Custom Search with Delphi & Rubicon

Oregon Delphi User Group
August 17, 2022

href
tools corp.

Today’s Talk

- Overview of a sample project
- See the End Result
- Zoom in: nuts & bolts of the Custom Search feature
- Links for source code, etc.
- Questions, Discussion

Optional Topics

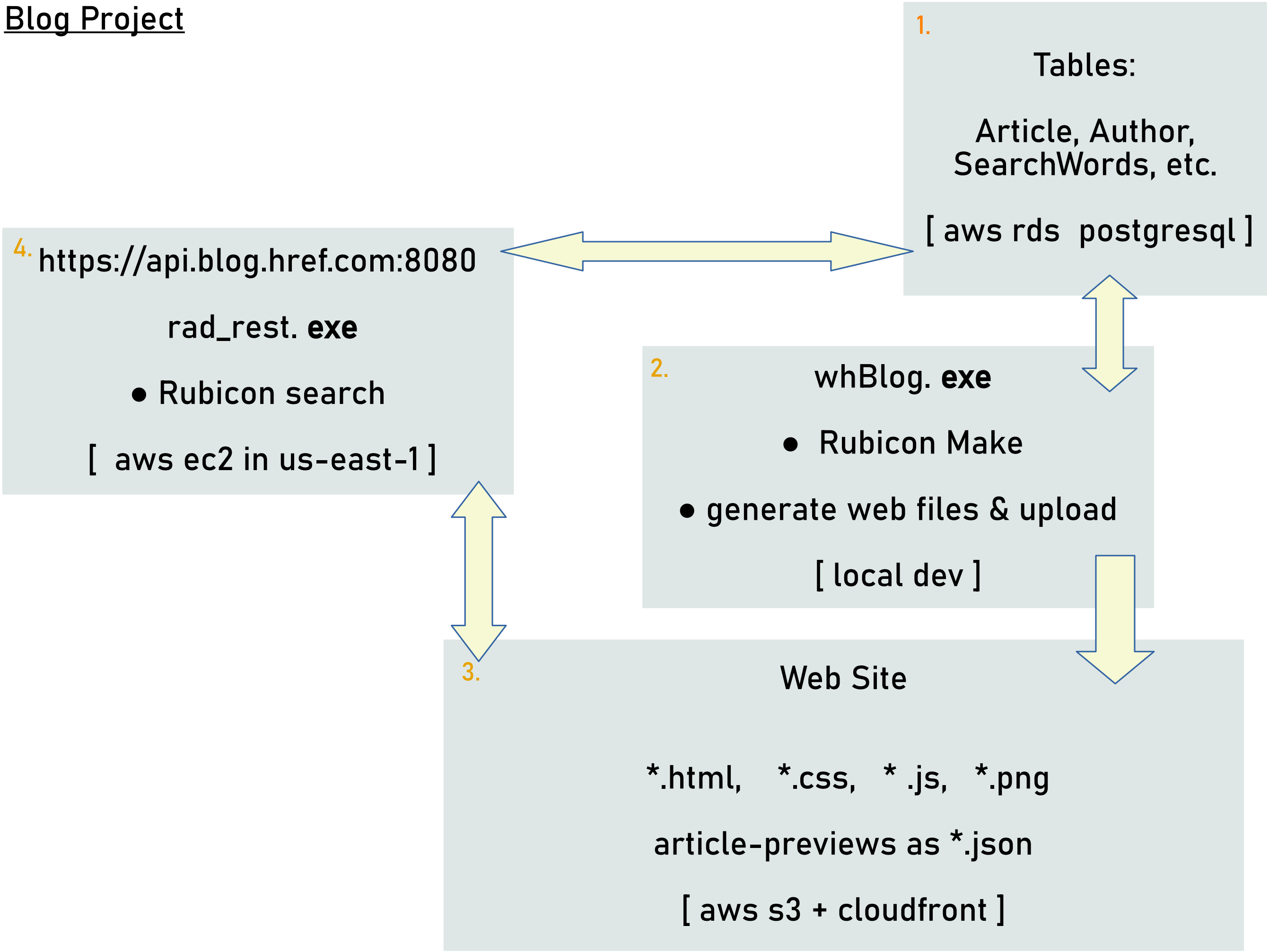
- Tools: Postman, Browser DevTools, Dezign for Databases
- XHR
- ZaphodsMap for configuration
- CC Image Search
- Javascript highlighting of matching words in HTML

- CORS
- Go vs WebHub
- Responsive Design with Bootstrap
- CloudFront distribution origins and behaviors
- Browser local storage

Sample Project Overview

- blogger.com → blog.href.com
- Internship project, “full stack” learning
- Rubicon search as “RAD REST” with results over https

Blog Project



1.

Tables:

Article, Author, SearchWords, etc.

[aws rds postgresql]

4. https://api.blog.href.com:8080

rad_rest. **exe**

- Rubicon search

[aws ec2 in us-east-1]

2. whBlog. **exe**

- Rubicon Make
- generate web files & upload

[local dev]

3. Web Site

*.html,   *.css,   * .js,   *.png

article-previews as *.json

[aws s3 + cloudfront]

AWS: Amazon Web Services

- RDS: Relational Database Service (managed postgresql)
- S3: Simple Storage Service (web files)
- CloudFront: Highly optimized Content Delivery over https
- EC2: Electronic Compute Cloud (Windows server)

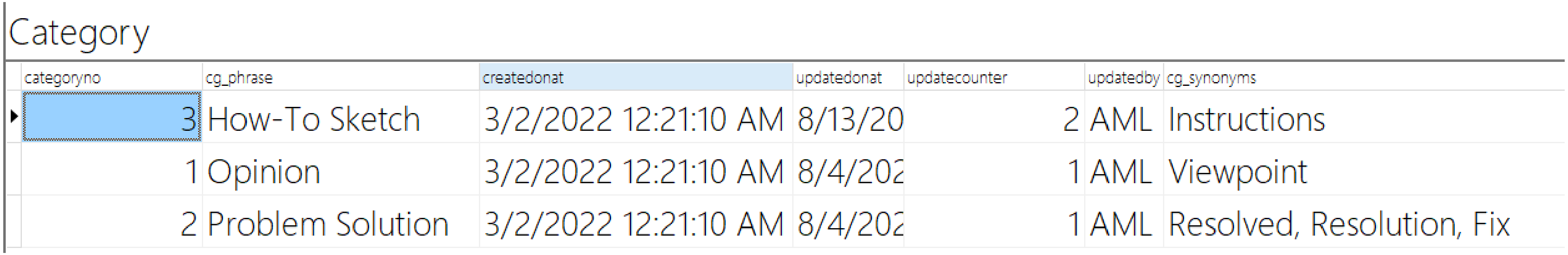
See the End Result

- https://blog.href.com
- https://blog.href.com/blog_search.html

Database Design

https://blog.href.com/postgres_design/erdiagram.html

Synonyms



Category

	categoryno	cg_phrase	createdonat	updatedonat	updatecounter	updatedby	cg_synonyms
▶	3	How-To Sketch	3/2/2022 12:21:10 AM	8/13/20	2	AML	Instructions
	1	Opinion	3/2/2022 12:21:10 AM	8/4/202	1	AML	Viewpoint
	2	Problem Solution	3/2/2022 12:21:10 AM	8/4/202	1	AML	Resolved, Resolution, Fix

Rubicon Full Text Search for Delphi

- Since Delphi 1.0   16-bit Windows !   AnsiString!   English!!
  author: Deven Hickingbotham

- Originally supported Paradox and dBase via BDE
- Added various component bridges….   FireDAC in year 2022
- Requires a unique integer identifier for each text location

Optimized, How?

- The WORDS Table

WORD	Count	Location List as compressed bitmap

Rubicon: “Make” once, then Search

- TrbMake: Make the “WORDS” table (SearchWords table)
  based on “TEXT” (Article table)

- TrbSearch

Create View as Select ...

- procedure TfmRubiconMake.Establish_MultiTable_View;

```
(select STRING_AGG( kw_phrase, '',''' )
         from Keyword k
        inner join Keyword_Relation on KR_ArticleNo = a.Articleno

        WHERE (K.keywordno  = KR_KeywordNo)

        ) as ArticleKeywords
```

Detour: DFM vs PAS

- RAD
- Object Inspector assistance
- Ideal for visual controls
- Components must be installed; not backwards compatible.

- All details visible in PAS
- IDE assistance at the dot
- Ideal near bleeding edge (Delphi beta) and with any project that needs to work in multiple versions of Delphi.

Detour: Components at Runtime

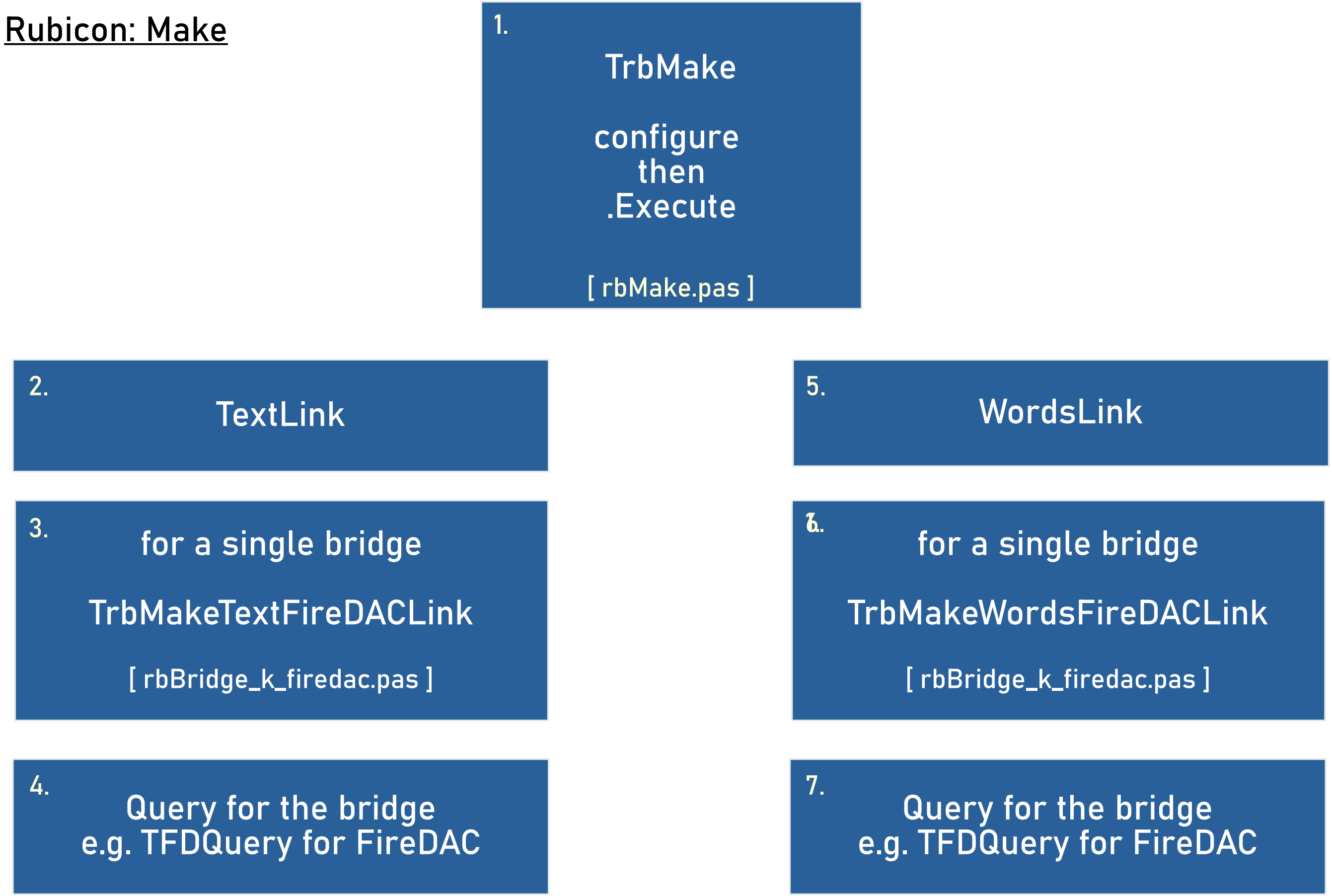
```
MySearch := TrbMake.Create( Self ); // pass in Owner component or nil
MySearch.Name := ‘MySearch’;  // unique at runtime



FreeAndNil(  MySearch );
```

Rubicon Make: overview

Rubicon: Make



1.
TrbMake

configure
then
.Execute

[rbMake.pas]

2.
TextLink

3.
for a single bridge

TrbMakeTextFireDACLink

[rbBridge_k_firedac.pas]

4.
Query for the bridge
e.g. TFDQuery for FireDAC

5.
WordsLink

6.
for a single bridge

TrbMakeWordsFireDACLink

[rbBridge_k_firedac.pas]

7.
Query for the bridge
e.g. TFDQuery for FireDAC

Rubicon Make: in detail

- whBlog.dpr
- unit whBlog_fmRubiconMake;
- procedure TfmRubiconMake.ActionMakeIndexExecute

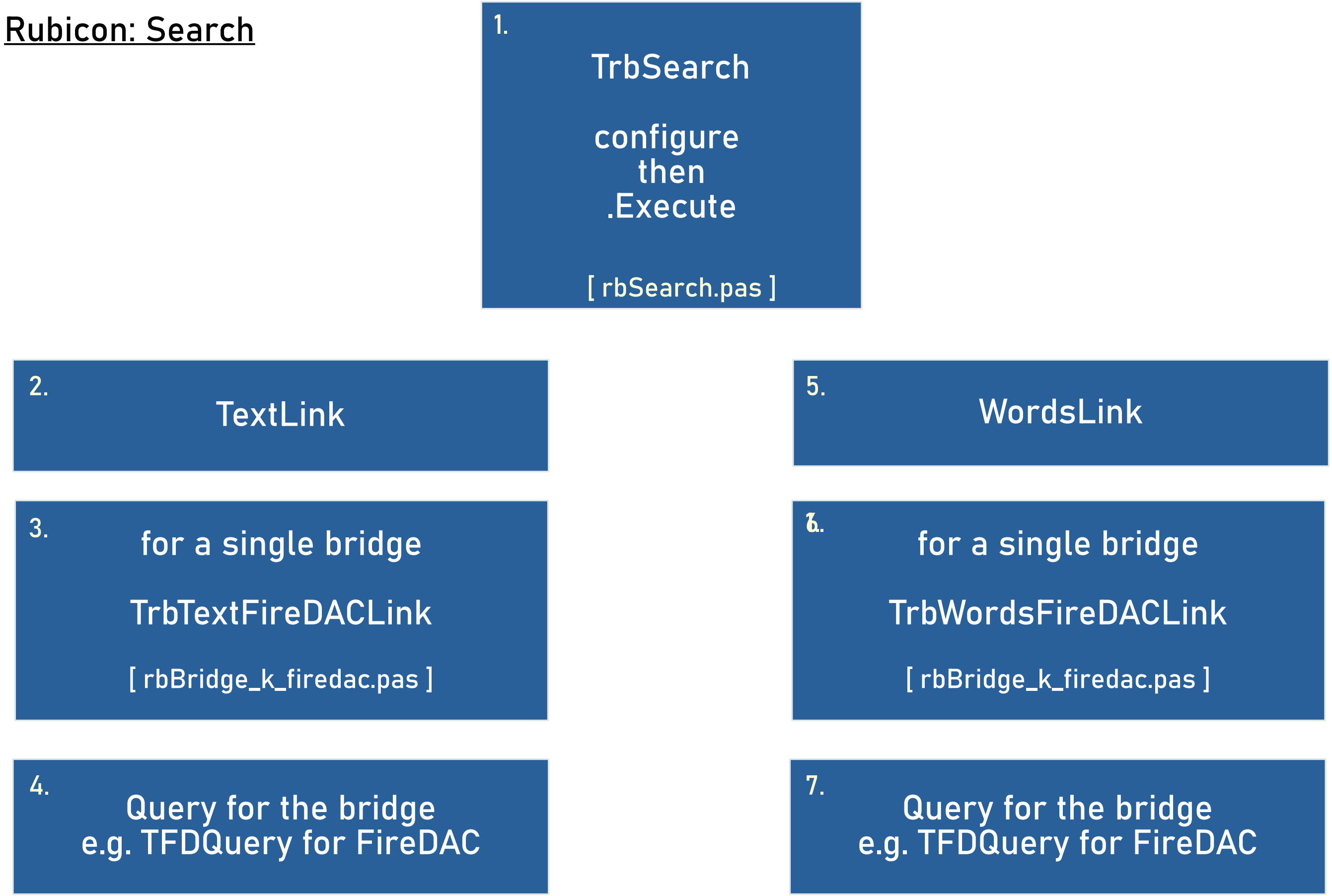
Rubicon Search: over https

- REST API designed with Postman

- XHR requests visible in DevTools in Chrome, Firefox, etc.

Rubicon Search: in PAS

Rubicon: Search



1.
TrbSearch

configure
then
.Execute

[rbSearch.pas]

2.
TextLink

3.
for a single bridge

TrbTextFireDACLink

[rbBridge_k_firedac.pas]

4.
Query for the bridge
e.g. TFDQuery for FireDAC

5.
WordsLink

6.
for a single bridge

TrbWordsFireDACLink

[rbBridge_k_firedac.pas]

7.
Query for the bridge
e.g. TFDQuery for FireDAC

Rubicon Search: in detail

- **RAD_REST.dpr**

- **ServerMethodsUnit1**.pas

  function TServerMethods1.updateSearchQueue(const PostedFormAsJson: string): integer;

  function TServerMethods1.Articles(const iSearchQueueNo: integer): TJSONObject;

- **RubiconModuleUnit1**.pas

  function SearchArticles(var rec: **TSearchQueueRec**): string;

OnPreviewLocation

To restrict by article publication date

```
rbSearch.OnPreviewLocation := PreviewArticleNo;
```

Blog Project Source is on SourceForge

- Rubicon Make
- Source\WHApps\DB Examples\whBlog\whBlog.dproj

- Rubicon Search
- Source\DelphiApps\Blog_Rad_Rest\rad_rest.dpr

svn checkout url

svn.code.sf.net/p/webhubdemos/code/trunk/webhubdemos

Links

- Rubicon Showcase Video
  https://youtu.be/vRTneMRC4yQ (skip ad, then it plays)
- **ZaphodsMap Configuration**
  https://sourceforge.net/p/zaphodsmap/wiki/Home/
- Windows upload to S3
  https://www.msp360.com/explorer/windows/amazon-s3/
- Amazon Web Services
  https://aws.amazon.com
- Go language
  https://www.usegolang.com/
- Postman
  https://www.postman.com
- CodeSite Logging
  https://raize.com/codesite/
- Bootstrap
  https://getbootstrap.com/docs/5.2/getting-started
- HREF Tools Corp.
  https://www.href.com/contact
  https://forum.href.com
  https://www.href.com/rubicon

www.href.com/rubicon

Please refer
your friends & colleagues.

Thank you.