The Input Transform: An Example

by George Timson

Introduction

Let's re-focus on the VA FileMan Input Transform. Remember first that "input transform" is a misnomer, since Input Transforms can do much besides "transform" what the user enters. In general, the Input Transform *transforms* and *validates* input, rejecting what it determines to be invalid. Let's examine a real-world example of how a M[UMPS] programmer can use the Input Transform to enhance FileMan's capabilities. Before we're through, we'll have touched on such diverse topics as cross-references, documentation, and the 1995 Standard.

SSN — the simple part

Let's consider the 9-digit *social security number*. Here is a data element which really can usefully be "transformed" to make data entry easier. Since SSNs are usually seen with punctuation in them (e.g., 081-77-2222), why not let an unsophisticated user enter the data in that familiar way, even if it takes more keystrokes? We will then want to remove from the incoming value any spaces or dashes the user enters, in order to achieve consistency and economy of storage. Remember that every FileMan Input Transform operates on the M[UMPS] variable "X." So we can simply begin our SSN Input Transform with

```
    SET X=$TR(X," -/")
```

to strip the punctuation characters " ", "-", and "/" from the input, X.

So much for "transforming." Now we begin "validating." Remember that the Input Transform must `KILL X` if it is determined to be invalid data. So, to make sure that the SSN is nine digits in length, as required, we add

```
     SET X=$TR(X," -/")
     KILL:X'?9N X
```

And we may also add another simple check, knowing that the U.S. Social Security Administration does not assign SSN's beginning with 8 or 9:

```
SET X=$TR(X," -/")
KILL:X'?9N X    IF $D(X)
KILL:X>799999999 X
```

I hope no one needs to be told why that "`I $D(X)`" is there! Elementary MUMPS, so far.

Uniqueness

Several newer database management systems offer a field-definition feature that is missing in VA FileMan: the attribute of *uniqueness*. That is, it is sometimes desirable to be able to declare that, over the universe of all entries, each particular field value may only occur once. In a database of citizens of the United States, social security number is an obvious example of uniqueness since, supposedly, no two people can have the same social security number. (Actually, most people who work with large databases have learned very well that the Social Security Administration often *does* inadvertently assign two individuals the same number, but we'll ignore that fact for the purposes of this discussion.)

Now let's recall from the last FileMan column (*M Computing*, March 1997), that at the moment that we want to check the validity or format of a FileMan input, X, the variable "DA" will hold the internal entry number of the entry for which the data is being entered. If we are looking at entry number "999999" in the PATIENT file, and want to find out if we can legally enter the social security number "081772222" for him, then

```
X = "081772222" and
DA = "999999"
```

Well, it will be easy to check whether X is already a social security number in the database, just so long as there is a *cross-reference* on social security number. In the PATIENT File used by the VA medical applications, for example, such a cross-reference is defined, and therefore there will be defined a Global node

```
^DPT("SSN",X,N)
```

if X exists as a social security number somewhere in the database. "N" will be the Internal Entry Number of the patient who has this SSN. In order to build our "uniqueness test," we are going to *rely* on the presence of this cross-reference. There is a simple way to insure against someone removing this "SSN" cross-reference (although the VA developers haven't done it!), and that is to edit the cross-reference as follows:

```
Select UTILITY OPTION:
CROSS-REFERENCE A FIELD
MODIFY WHAT FILE: PATIENT
Select FIELD: SOCIAL SECU-
RITY NUMBER

CURRENT CROSS-REFERENCES:
  ...
7 REGULAR 'SSN' INDEX OF
FILE
  ...

Choose E (Edit)/D
(Delete)/C(Create): EDIT
WHICH NUMBER: 7

NO-DELETION MESSAGE: Used
to insure uniqueness of
SSN's
DESCRIPTION:
        1>
```

FileMan will not allow deleting a cross-reference that has a "NO-DELETION MESSAGE" attached to it. This message line is also a simple documentation feature for cross-references; the NO-DELETION MESSAGE (if any) shows up wherever cross-references are listed. The "no-deletion" feature is used less than it should be; there are plenty of crucial cross-references sitting around in large FileMan databases just ripe for deletion. Maybe the "documentation" aspect scares developers away from filling in this message; we know how developers hate to write in English!

Verifying fields

All right, enough preaching. Let's go back to the Input Transform. The easiest way to make sure that we don't assign an already-used X to our current DA patient would be to expand our Input Transform to read:

```
SET X=$TR(X," -/")
KILL:X'?9N X
IF $D(X) KILL:X>799999999 X
IF $D(X)
IF $D(^DPT("SSN",X))
KILL X
```

But there is a reason why the test should be more complicated; we should really be making sure that, if there is a descendant from `^DPT("SSN",X)`, it is equal to DA. In other words, if `^DPT("SSN",X,N)` exists, N had better equal DA.

Why would the SSN Input Transform for entry DA be executed when

```
^DPT("SSN",X,DA)
```

already exists? Well, it *is* possible that a user might be typing the same value in "on top of itself" for the same patient. But the real reason is that the Input Transform is executed in another context besides input. It is used in the Utility Option called "VERIFY FIELDS." This Verify option (a typical feature of any DBMS) allows one to re-check retrospectively *all* the SSN's on file, and the checking is done, in part, by re-running the SSN Input Transform on each SSN value that exists. You could say, from an object oriented perspective, that the Input Transform "encapsulates" the implementation of the SSN data field and therefore has usefulness apart from the actual entry of data.

Now if we use the Verify option and run through all DAs, find their X SSN's, and execute

```
...IF $D(^DPT("SSN",X))
KILL X
```

what will happen? Right! Every X we find will already exist in the cross-reference, and we'll be told that each entry's SSN value "fails the Input Transform." Now, in some cases it would be very hard to re-write an Input Transform so that it is "encapsulated" to run in the future. One thinks of dates that "cannot be less than today." But the social security number code can easily be improved. How about:

```
SET X=$TR(X," -/")
KILL:X'?9N X
IF $D(X) KILL:X>799999999 X
IF $D(X)
NEW SSN
MERGE SSN=^DPT("SSN",X)
KILL SSN(DA)
IF $D(SSN)
KILL X
```

This says that, if there's anything hanging below X other than DA, it's an error condition. We use the local array "SSN" to do the examining, so that a `KILL` won't be a `KILL` to the database.

Parsing 1995 MUMPS

But what about that MERGE command? It's only valid M[UMPS] code as of the recent (1995) Standard. This leads me to my last digression, which is to point out that VA FileMan Version 21 has recently been patched to allow 1995 syntax like "MERGE." The patch (of the "DIM*" routines) has been distributed inside the VA, and outside the VA it is available to anyone who can download from the "Hardhats" Web site. Stop by at:

www.hardhats.org/fileman/FMrepairs.html

and click on "DIM* patch"! **M**

George Timson (gtimson@pacbell.net) was the principal author of VA FileMan. He is now an independent consultant living in Berkeley, CA.

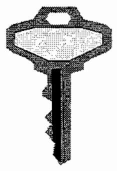
MAXIMUS seeks qualified professionals with background in MUMPS to support a new project in Cairo, Egypt, focusing on health care delivery systems. Experience with VA DHCP/VISTA is a plus. Short term and full time positions available, based in Cairo, including implementation, technical and software specialists. MAXIMUS, a world leader in program mgt. and consulting services, offers a comprehensive benefits pkg., including 401k (with matching corp. contrib.) and Flex. Benefits Account (Sect. 125) plans. Fax or send resume to:

8150 Leesburg Pike, Suite 1000
Vienna, VA 22182
(Attn: Jeremy Ames)
Fax (703) 748-1465

Visit MTA's Web site today for information on:

- First ever MTA salary survey
- Ordering books and other MTA publications
- Renewing your membership
- News

www.mtechnology.org



MTA Job Referral Service

Your key to a successful job search!
Your key to finding a qualified employee!

EmployEEs:

- Free listing in **JRS** to **MTA** Members
- Receive names of **ALL** employers in **JRS**
- Your resume will be sent out for 6 months

Employ**ERs**:

- Employers may use the **JRS** for a fee
- Receive an initial packet of at least 40 to 50 resume briefs
- Your job opening(s) sent to **ALL** job candidates

Let MTA help you unlock doors.
To receive your key, contact Marlo Brown:
M Technology Association - North America
phone: (301) 431-4070 fax: (301) 431-0017
email: **MTA@mtechnology.org**