ) ]
2 Set Variable [ $state1 ; Value: JSONGetElement ( Demo::JSON1 ; "state" ) ]
3 Set Variable [ $name2 ; Value: JSONGetElement ( Demo::JSON2 ; "name" ) ]
4 Set Variable [ $state2 ; Value: JSONGetElement ( Demo::JSON2 ; "state" ) ]
```

Help from Cache

Demo - Best 

```
1 Set Variable [ $json1 ; Value: JSONParse ( Demo::JSON1 ) ]
2 Set Variable [ $json2 ; Value: JSONParse ( Demo::JSON2 ) ]
3
4 Set Variable [ $name1 ; Value: JSONGetElement ( $json1 ; "name" ) ]
5 Set Variable [ $name2 ; Value: JSONGetElement ( $json2 ; "name" ) ]
6 Set Variable [ $state1 ; Value: JSONGetElement ( $json1 ; "state" ) ]
7 Set Variable [ $state2 ; Value: JSONGetElement ( $json2 ; "state" ) ]
```

Pure Native JSON

FileMaker Upgrade Tool Patching



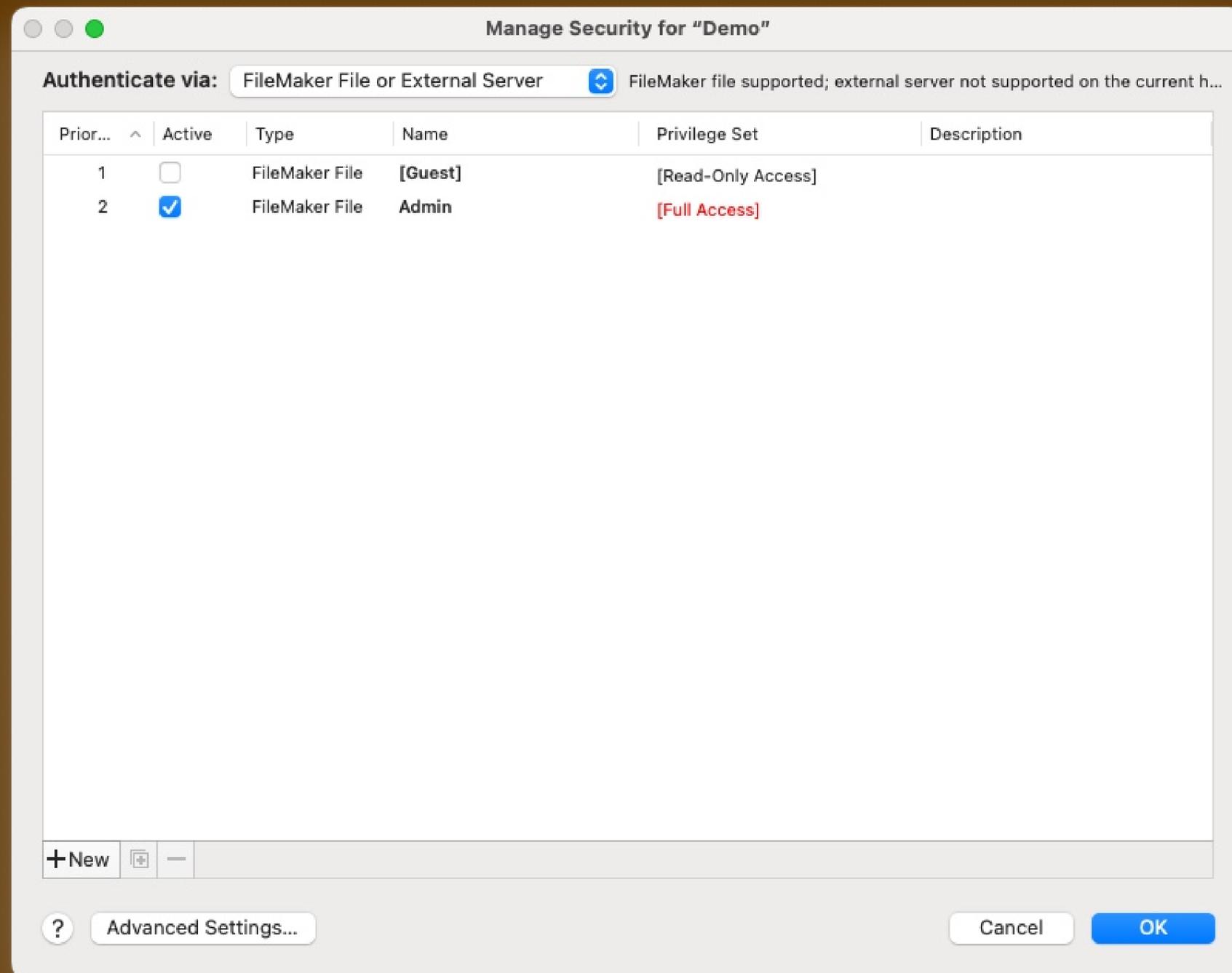
The command line tools

- FMDataMigration, FMDeveloperTool, and the FMUpgradeTool
- Now part of the FileMaker Server installation and found in the FileMaker Server/Database Server/bin folder
- All of them can load plug-ins using the `-plugin_folder` option so that calculations depending on plug-ins will be processed correctly
- Plug-ins that are built to run on server should work but check with your plug-in vendor
- A separate command line tool installer is planned for the future

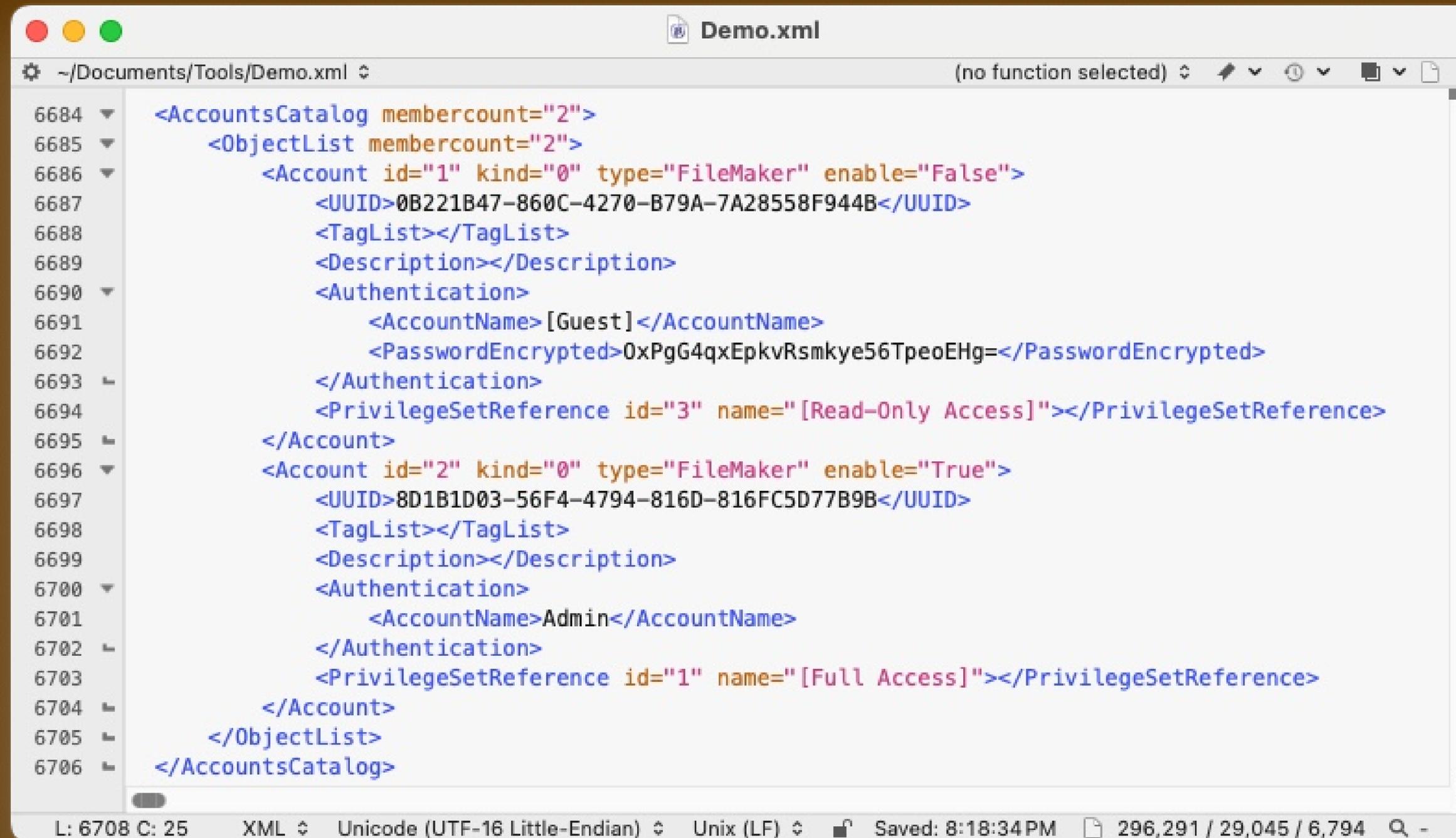
What are catalogs in FileMaker

- How all (with a few exceptions) schema is stored in FileMaker
- All the catalogs have the same base C++ class so they all work similarly
- Two main flavors of catalogs: item by item catalogs and unified catalogs
- Catalogs are very independent, especially locking in multiuser cases

The Account Catalog



The Account Catalog



```
6684 <AccountsCatalog membercount="2">
6685   <ObjectList membercount="2">
6686     <Account id="1" kind="0" type="FileMaker" enable="False">
6687       <UUID>0B221B47-860C-4270-B79A-7A28558F944B</UUID>
6688       <TagList></TagList>
6689       <Description></Description>
6690       <Authentication>
6691         <AccountName>[Guest]</AccountName>
6692         <PasswordEncrypted>0xPgG4qxEpkvRsmkye56TpeoEHg=</PasswordEncrypted>
6693       </Authentication>
6694       <PrivilegeSetReference id="3" name="[Read-Only Access]"></PrivilegeSetReference>
6695     </Account>
6696     <Account id="2" kind="0" type="FileMaker" enable="True">
6697       <UUID>8D1B1D03-56F4-4794-816D-816FC5D77B9B</UUID>
6698       <TagList></TagList>
6699       <Description></Description>
6700       <Authentication>
6701         <AccountName>Admin</AccountName>
6702       </Authentication>
6703       <PrivilegeSetReference id="1" name="[Full Access]"></PrivilegeSetReference>
6704     </Account>
6705   </ObjectList>
6706 </AccountsCatalog>
```

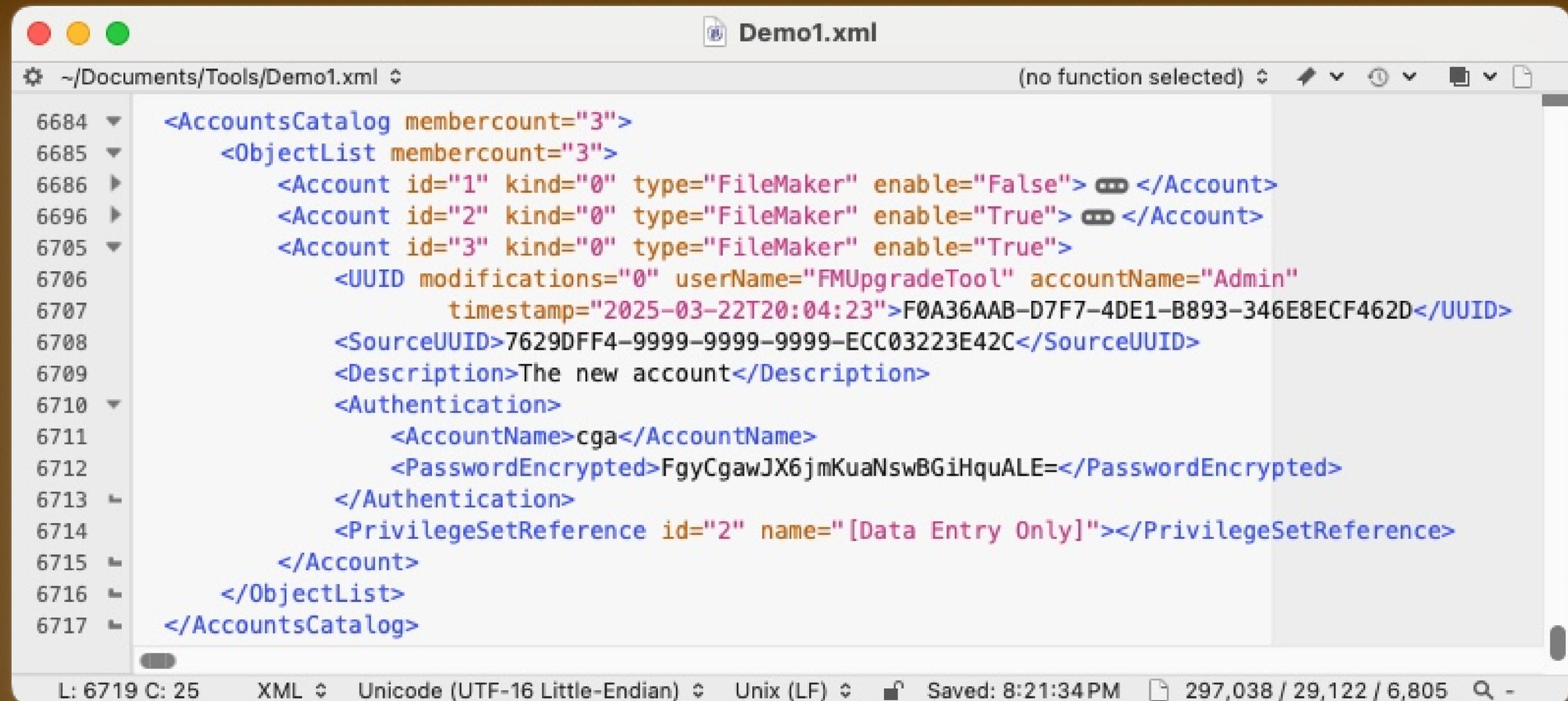
L: 6708 C: 25 XML Unicode (UTF-16 Little-Endian) Unix (LF) Saved: 8:18:34 PM 296,291 / 29,045 / 6,794

Adding with the upgrade tool

```
1 <?xml version="1.0"?><FMUpgradeToolPatch version="2.2.2.0"><Structure><AddAction>
2   <AccountsCatalog><ObjectList>
3     <Account kind="0" type="FileMaker" enable="True">
4       <UUID>7629DFF4-9999-9999-9999-ECC03223E42C</UUID>
5       <Description>The new account</Description>
6       <Authentication>
7         <AccountName>cga</AccountName>
8         <Password>bbb</Password>
9       </Authentication>
10      <PrivilegeSetReference id="2" name="[Data Entry Only]"/>
11    </Account>
12  </ObjectList></AccountsCatalog>
13 </AddAction></Structure></FMUpgradeToolPatch>
```

L: 14 C: 1 XML Unicode (UTF-8) Unix (LF) Saved: 8:07:26 PM 521 / 50 / 14 100%

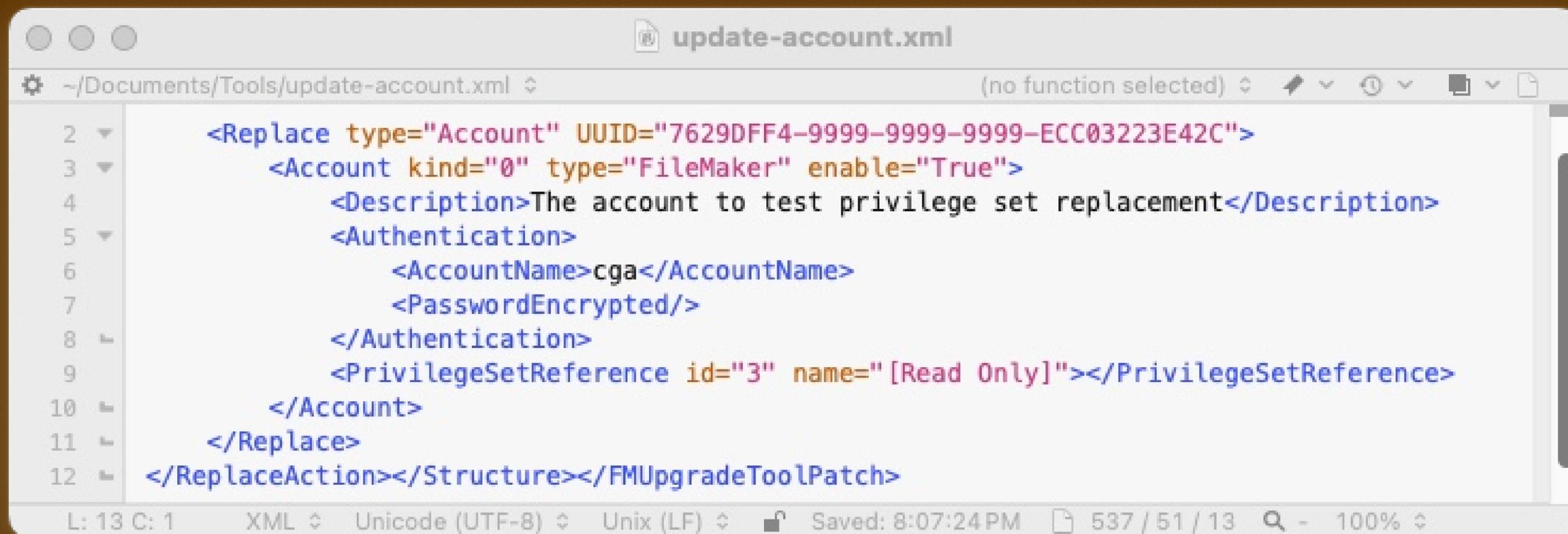
Adding with the upgrade tool



```
6684 <AccountsCatalog membercount="3">
6685   <ObjectList membercount="3">
6686     <Account id="1" kind="0" type="FileMaker" enable="False"> </Account>
6696     <Account id="2" kind="0" type="FileMaker" enable="True"> </Account>
6705     <Account id="3" kind="0" type="FileMaker" enable="True">
6706       <UUID modifications="0" userName="FMUpgradeTool" accountName="Admin"
6707         timestamp="2025-03-22T20:04:23">F0A36AAB-D7F7-4DE1-B893-346E8ECF462D</UUID>
6708       <SourceUUID>7629DFF4-9999-9999-9999-ECC03223E42C</SourceUUID>
6709       <Description>The new account</Description>
6710       <Authentication>
6711         <AccountName>cga</AccountName>
6712         <PasswordEncrypted>FgyCgawJX6jmKuaNswBGiHquALE=</PasswordEncrypted>
6713       </Authentication>
6714       <PrivilegeSetReference id="2" name="[Data Entry Only]"></PrivilegeSetReference>
6715     </Account>
6716   </ObjectList>
6717 </AccountsCatalog>
```

L: 6719 C: 25 XML Unicode (UTF-16 Little-Endian) Unix (LF) Saved: 8:21:34 PM 297,038 / 29,122 / 6,805

Replacing with the upgrade tool

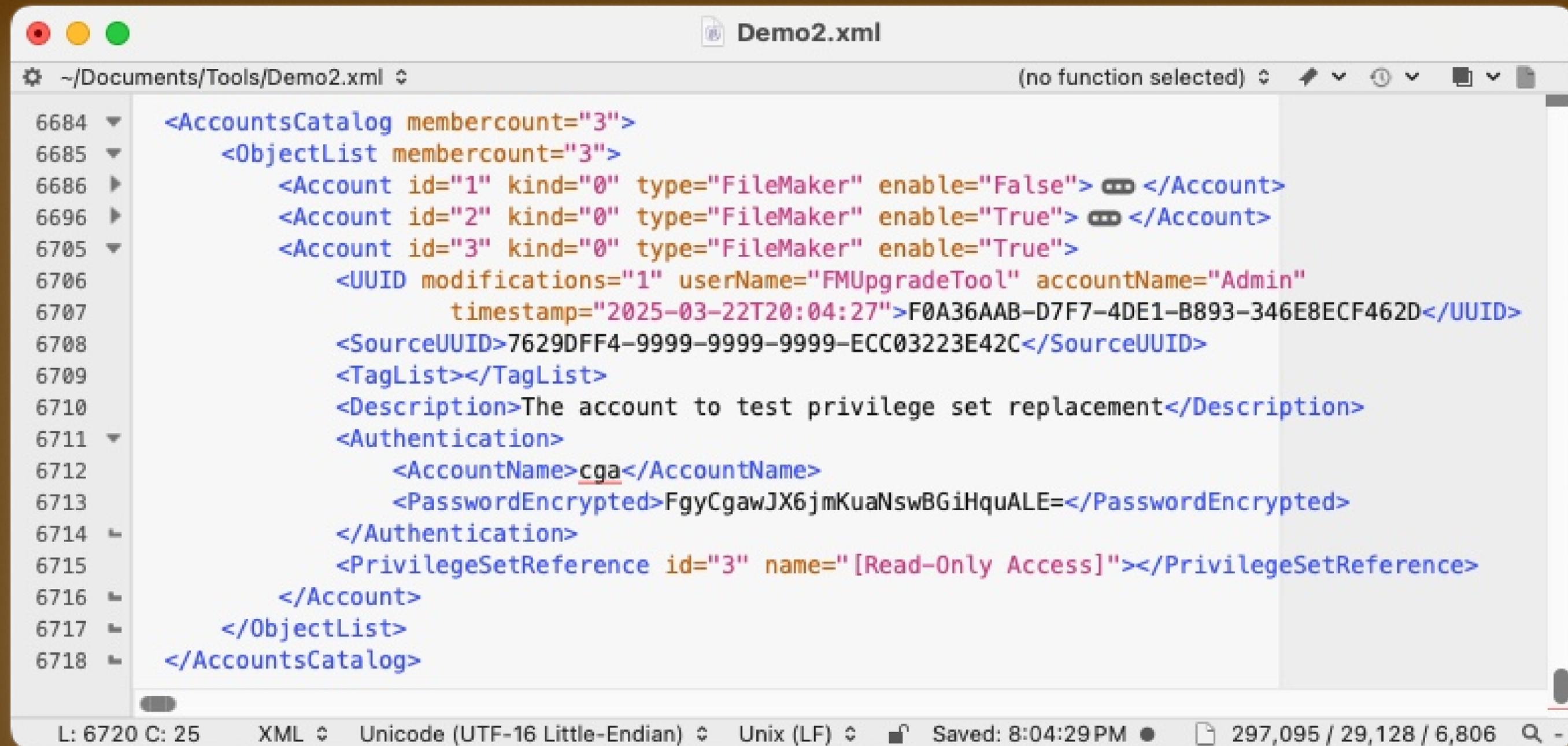


The image shows a code editor window titled "update-account.xml". The editor displays XML code for replacing an account. The code is as follows:

```
2 <Replace type="Account" UUID="7629DFF4-9999-9999-9999-ECC03223E42C">
3   <Account kind="0" type="FileMaker" enable="True">
4     <Description>The account to test privilege set replacement</Description>
5     <Authentication>
6       <AccountName>cga</AccountName>
7       <PasswordEncrypted/>
8     </Authentication>
9     <PrivilegeSetReference id="3" name="[Read Only]"></PrivilegeSetReference>
10   </Account>
11 </Replace>
12 </ReplaceAction></Structure></FMUpgradeToolPatch>
```

The editor interface includes a toolbar with various icons and a status bar at the bottom. The status bar shows the current cursor position (L: 13 C: 1), the file type (XML), the encoding (Unicode (UTF-8)), the line ending (Unix (LF)), the save status (Saved: 8:07:24 PM), and the zoom level (100%).

Replacing with the upgrade tool



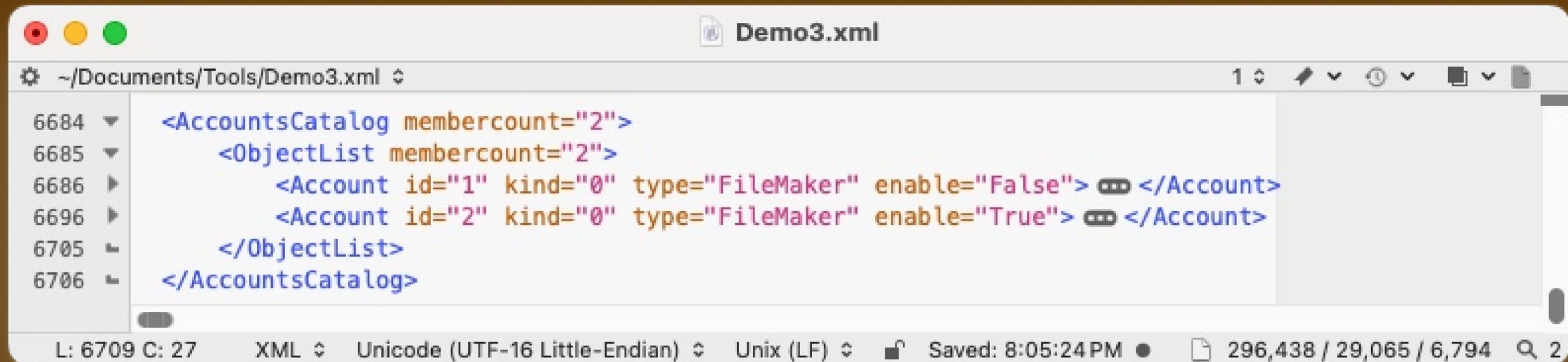
```
6684 <AccountsCatalog membercount="3">
6685   <ObjectList membercount="3">
6686     <Account id="1" kind="0" type="FileMaker" enable="False"> </Account>
6696     <Account id="2" kind="0" type="FileMaker" enable="True"> </Account>
6705     <Account id="3" kind="0" type="FileMaker" enable="True">
6706       <UUID modifications="1" userName="FMUpgradeTool" accountName="Admin"
6707         timestamp="2025-03-22T20:04:27">F0A36AAB-D7F7-4DE1-B893-346E8ECF462D</UUID>
6708       <SourceUUID>7629DFF4-9999-9999-9999-ECC03223E42C</SourceUUID>
6709       <TagList></TagList>
6710       <Description>The account to test privilege set replacement</Description>
6711       <Authentication>
6712         <AccountName>cga</AccountName>
6713         <PasswordEncrypted>FgyCgawJX6jmKuaNswBGiHquALE=</PasswordEncrypted>
6714       </Authentication>
6715       <PrivilegeSetReference id="3" name="[Read-Only Access]"></PrivilegeSetReference>
6716     </Account>
6717   </ObjectList>
6718 </AccountsCatalog>
```

L: 6720 C: 25 XML Unicode (UTF-16 Little-Endian) Unix (LF) Saved: 8:04:29 PM 297,095 / 29,128 / 6,806

Deleting with the upgrade tool

```
delete-account.xml
~/Documents/Tools/delete-account.xml (functions)
1 <?xml version="1.0"?><FMUpgradeToolPatch><Structure><DeleteAction>
2   <ItemReference UUID="7629DFF4-9999-9999-9999-ECC03223E42C" type="AccountReference"/>
3 </DeleteAction></Structure></FMUpgradeToolPatch>
L: 4 C: 1 XML Unicode (UTF-8) Unix (LF) Saved: 8:05:33 PM 202 / 18 / 4 100%
```

Deleting with the upgrade tool



The screenshot shows a text editor window titled "Demo3.xml" with the following XML content:

```
6684 <AccountsCatalog membercount="2">
6685   <ObjectList membercount="2">
6686     <Account id="1" kind="0" type="FileMaker" enable="False"> ☒ </Account>
6696     <Account id="2" kind="0" type="FileMaker" enable="True"> ☒ </Account>
6705   </ObjectList>
6706 </AccountsCatalog>
```

The status bar at the bottom indicates the file is in XML format, using UTF-16 Little-Endian encoding with Unix (LF) line endings. It also shows the file size as 296,438 / 29,065 / 6,794 bytes and a search count of 2.

Futures

- Update the XML grammar
 - Unify the catalog name
 - Object List and TagList usage
 - Turn bitmaps into boolean names
 - Other name changes to be more consistent
 - Make it easier for LLM consumption and generation
- Create the DTD (Document Type Definition)
- Recreate a FileMaker file from the XML output
- More documentation

Other changes



Third Party library updates

- boost 1.82 -> 1.87
- date 3.01 -> 3.03
- icu 74.2 -> 76.1
- imagemagick 7.1.0.18 -> 7.1.1.43
- iodbc 3.52.13 -> 3.52.16
- jwpc 1.1 -> 1.2
- jwtcpp 0.6.0 -> 0.7.0
- libantler 3 -> 4.13.1
- libcurl 8.4.0 -> 8.10.1
- libevent 2.0.21 -> 2.1.12
- libxml2 2.10 -> 2.13.5
- libxslt 1.1.37 -> 1.1.42
- NodeJS 18.20 -> 22.14
- opencv 4.6.0 -> 4.11.0
- openssl rebuilt with optimization flags
- pdfium 5304.v2 -> 6957.v1
- pdfwriter 3.7 -> 4.6.6
- speller 19.2 -> 22
- thrift 0.17.0 -> 0.21.0
- tomcat 10.1.31 -> 10.1.33
- vaadin 8.18.0 -> 8.27.4
- zbar 0.23.92 -> 0.23.93

OData changes

- More aggregate support
- Better container support
- Value list support (also in xDBC)
- Webhook support for notifications, mainly for Studio and Connect
- Future for the ODBC and JDBC replacement
- Preferred data API to use with FileMaker
- FM Data API will go the way of the XML API, pretty much ignored

Recent vRev Release

- 19.6.4 / 20.3.2
 - Security issues from Apple
- 21.1.2
 - Security reconnect issues
- 21.1.3
 - Server scripts memory leak & CPU starvation
- 21.1.4
 - Simultaneous server scripts & security caching issues

Questions and answers

