

December, 1999

Advisor Answers

Looking for an Index by Key

Visual FoxPro 6.0, 5.0, 3.0 and FoxPro 2.x

Q: How do I check if a field has an index? I would like to know if my REPNUM field (or any field) has an index but I can't find a way of doing this.

I have been trying TAG(), TAGNO(), TAGCOUNT(), CDX() and other functions but can't find an answer.

—Morys Azcarate (via Advisor.COM)

A: As you note, there are quite a few functions that provide information about index tags. However, there's no function that takes a field name and tells you whether there's a tag whose key is that field. Fortunately, it's not hard to write one, though brute force is required.

Due to FoxPro's richness, there are a number of ways to write this function, but the simplest uses KEY(). Here's the function:

```
* FindTag.PRG
* This function finds a tag with a given key.
* If it's found, it returns the name of the tag.
* If the table has no such tag, it returns the empty string.
```

```
LPARAMETERS cKey, cAlias
  * cKey = the key to find
  * cAlias = the alias in which to look.
  *           If omitted, use the current alias.
```

```
LOCAL cUseAlias
```

```
* Check parameters
```

```
IF VarType(cKey) <> "C" OR EMPTY(cKey)
```

```
  * Might want an assertion here, too.
```

```
  RETURN "" && Nothing to do
```

```
ENDIF
```

```
IF VarType(cAlias) <> "C" OR EMPTY(cAlias)
```

```
  * Use current alias
```

```
  cUseAlias = ALIAS()
```

```
  IF EMPTY(cUseAlias)
```

```
    RETURN "" && Nothing to do
```

```
  ENDIF
```

```
ELSE
```

```
  * Use the one passed
```

```
  cUseAlias = cAlias
```

```
ENDIF
```

```
* If alias is good, switch to it, saving work area
```

```
IF USED(cUseAlias)
```

```
  LOCAL nOldSelect
```

```
  nOldSelect = SELECT()
```

```

        SELECT (cUseAlias)
ELSE
    RETURN ""  && Nothing to do
ENDIF

* Now time to look for the key
LOCAL nTagNum, cTagName
nTagNum = 1
DO WHILE NOT EMPTY(KEY(nTagNum)) ;
    AND NOT (UPPER(KEY(nTagNum)) == UPPER(cKey))
    nTagNum = nTagNum + 1
ENDDO
* Don't need to check because TAG() returns
* empty string if no such tag
cTagName = TAG(nTagNum)

* Restore original work area
SELECT (nOldSelect)

RETURN cTagName

```

It takes two parameters, the index key to look for (which doesn't have to be just a field name – a compound key works, too) and the alias of the table. The alias is optional; if it's omitted, the function looks in the current work area. After confirming that the parameters are valid, the function loops through the tags of the table and compares their keys to the key it's looking for. When it finds a matching tag, it returns the name of that tag.

This version searches the tags of all open index files: its structural index, that is, the .CDX file that has the same name as the table, and all other open .CDX files and .IDX files as well. You'll need some additional checking following the DO WHILE loop if you want to limit the search to the structural index only.

The function has one weakness. It returns the empty string whenever it fails, no matter what the reason for the failure. You get the same return value whether you passed a bad parameter or there is no tag with the indicated key.

There are some alternatives. You could return .F. for an error condition and the empty string only when there's no matching tag, for example, but that requires the code that uses the function to check the type of the return value every time you call it. In Visual FoxPro, a better solution is to return a .NULL. value, because this doesn't introduce a new data type for the error condition. ISNULL() and EMPTY() can be used to distinguish between a missing index key and an error. Another possibility is to use VFP's ERROR command to raise an error for a bad parameter and let the error handler deal with it. In some applications, that might be the right choice, especially if you want to view this function as an extension of the VFP language. Since passing the wrong parameter is actually a bug in the calling code, you might consider using an assertion to inform the developer about this problem during testing. But you should be aware that an assertion is a testing tool only. To make the function fail-safe at runtime, as well, you need a return value that indicates an error and to handle this return value accordingly in the calling code.

Given these alternatives, I preferred to return the empty string every time. You may want to make a different choice.

Finally, although the version of FindTag above works only in VFP 6, it's quite easy to modify it to work in earlier versions of VFP and even in FoxPro 2.x. Depending on the FoxPro version, use TYPE() rather than VarType(), PARAMETER instead of LPARAMETER, PRIVATE instead of LOCAL and SYS(14) instead of KEY(). You'll find this version on this month's Professional Resource CD.

-- Tamar