# VA FileMan DDE Utility

# Tutorial


**April 2023**

**Department of Veterans Affairs (VA)**

**Office of Information and Technology (OIT)**

**Software Product Management (SPM)**

# Revision History

| Date | Revision | Description | Author |
|---|---|---|---|
| 04/03/2023 | 1.0 | Initial release of VA FileMan 22.2 Advanced User Manual. | Virtual Patient Record (VPR) Development Team<br>VistA Infrastructure Shared Services (VISS) Development Team |

# Table of Contents

## Table of Figures

# 1 Introduction

The ENTITY (#1.5) file can map Veterans Health Information Systems and Technology Architecture (VistA) files and fields to any data model, including common standards such as Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR) or InterSystems' Summary Document Architecture (SDA). The VA FileMan DDE utility uses the ENTITY (#1.5) file as a template for producing eXtensible Markup Language (XML) or JavaScript Object Notation (JSON) output from VistA data.

## 1.1 What is an Entity?

An Entity is simply a collection of data elements, similar to a data class in an object-oriented system. If a specific data model is being followed, that model determines the tag names and allowable values of each element. The Entity assigns each tag to a file and/or field in VistA, transforming the data as needed to meet any requirements or constraints of the model.

Usually, an Entity returns a single record from a VistA file, accepting a record Internal Entry Number (IEN) and returning the defined fields as the data elements. An Entity can accept any string as its input, however; for example, an Entity could be created to take a VistA name string and return its components as separate elements within a group.

Like classes, entities can be nested. Commonly used data elements, such as locations, for example, can have their own Entity that accepts a pointer to the HOSPITAL LOCATION (#44) file and returns a group of attributes (code and description, specialty, phone#, etc.) from that file, which can then be embedded in other entities.

## 1.2 VA FileMan DDE Utility

DDE is the VA FileMan utility that uses an Entity to build XML or JSON output. The ENTITY (#1.5) file is stored in the **^DDE** global. Entry points $$GET1 and GET in routine **DDE** are supported calls that can be used to return VistA data, one or more records at a time, according to the specification defined by the Entity.

The exercises in this tutorial use the **Data Mapping** [DDE ENTITY MAPPING] options on the **Other Options** [DIOTHER] menu in VA FileMan.

**REF:** For details on how to use these calls and options, see the "Data Mapping" section in the *VA FileMan Developer's Guide*.

# 2 Lesson 1: Create an Entity

In this lesson you will learn how to:

- Create an Entity.
- Define basic information for the Entity.
- Indicate the source for the Entity's data.

## 2.1 Exercise 1: Create a Test Entity

### 2.1.1 Step 1. Create a New Entity

**Figure 1: Exercise 1—Creating a New Entity: ZZZ NEW PERSON**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
  Are you adding 'ZZZ NEW PERSON' as a new ENTITY (the 279TH)? No// Y <Enter>
(Yes)
```

ENTITY (#1.5) is a shared file similar to the OPTION (#19) file, so you should use a unique name beginning with your project's assigned VistA namespace when creating a new Entity.

A ScreenMan form opens to guide you through the necessary components to create an Entity (Figure 2). The first page contains basic information about the Entity and the source of the data it is returning.

### 2.1.2 Step 2. Complete Page 1 of Form

**Figure 2: Exercise 1—Page 1: Completing Page 1 of Form**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                    Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ NEW PERSON
 DISPLAY NAME: NewPerson

 DEFAULT FILE: 200

      SORT BY:                                  DATA MODEL:
    FILTER BY:                                   READ ONLY:
       SCREEN:

QUERY ROUTINE:

  DESCRIPTION: <Enter>

_______________________________________________________________________________

Exit    Save    Next Page    Previous Page    Refresh    Quit

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND:
```

Use the **DISPLAY NAME** for the outer tags in the XML or JSON output; if this Entity is to be compliant with a specific data model, use the name given to the object or class in that model.

Enter the number of the primary VistA source file for the data at the "DEFAULT FILE:" prompt.

Press **<Enter>** to bypass the prompts in the middle of the screen for now; these will be addressed in a later lesson.

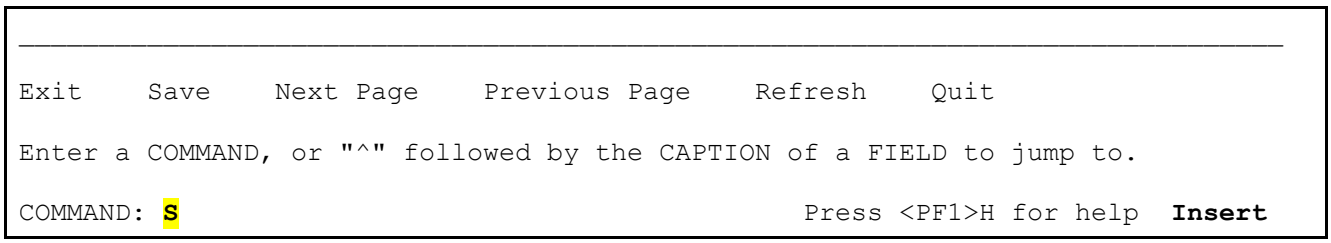
When you arrive at the "DESCRIPTION:" prompt, press **<Enter>** to open the word-processing field (Figure 3). It is *highly recommended* to explain the purpose of the Entity and the expected input value.

**Figure 3: Exercise 1—Entering Entity Description**

```
  1>This Entity is for demonstration purposes only. It accepts a pointer to
  2>the NEW PERSON (#200) file and returns various properties.
  3><Enter>
EDIT Option:
```

**Exit** the word-processing editor, then enter **S** (**Save**) at the Command prompt to save your work (Figure 4).

**Figure 4: Exercise 1—Saving Entity Details**




End of Exercise 1.

# 3 Lesson 2: Add Simple Properties

In this lesson you will learn how to add simple properties to an Entity. These are properties that return a single value, regardless of their source data type in a VA FileMan data dictionary. A property's Item Type tells DDE how to use the VistA field value to create the property. DDE uses the VA FileMan DIQ utility to retrieve the value of the specified field, regardless of its data type.

Simple properties include the following Item Types:

- **I**—Record Identifier (ID). The DDE automatically uses the record identifier as the property value
- **F**—Fixed (static or constant) String.
- **S**—Simple VistA Field.

## 3.1 Exercise 2.1: Add an ID property

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again for these exercises and select your test Entity if you exited the form after Lesson 1. Use the arrow keys to drop down to the Command prompt at the bottom of the page, then enter **N** (**Next**) to go to Page 2 of the form (Figure 5).

### 3.1.1 Step 1. Create New Property

Enter the new property's name under the **Item** heading, use "**IEN**" for this exercise, then answer **YES** when asked if adding a new Item.

**Figure 5: Exercise 2.1—Page 2: Creating a New Property: IEN**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
IEN


_____________________________________________________________________________
  Are you adding 'IEN' as a new ITEM? No// Y
```

### 3.1.2 Step 2. Assign Sequence Number and Item Type

Press **<Enter>** to jump to the **Seq** column and enter a number to indicate the order in which these properties should be returned in the results; for this tutorial you will simply start with **1**.

**Figure 6: Exercise 2.1—Page 2: Assigning Sequence Number and Item Type to IEN Property**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
IEN                             1       I
```

Press **<Enter>** to jump to the **Type** column and enter the letter **I (**Record ID). A popup dialog opens containing the attributes that may be entered for an ID item (Figure 7).

### 3.1.3 Step 3. Complete Attributes for Item Type

**Figure 7: Exercise 2.1—Page 2 Popup Dialog: Completing Attributes for Item Type for the IEN Property**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                             .
.      GET ACTION:                                                            .
.                                                                             .
.                                                                             .
.OUTPUT TRANSFORM:                                                            .
. INPUT TRANSFORM:                                                            .
.                                                                             .
.                                                                             .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
______________________________________________________________________________

Close    Refresh

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: Close
```

DDE uses the selected record identifier as the value for an ID property, so nothing more needs to be entered; the **GET ACTION** and the transform fields will be addressed in a later lesson. Press **<Enter>** to get to the Command prompt to **Close** the dialog and return to the form.

End of Exercise 2.1.

## 3.2 Exercise 2.2: Add a Fixed String Property

Continue in the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option on Page 2 of the form to add another property (Figure 8).

### 3.2.1 Step 1. Create New Property

Enter a new property called "**Source**" under **IEN** in the **Item** column and respond **YES** when asked if adding a new item, as in the previous exercise (Exercise 2.1).

**Figure 8: Exercise 2.2—Page 2: Creating a New Property: Source**

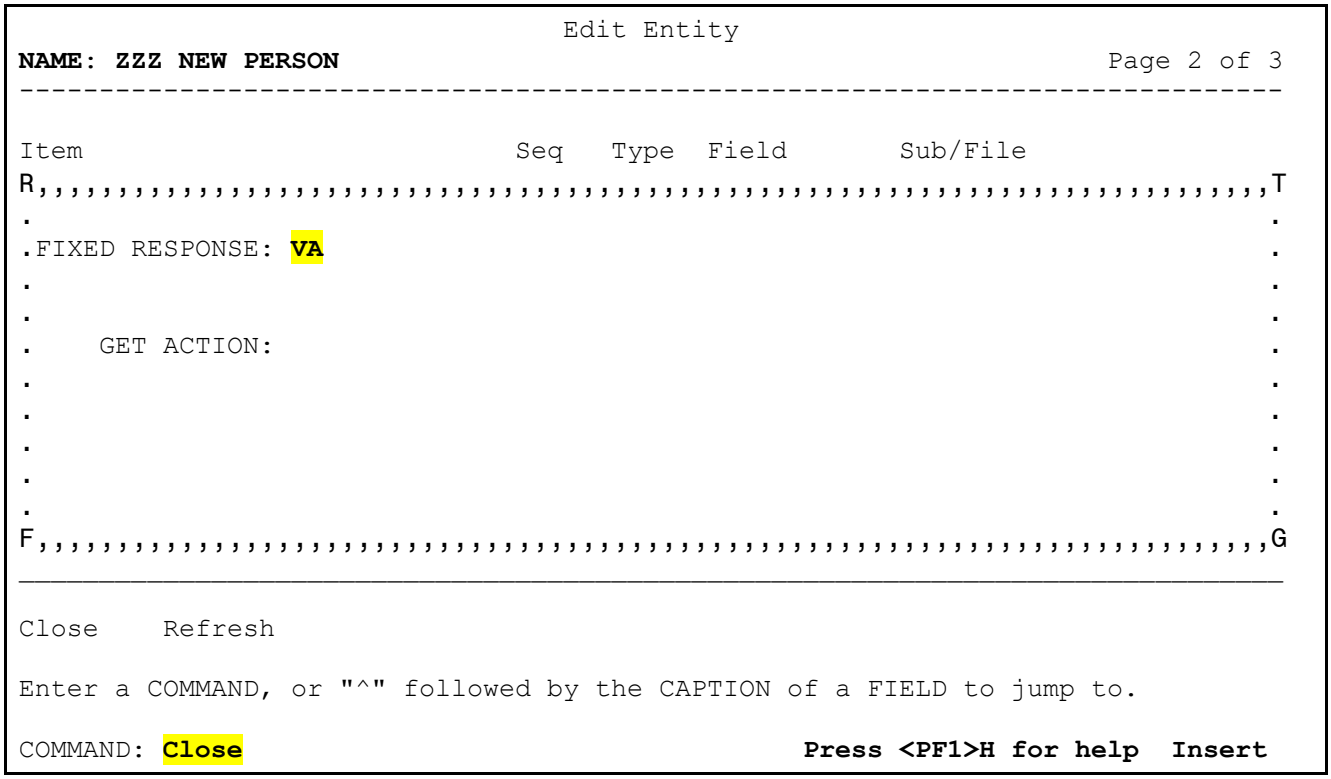
```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type  Field       Sub/File
IEN                             1        I
Source                          2        F
```

Again, press **<Enter>** to jump to the **Seq** column and enter the number **2**. Press **<Enter>** again to jump to the **Type** column and enter the letter **F** (Fixed String); pressing **<Enter>** one more time opens the "Fixed String" dialog (Figure 9).

### 3.2.2 Step 2. Complete Attributes for this Item Type

**Figure 9: Exercise 2.2—Page 2 "Fixed String" Dialog: Completing Attributes for this Item Type**




Use a Fixed String (**F**) **Type** to return a static string of text. For this exercise, enter "**VA**" at the **FIXED RESPONSE** field.

Use of the **GET ACTION** attribute will be addressed in a later lesson. Press **<Enter>** to get to the Command prompt, and again to **Close** the dialog and return to the form.

End of Exercise 2.2.

## 3.3 Exercise 2.3: Add Simple Field Properties

Continue in the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option on Page 2 of the form to add additional properties. In this exercise, you will add a variety of properties that pull data from the default NEW PERSON (#200) file.

### 3.3.1 Step 1. Create New Property

Enter a new property called "**Name**" in the **Item** column and respond **YES** when asked if adding a new item, as in the previous exercises.

**Figure 10: Exercise 2.3—Page 2: Creating a New Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq    Type  Field       Sub/File
IEN                               1        I
Source                            2        F
Name                              3        S
```

Again, press **<Enter>** to jump to the **Seq** column and enter the number **3**. Press **<Enter>** to jump to the **Type** column and enter the letter **S** (Simple Field); pressing **<Enter>** one more time open the "Simple Field" dialog (Figure 11).

### 3.3.2 Step 2. Complete Attributes for this Item Type

**Figure 11: Exercise 2.3—Page 2 "Simple Field" Dialog: Completing Attributes for this Item Type**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                             Seq    Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                             .
.           FIELD#: .01                      FILE#: 200                       .
.     EXT PTR LKUP:                                                           .
.   INTERNAL VALUE:                                                           .
.                                                                             .
.       GET ACTION:                                                           .
.                                                                             .
.OUTPUT TRANSFORM:                                                            .
. INPUT TRANSFORM:                                                            .
.                                                                             .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
------------------------------------------------------------------------------

Close    Refresh

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: Close                                  Press <PF1>H for help  Insert
```

The dialog opens with the cursor on the "FILE#:" prompt, which is set to the **DEFAULT FILE** value from Page 1 of the form (**200**). Press **<Enter>** to jump over to the "FIELD#:" prompt and enter **.01** to return the value of the NAME (#.01) field in the NEW PERSON (#200) file. DDE automatically returns the external form of the field value.

The other attributes here will be addressed in other exercises. Press **<Enter>** to get to the Command prompt to **Close** the dialog and return to the form. You can see that the **Field** and **File** numbers are now displayed (Figure 12).

**Figure 12: Exercise 2.3—Page 2: Verifying Field and File Numbers**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                             Seq    Type  Field       Sub/File
IEN                              1        I
Source                           2        F
Name                             3        S   .01         200
```

### 3.3.3 Step 3. Create Four More Simple Properties

Repeat Steps 1 and Step 2 to create the following properties of various data types as shown, for use throughout this tutorial:

- **Gender**
- **LastSignOn**
- **Language**
- **State**

**Figure 13: Exercise 2.3—Page 2: Creating Four More Simple Properties**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type   Field       Sub/File
IEN                             1        I
Source                          2        F
Name                            3        S    .01         200
Gender                          4        S    4           200
LastSignOn                      5        S    202         200
Language                        6        S    200.07      200
State                           7        S    .115        200
```

### 3.3.4 Step 4. Use Extended Pointer Reference

Still on Page 2 of the form, use the arrow keys to move the cursor to the **State** property in the **Item** list. Press **<Enter> three** times to move to the **Type** column and open the "Simple Field" edit dialog (Figure 14).

**Figure 14: Exercise 2.3—Page 2 "Simple Field" Dialog: Using Extended Pointer Reference**

```
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                              .
.           FIELD#: .115                    FILE#: 200                         .
.     EXT PTR LKUP: 1                                                          .
.   INTERNAL VALUE:                                                            .
.                                                                              .
.       GET ACTION:                                                            .
.                                                                              .
.OUTPUT TRANSFORM:                                                             .
. INPUT TRANSFORM:                                                             .
.                                                                              .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

DDE returns the standard external format of a field value, unless otherwise specified; for pointers, the external form of the **.01** field in the pointed-to file is returned. Use the **EXT PTR LKUP** (Extended

Pointer Lookup) attribute to return a different field. For this exercise, enter a **1** to return the ABBREVIATION (#1) field from the STATE (#5) file instead of the usual **#.01** field.

**Close** this dialog and return to Page 2 of the form.

### 3.3.5 Step 5. Save Changes and Exit

Use the arrow keys to move the cursor down to the Command prompt, then enter **S** to **Save** your work. Enter **E** to **Exit** the form and editor.

End of Exercise 2.3.

# 4 Lesson 3: Test the Entity

In this lesson, you will learn how to test your Entity.

## 4.1 Exercise 3: $$GET1^DDE

### 4.1.1 Step 1. Exit VA FileMan to Programmer Mode

Exit VA FileMan to programmer mode.

### 4.1.2 Step 2. Test Entity Using $$GET1^DDE

Run the $$GET1^DDE function to return the results of a single instance of an Entity as a string:

**Figure 15: Exercise 3—Testing Entity: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":"VA", "Name":"PROGRAMMER,ZZZ ONE", "Gender":"
MALE", "LastSignOn":"SEP 08, 2022@11:18:19", "Language":"ENGLISH", "State":"UT"}
>
```

This function accepts various input parameters but requires at a minimum the Entity to run and the identifier of the desired record in the source file. The default format for returning results is JSON, but XML is also available.

**REF:** For full details on using the DDE functions, see the *VA FileMan Developer's Guide*.

End of Exercise 3.

# 5 Lesson 4: Add Complex Properties

Sometimes a simple value for a field is not sufficient, especially for fields that point to another file where multiple attributes may be needed. In this lesson you will learn how to create and add complex properties.

Complex properties include the following Item Types:

- **E**—Embedded Entity.
- **C**—Complex Group of Items.

DDE is able to call itself recursively, allowing complex properties to be defined as entities themselves. This is especially useful for fields that point to common reference files, such as locations or coding systems, as these entities can then be re-used. Simple properties can also be collected into a complex property group.

## 5.1 Exercise 4.1: Create an Entity for a Pointer Field

### 5.1.1 Step 1. Create ZZZ LANGUAGE Entity

**Figure 16: Exercise 4.1—Creating an Entity for a Pointer Field**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ LANGUAGE
  Are you adding 'ZZZ LANGUAGE' as a new ENTITY (the 280TH)? No// Y <Enter>
(Yes)
```

Repeat the exercises in Lesson 1 to create a new Entity called "**ZZZ LANGUAGE**". Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again to create a new Entity to pull from the LANGUAGE (#.85) file as shown in Figure 17.

**Figure 17: Exercise 4.1—Page 1: Entering Properties for the ZZZ LANGUAGE Entity**

```
                                Edit Entity
NAME: ZZZ LANGUAGE                                                  Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ LANGUAGE
 DISPLAY NAME: Language

 DEFAULT FILE: .85
```

Enter a description, then enter **N** (**Next**) at the Command prompt to continue to Page 2 of the form.

**Figure 18: Exercise 4.1— Entering a Description for the ZZZ LANGUAGE Entity**

```
  1>This Entity expects a pointer to the LANGUAGE (#.85) file, returning a
  2>coded element.
  3><Enter>
EDIT Option:
```

## 5.1.2 Step 2. Add Two Simple Properties

Create the following properties; refer to Exercise 2.3 in Lesson 2 if needed.

**Figure 19: Exercise 4.1—Page 2: Adding Two Simple Properties**

```
                                Edit Entity
NAME: ZZZ LANGUAGE                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type   Field        Sub/File
Code                            1        S    .03          .85
Name                            2        S    .01          .85
```

Use the arrow keys to drop down to the Command prompt, then enter **S** to **Save** your work. Enter **E** to **Exit** the edit form for this Entity.

End of Exercise 4.1.

## 5.2 Exercise 4.2: Convert a Simple Property to an Entity

### 5.2.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 20: Exercise 4.2—Opening the ZZZ NEW PERSON Entity**

```
Select ENTITY: ZZZ NEW PERSON
```

Stay in the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option and select the **ZZZ NEW PERSON** Entity. Use the arrow keys on Page 1 to go right to the Command prompt, then enter **N** (**Next**) to go to Page 2.

### 5.2.2 Step 2. Change Type of Language Property

**Figure 21: Exercise 4.2—Page 2: Changing Type of Language Property**

```
                              Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                          Seq   Type  Field       Sub/File
Gender                        4       S   4           200
IEN                           1       I
Language                      6       S   200.07      200
LastSignOn                    5       S   202         200
Name                          3       S   .01         200
Source                        2       F
State                         7       S   .115        200
```

Use the arrow keys to drop down to the **Language** property, then press **<Enter> twice** to move the cursor to the **Type** column. Enter **E** to change the type from a Simple Field (**S**) to an Entity (**E**); press **<Enter>** to open the dialog for editing a nested Entity item (Figure 22).

### 5.2.3 Step 3. Add ZZZ LANGUAGE Entity to Language Property

**Figure 22: Exercise 4.2—Page 2: Adding ZZZ LANGUAGE Entity to Language Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.          FIELD#: 200.07                   FILE#: 200                        .
.    EXT PTR LKUP:                                                           .
.  INTERNAL VALUE: YES                                                       .
.                                                                            .
.      GET ACTION:                                                           .
.                                                                            .
.OUTPUT TRANSFORM:                                                           .
. INPUT TRANSFORM:                                                           .
.                                                                            .
.          ENTITY: ZZZ LANGUAGE                                              .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
------------------------------------------------------------------------------

Close    Refresh

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: Close                                  Press <PF1>H for help  Insert
```

This dialog (Figure 22) looks almost identical to the one for editing Simple Fields (Figure 14), except for the addition of the **ENTITY** attribute at the bottom. Most nested entities are used with pointer-type fields, to return multiple attributes from a pointed-to file. The field value (i.e., the pointer) is passed to the nested Entity as its input value, and the results of the Entity are now returned as the value of this property instead of the **.01** field or a simple extended pointer reference.

Enter **YES** at the "INTERNAL VALUE:" prompt and **ZZZ LANGUAGE** for the ENTITY.

The **ZZZ LANGUAGE** Entity was created to expect a LANGUAGE (#.85) file IEN as its input value. The **INTERNAL VALUE** attribute tells DDE to request the internal form of the field value from DIQ, instead of the default external form. The internal value, or pointer, will now be passed into **ZZZ LANGUAGE** when DDE calls itself to process this nested Entity.

Use the arrow keys to go to the Command prompt and **Close** the "Simple Field" dialog, returning to Page 2 of the form.

### 5.2.4 Step 4. Save Changes

Use the arrow keys to move the cursor down to the Command prompt, then enter **S** to **Save** your work.

End of Exercise 4.2.

## 5.3 Exercise 4.3: Create a Complex Group Property

Sometimes a data model will expect properties to be collected in a group that are independent fields in the VistA source file, rather than expanding a pointer into multiple fields. This can be accomplished by using a complex group item.

### 5.3.1 Step 1. Create Two Phone Number Properties

Stay on Page 2 of the form in the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option. Use the arrow keys to drop down to the bottom of the Item list and add two new simple properties as shown in Figure 23:

**Figure 23: Exercise 4.3—Page 2: Creating Two Phone Number Properties: HomePhone and OfficePhone**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq   Type  Field       Sub/File
Gender                            4       S   4           200
IEN                               1       I
Language                          6       E   200.07      200
LastSignOn                        5       S   202         200
Name                              3       S   .01         200
Source                            2       F
State                             7       S   .115        200
HomePhone                               S     .131        200
OfficePhone                             S     .132        200
```

Do ***not*** enter a value in the **Seq** column for these two properties. Leaving this field blank causes DDE to skip these two items in the main Item loop when processing an Entity.

### 5.3.2 Step 2. Create Group Property

Add one more property called "**PhoneNumbers**", assigning a **Seq** of **8** and a **Type** of **C**.

**Figure 24: Exercise 4.3—Page 2: Creating a Group Property: PhoneNumbers**

```
HomePhone                               S     .131        200
OfficePhone                             S     .132        200
PhoneNumbers                      8     C
```

Press **<Enter>** to open the "Complex Group" dialog (Figure 25).

**Figure 25: Exercise 4.3—"Complex Group" Dialog: Adding HomePhone and OfficePhone to the group**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                  Page 2 of 3
------------------------------------------------------------------------------

Item                             Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                              .
. GET ACTION:                                                                  .
.                                                                              .
.   Seq   Item                                                                 .
.     1   HomePhone                                                           .
.     2   OfficePhone                                                         .
.                                                                              .
.                                                                              .
.                                                                              .
.                                                                              .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Add the two phone number properties created in Step 1. Enter a **Seq** number inside the group for each property, answering **YES** when asked if adding a new Complex Type, then enter the name of the item when prompted for the Complex Item Name.

Press **<Enter>** to get down to the Command prompt and **Close** this dialog to return to Page 2 of the form.

### 5.3.3 Step 3. Save Changes and Exit

Use the arrow keys to drop down to the Command line, **Save** your changes and **Exit** the form.

End of Exercise 4.3.

## 5.4 Exercise 4.4: Test Your Changes

### 5.4.1 Step 1. Exit VA FileMan to Programmer Mode

Exit VA FileMan to programmer mode.

### 5.4.2 Step 2. Test Entity Using $$GET1^DDE

Run $$GET1^DDE again as before, noting the change to the **Language** property and the new **PhoneNumbers** property.

**Figure 26: Exercise 4.4—Test Entity: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":"VA", "Name":"PROGRAMMER,ZZZ ONE", "Gender":"
FEMALE", "LastSignOn":"SEP 08, 2022@11:18:19", "Language":{"Code":"ENG", "Name":
"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePho
ne":"555-123-4567"}}
>
```

End of Exercise 4.4.

# 6 Lesson 5: Add List Properties

Unlike an Entity, which can expand a single VistA value into multiple properties, a **List** property can return multiple instances of a field or record. In this lesson you will learn how to create and add a **List** property.

## 6.1 Exercise 5.1: Create a Subfile List Property

### 6.1.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 27: Exercise 5.1—Creating a Subfile List Property**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity. Use the arrow keys on Page 1 to go right to the Command prompt, then enter **N** (**Next**) to go to Page 2.

### 6.1.2 Step 2. Create Divisions Property

Use the arrow keys to drop down to the bottom of the Item list. Enter a new property name of "**Divisions**" in the **Item** column and respond **YES** when asked if adding a new item.

**Figure 28: Exercise 5.1—Page 2: Creating Divisions Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
Divisions                       9       L
```

Again, press **<Enter>** to jump to **Seq** and enter the number **9**. Press **<Enter>** to jump to the **Type** and enter the letter **L** (**List**); pressing **<Enter>** one more time will open the "List" dialog (Figure 29).

### 6.1.3 Step 3. Define Source of List

For this exercise, you will be adding the list of divisions that this person can sign in to from the DIVISION (#16) field, which is a Multiple; so, select **SUBFILE** for the **LIST TYPE**.

**Figure 29: Exercise 5.1—Page 2 "List" Dialog: Defining Source of List**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                           Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: SUBFILE                                                         .
.                                                                            .
.GET ACTION:                                                                 .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY:                                    FIELD#:                      .
.                                              EXT PTR:                      .
.                                              INT VAL:                      .
.                                                                            .
.   XML TAG:                                                                 .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

This opens another dialog where the sub-file source and any filters can be defined (Figure 30).

**Figure 30: Exercise 5.1—Defining Subfile Source**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                           Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T .
..Define the sub/file search for the desired records:                      . .
..                                                                         . .
..    FILE#: 200.02                                                        . .
..     XREF:                                                               . .
..FILTER BY:                                                               . .
..                                                                         . .
..   SCREEN:                                                               . .
..                                                                         . .
.F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Enter the subfile number at the "FILE#:" prompt, which is **200.02** for the DIVISION (#16) field. The other fields on this screen will be addressed in a later lesson. Press **<Enter>** to get to the Command prompt at the bottom and select **Close** to return to the "List Property" dialog.

### 6.1.4 Step 4. Complete Attributes for this List

**Figure 31: Exercise 5.1—Page 2 "List Property": Completing Attributes for this List**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                          Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: SUBFILE                                                         .
.                                                                            .
.GET ACTION:                                                                 .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY:                                    FIELD#: .01                  .
.                                              EXT PTR: 99                   .
.                                              INT VAL:                      .
.                                                                            .
.   XML TAG: Division                                                        .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
______________________________________________________________________________

Close    Refresh

Enter a COMMAND, or "^" followed by the CAPTION of a FIELD to jump to.

COMMAND: Close                                  Press <PF1>H for help  Insert
```

A subfile list supports the ability to return multiple instances of a single field from within that subfile. Press **<Enter>** to get to the "FIELD#:" prompt and enter **.01** to add a list of divisions, then press **<Enter>** again and enter **99** to return the Station Number from the pointed-to INSTITUTION (#4) file instead of its name.

XML places a set of tags around each instance, which can be defined using the **XML TAG** attribute; DDE will use the property name if left empty, but it is common for the list name to be plural and each instance to use its singular form. For this exercise, enter "**Division**" at the "XML TAG:" prompt.

### 6.1.5 Step 5. Save Changes and Exit

Select **Close** at the Command prompt to return to the edit form. Press **<Enter>** or use the arrow keys to get to the Command prompt again, **Save** your changes, and **Exit** the form.

End of Exercise 5.1.

## 6.2 Exercise 5.2: Test Your Changes

### 6.2.1 Step 1. Exit VA FileMan to Programmer Mode

Exit VA FileMan to programmer mode.

### 6.2.2 Step 2. Test Entity Using $$GET1^DDE

Run $$GET1^DDE again as before, noting the addition of the Divisions list inside the square **[ ]** brackets (Figure 32):

**Figure 32: Exercise 5.2—Testing Entity: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":"VA", "Name":"PROGRAMMER,ZZZ ONE", "Gender":"
FEMALE", "LastSignOn":"SEP 08, 2022@11:18:19", "Language":{"Code":"ENG", "Name":
"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePho
ne":"555-123-4567"}, "Divisions":[500, "500A4"]}
>
```

### 6.2.3 Step 3. Run Test as XML

Run $$GET1^DDE again but include a **1** in the **fourth** parameter to return the results as XML, noting the "Division" tags around each instance (Figure 33):

**Figure 33: Exercise 5.2—Testing Entity: Running $$GET1^DDE: XML Results**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ,,1)
<NewPerson><IEN>11948</IEN><Source>VA</Source><Name>PROGRAMMER,ZZZ ONE</Name><Ge
nder>FEMALE</Gender><LastSignOn>SEP 08, 2022@11:18:19</LastSignOn><Language><Cod
e>ENG</Code><Name>ENGLISH</Name></Language><State>UT</State><PhoneNumbers><HomeP
hone>555-867-5309</HomePhone><OfficePhone>555-123-4567</OfficePhone></PhoneNumbe
rs><Divisions><Division>500</Division><Division>500A4</Division></Divisions></Ne
wPerson>
>
```

End of Exercise 5.2.

## 6.3 Exercise 5.3: Convert Subfile List to an Entity [Optional]

As an optional exercise, you can elect to convert the Divisions list to use an Entity; this can be done in either of two ways:

- Create an Entity for the INSTITUTION (#4) file and use this as the Entity with Field #.01 instead of the Extended Pointer reference in the Subfile list (see Exercise 4.1 and Exercise 4.2).
- Create an Entity to return each instance or record in the subfile instead of the single **#.01** field.

This exercise walks you through the latter, creating an Entity to return a subfile record instead of a single field. Refer to the earlier lessons in this tutorial as needed.

### 6.3.1 Step 1. Create Entity for Division Subfile

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option to create a new Entity called "**ZZZ DIVISION**" to pull from the DIVISION (#200.02) subfile. Enter "**Division**" for the **DISPLAY NAME** and the subfile number **200.02** as the **DEFAULT FILE**.

**Figure 34: Exercise 5.3—Page 1: Creating Entity for Division Subfile**

```
                                Edit Entity
NAME: ZZZ DIVISION                                                Page 1 of 3
------------------------------------------------------------------------------

         NAME: ZZZ DIVISION
 DISPLAY NAME: Division

 DEFAULT FILE: 200.02
```

Include a description, then enter **N** (**Next**) at the Command prompt to continue to Page 2 of the form. Create the following two simple field properties (Figure 35):

**Figure 35: Exercise 5.3—Page 2: Creating Two Simple Field Properties: Name and Default**

```
                                Edit Entity
NAME: ZZZ DIVISION                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type   Field        Sub/File
Name                            1        S    .01          200.02
Default                         2        S    1            200.02
```

**Save** your changes and **Exit** the form.

### 6.3.2 Step 2. Modify Divisions Property in ZZZ NEW PERSON Entity

Stay in the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option and select the **ZZZ NEW PERSON** Entity. Use the arrow keys on Page 1 to go right to the Command prompt, then enter **N** (**Next**) to go to Page 2.

**Figure 36: Exercise 5.3—Page 2: Modifying Divisions Property in ZZZ NEW PERSON Entity**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L   .01         200.02
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
```

The cursor should be on the **Divisions** property at the top, so press **<Enter> three** times to open the "List" dialog (Figure 37).

**Figure 37: Exercise 5.3—Page 2 "List" Dialog: Selecting and Editing the ZZZ DIVISION Entity**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: SUBFILE                                                         .
.                                                                            .
.GET ACTION:                                                                 .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY: ZZZ DIVISION                        FIELD#: @                   .
.                                              EXT PTR:                      .
.                                              INT VAL:                      .
.                                                                            .
.   XML TAG: Division                                                        .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Use the arrow keys to move down to the "ENTITY:" prompt and enter the name of the new Entity (**ZZZ DIVISION**). Press **<Enter>** to move to the "FIELD#:" prompt and enter an **at-sign** (**@**) to remove the value. **Close** this dialog and **Save** your changes before you **Exit** the form.

### 6.3.3 Step 3. Test Changes

Exit VA FileMan to programmer mode and use $$GET1^DDE to test your changes, noting the **Division** changes (Figure 38).

**Figure 38: Exercise 5.3—Testing Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":"VA", "Name":"PROGRAMMER,ZZZ ONE", "Gender":"
FEMALE", "LastSignOn":"SEP 08, 2022@11:18:19", "Language":{"Code":"ENG", "Name":
"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePho
ne":"555-123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER", "Default":"Yes"}, {"
Name":"ALBANY OPC"}]}
>
```

### 6.3.4 Step 4. Test Again as XML

Run $$GET1^DDE again with a **1** in the **fourth** parameter to return the results as XML (Figure 39).

**Figure 39: Exercise 5.3—Testing Changes: Running $$GET1^DDE: XML Results**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ,,1)
<NewPerson><IEN>11948</IEN><Source>VA</Source><Name>PROGRAMMER,ZZZ
ONE</Name><Gender>FEMALE</Gender><LastSignOn>SEP 08,
2022@11:18:19</LastSignOn><Language><Code>ENG</Code><Name>ENGLISH</Name></Langua
ge><State>UT</State> <PhoneNumbers><HomePhone>555-867-
5309</HomePhone><OfficePhone>555-123-
4567</OfficePhone></PhoneNumbers><Divisions><Division><Name>MEDICAL
CENTER</Name><Default>Yes</Default></Division><Division><Name>ALBANY
OPC</Name></Division></Divisions></NewPerson>
>
```

End of Exercise 5.3.

# 7 Lesson 6: Using Action Fields

In this lesson, you will learn how to customize your Entity by using the M code fields in the ENTITY (#1.5) file.

## 7.1 Exercise 6.1: GET ACTION

Every item type supports the **GET ACTION** attribute to customize its behavior. It can also be used to set the value of a property instead of calling DIQ with the field number.

### 7.1.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 40: Exercise 6.1—Opening ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity. Use the arrow keys on Page 1 to go right to the Command prompt, then enter **N** (**Next**) to go to Page 2.

### 7.1.2 Step 2. Open Source Property

**Figure 41: Exercise 6.1—Page 2: Opening Source Property**

```
                              Edit Entity
NAME: ZZZ NEW PERSON                                                  Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L               200.02
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
```

Use the arrow keys to drop down to the **Source** item, press **<Enter> three** times to open the "Fixed String" dialog (Figure 42).

### 7.1.3 Step 3. Modify Source Property

**Figure 42: Exercise 6.1—"Fixed String" Dialog: Modifying Source Property**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                  Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                              .
.FIXED RESPONSE: VA                                                            .
.                                                                              .
.                                                                              .
.    GET ACTION: S VALUE=$G(DUZ(2))                                            .
.                                                                              .
.                                                                              .
.                                                                              .
.                                                                              .
.                                                                              .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Add this line of code at the "GET ACTION:" prompt as shown in Figure 42, to return the user's current division rather than the static string "**VA**". This allows the property to return a dynamic value that is *not* dependent on a field in the file.

DDE attempts to create the local **VALUE** variable for every item containing its results. **GET ACTION** is always executed first for any Item type; if it defines **VALUE**, the usual means of obtaining a result for the Item, such as calling DIQ, is *not* performed.

### 7.1.4 Step 4. Close Item

Press **<Enter>** to get to the Command line and **Close** this dialog, returning to Page 2 of the form.

End of Exercise 6.1.

## 7.2 Exercise 6.2: OUTPUT TRANSFORM

If a data model expects the property value to be returned in a different format than stored in VistA, the **OUTPUT TRANSFORM** can re-format the results as needed.

### 7.2.1 Step 1. Open LastSignOn Property

**Figure 43: Exercise 6.2—Page2 : Opening LastSignOn Property**

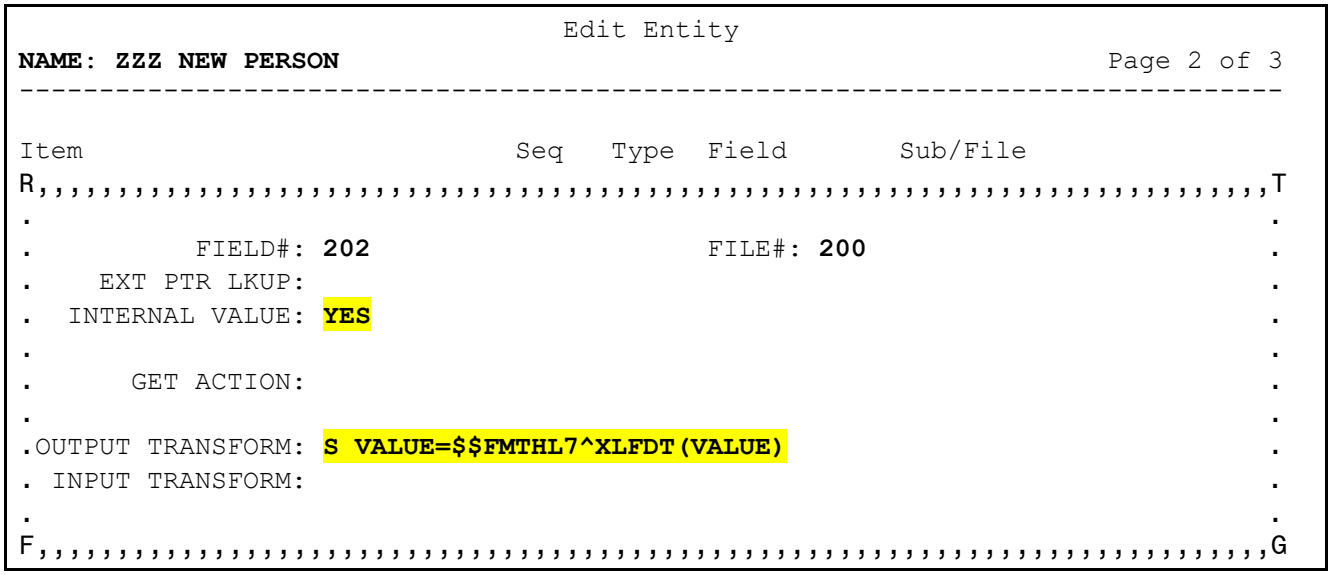
```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L               200.02
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
```

Still on Page 2 of the form, use the arrow keys to move to the **LastSignOn** property in the Item list. Press **<Enter> three** times to open the "Simple Field" dialog (Figure 44).

### 7.2.2 Step 2. Modify LastSignOn Property

**Figure 44: Exercise 6.2—"Simple Field" Dialog: Modifying LastSignOn Property**




Add this line of code at the "OUTPUT TRANSFORM:" prompt as shown and set **INTERNAL VALUE** to "**YES**".

The Output Transform is executed if the **VALUE** variable is defined after executing the **GET ACTION** code or calling DIQ. (If **VALUE** is **NULL** or undefined, the Output Transform is *not* executed.) This is M code that expects the field contents to be in **VALUE**; it should perform any needed modifications to **VALUE**, especially formatting changes, and leave the results in **VALUE** when finished.

**NOTE:** You may now need to retrieve the internal form of the field contents, depending on its VistA data type and how it needs to be modified. The **INTERNAL VALUE** attribute tells DDE to request the internal form of the field value from DIQ.

### 7.2.3 Step 3. Close Item

Press **<Enter>** to get to the Command line and **Close** this dialog, returning to Page 2 of the form.

End of Exercise 6.2.

## 7.3 Exercise 6.3: GET ID ACTION

Sometimes a record may need to be validated, to ensure that it is appropriate to include in the results. **GET ID ACTION** is M code that is executed for each record, before any field values have been retrieved.

### 7.3.1 Step 1. Move to Page 3 of Form

Still on Page 2 of the form, use the arrow keys to move to the Command prompt at the bottom; enter **N** (**Next**) to move to the next page.

### 7.3.2 Step 2. Add Code to GET ID ACTION Field

Press **<Enter>** twice to go to the **GET ID ACTION** field, then enter the following line of code (Figure 45):

**Figure 45: Exercise 6.3—Adding Code to the GET ID ACTION Field**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 3 of 3
------------------------------------------------------------------------------

GET ENTRY ACTION:
 GET EXIT ACTION:
   GET ID ACTION: I '$$ACTIVE^XUSER(DIEN) S DDEOUT=1
```

The **GET ID ACTION** is executed at the start of processing a record. The local **DIEN** variable contains the record identifier that was passed into DDE and can be referenced throughout the Entity.

For this exercise, you will only return the record if the user is active. If it is determined that the user should *not* be included in the results, then the **DDEOUT** variable can be set to **1**. This causes DDE to stop processing this record and *not* return any results for it.

### 7.3.3 Step 3. Save Changes

Press **<Enter>** to drop down to the Command line and enter **S** to **Save** your changes.

End of Exercise 6.3.

## 7.4 Exercise 6.4: GET ENTRY/EXIT ACTIONS

The **GET ENTRY ACTION** field can perform any needed set up for the Entity, such as retrieving package parameters; it should *not* be used for setting up an individual record. The **GET ID ACTION**, however, can be used to retrieve data specific to a record, as well as for validation as in the previous exercise.

**GET EXIT ACTION** is executed after all records for an Entity have been processed and should be used to clean up any local variables created by any "Action" field throughout the Entity.

### 7.4.1 Step 1. Modify GET ID ACTION Field

Stay on Page 3 of the form and use the arrow keys to move back to the **GET ID ACTION** field.

**Figure 46: Exercise 6.4—Page 3: Modifying the GET ID ACTION Field**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 3 of 3
------------------------------------------------------------------------------

GET ENTRY ACTION:
 GET EXIT ACTION:
   GET ID ACTION: S ZZACTIVE=$$ACTIVE^XUSER(DIEN) I 'ZZACTIVE S DDEOUT=1
```

Change the code to save the results of the $$ACTIVE function in local **ZZACTIVE** variable and change the condition for setting **DDEOUT** to use **ZZACTIVE** now.

### 7.4.2 Step 2. Kill ZZACTIVE in GET EXIT ACTION Field

Use the arrow keys to move up to the **GET EXIT ACTION** field.

**Figure 47: Exercise 6.4—KILLing ZZACTIVE in GET EXIT ACTION Field**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 3 of 3
------------------------------------------------------------------------------

GET ENTRY ACTION:
 GET EXIT ACTION: K ZZACTIVE
   GET ID ACTION: S ZZACTIVE=$$ACTIVE^XUSER(DIEN) I 'ZZACTIVE S DDEOUT=1
```

Add code to **KILL** the **ZZACTIVE** variable.

Press **<Enter>** to get to the Command prompt and enter **S** to **Save** your changes.

### 7.4.3 Step 3. Modify LastSignOn Property to Use ZZACTIVE

At the Command prompt enter **P** (**Previous**) to move to the previous page (Page 2).

**Figure 48: Exercise 6.4—Modifying LastSignOn Property to Use ZZACTIVE**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L               200.02
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
```

Use the arrow keys to move to the **LastSignOn** property, then press **<Enter> three** times to open the "Item" dialog (Figure 49).

**Figure 49: Exercise 6.4—Page 2 "Item" Dialog: Adding Code to the GET ACTION Field**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                             .
.          FIELD#: 202                      FILE#: 200                        .
.    EXT PTR LKUP:                                                            .
.  INTERNAL VALUE: YES                                                        .
.                                                                             .
.      GET ACTION: S VALUE=$P(ZZACTIVE,"^",3)                                 .
.                                                                             .
.OUTPUT TRANSFORM: S VALUE=$$FMTHL7^XLFDT(VALUE)                              .
. INPUT TRANSFORM:                                                            .
.                                                                             .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Use the arrow keys to move to the **GET ACTION** field and add the highlighted code shown in Figure 49.

If the call used in the **GET ID ACTION** returns data, it can be saved and referenced inside the appropriate item rather than retrieving it from the file again. The $$ACTIVE^XUSER application programming interface (API) returns the last signon date for active users in the third piece of the result string, in internal VA FileMan format. You can get the value from here and save a call to DIQ.

It is *recommended* to retain the field number and any needed DIQ parameters in the item definition, even when using the **GET ACTION** field to set the value. If there is no value in this piece, then DDE attempts to retrieve the field value via DIQ as usual. It also documents the source of the item's value.

This approach can be especially useful for getting data from another VistA package. Many Integration Control Registrations (ICRs) grant permission to access data via an API, which can be called in the **GET ID ACTION**. These results can be saved into a local variable or array that is now available to be referenced throughout the Entity. Remember to **KILL** any local variables you create in the **GET EXIT ACTION** of the Entity.

### 7.4.4 Step 4. Save Changes and Exit

Press **<Enter>** to get to the Command prompt and **Close** the dialog. Use the arrow keys to drop to the Command prompt again, **Save** your changes, and **Exit** the form.

End of Exercise 6.4.

## 7.5 Exercise 6.5: Test Your Changes

### 7.5.1 Step 1. Exit VA FileMan to Programmer Mode

Exit VA FileMan to programmer mode.

### 7.5.2 Step 2. Test Entity Using $$GET1^DDE

Run $$GET1^DDE again as before, noting the changes to the **Source** and **LastSignOn** properties:

**Figure 50: Exercise 6.5—Testing Entity: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":"PROGRAMMER,ZZZ ONE",
"Gender":"FEMALE", "LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG",
"Name":"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309",
"OfficePhone":"555-123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER",
"Default":"Yes"}, {"Name":"ALBANY OPC"}]}
>
```

End of Exercise 6.5.

# 8 Lesson 7: Queries

An Entity can be used to return a list of records, rather than just one at a time. Parameters can be defined to support a DIC call on the default file, or you can create a special lookup routine. In this lesson, you will learn how to add a query to an Entity and test it.

## 8.1 Exercise 7.1: DIC Parameters

The ENTITY (#1.5) file supports the input parameters needed to make a simple DIC call. Specify the search index to use on the default file, and optionally, a screen or value to search for, and DDE returns the matching records according to the Entity definition.

### 8.1.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 51: Exercise 7.1—Opening ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity.

### 8.1.2 Step 2. Add Search Parameters

For this exercise, you will set up the Entity to look for users who have names that begin with "**ZZZ**". Use the arrow keys to go to the **SORT BY** and **FILTER BY** fields, enter the values as shown in Figure 52.

**Figure 52: Exercise 7.1—Page 1: Adding Search Parameters**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ NEW PERSON
 DISPLAY NAME: NewPerson

 DEFAULT FILE: 200

      SORT BY: B                              DATA MODEL:
    FILTER BY: ZZZ                             READ ONLY:
       SCREEN:

QUERY ROUTINE:
```

The **SORT BY** field holds the name of the cross-reference that DIC should use for the search. **FILTER BY** can be set to the value to look for in that index; this value is optional, and all entries in the file or index are returned if left blank. DIC returns partial matches, so using the string "**ZZZ**" finds all names in the "**B**" index that begin with those characters. A **Screen** is optional M code that can be applied to each record for additional filtering.

**REF:** For details on creating a screen, see any of the DIC calls in the *VA FileMan Developers Guide*.

Recall that in Exercise 5.1 the definition for a subfile list included the **XRef**, **Filter By**, and **Screen** attributes. Those are the same as the **Sort By**, **Filter By**, and **Screen** attributes shown in Figure 52, respectively. They allow an index, search value, or screen to be defined for finding records to populate a list property.

### 8.1.3 Step 3. Save Changes and Exit

Use the arrow keys to drop down to the Command prompt, **Save** your changes, and **Exit** the form.

End of Exercise 7.1.

## 8.2 Exercise 7.2: GET^DDE

To test the query function of an Entity, you *must* use a different call since $$GET1^DDE is expecting an identifier for a single record. GET^DDE will attempt to execute our query and return a list of matches, formatted according to our Entity definition.

### 8.2.1 Step 1. Exit VA FileMan to Programmer Mode

Exit VA FileMan to programmer mode.

### 8.2.2 Step 2. Test Entity Using GET^DDE and New Search Parameters

Run GET^DDE this time so that DIC will look for matching records. To use XML format for this test, pass a **1** in the **fourth** parameter as with $$GET1.

**REF:** For details on how to use the DDE calls, see the *VA FileMan Developers Guide*.

**Figure 53: Exercise 7.2—Testing Entity: Running GET^DDE Using New Search Parameters**

```
>D GET^DDE("ZZZ NEW PERSON",,,1)

>D ^%G

For help on global specifications DO HELP^%G
Global ^TMP("DDE GET",$J
^TMP("DDE GET",3282,0)=2
                      1)="<NewPerson><IEN>1008</IEN><Source>500</Source><Name>ZZZP
RIBBLE,MAMIE</Name><Gender>FEMALE</Gender><LastSignOn>20220817134256-0500</LastS
ignOn></NewPerson>"
                      2)="<NewPerson><IEN>1011</IEN><Source>500</Source><Name>ZZZS
ICARD,JEROME</Name><LastSignOn>20220731093412-0500</LastSignOn></NewPerson>"

Global ^
```

**At this point, the user presses the <Enter> key to exit.**

```
>
```

Results are returned in **^TMP(“DDE GET”,$J)**, so you can use the global lister to view the array. The total number of records found is returned in the **zero** node of the array.

### 8.2.3 Step 3. D GET^DDE Again with Different Filter String

GET^DDE can accept another filter value as an input parameter that overrides the default stored in the Entity. Run this test again, requesting a different last name, such as "**SMITH**" in the **third** parameter.

**Figure 54: Exercise 7.2—Testing Entity: Running GET^DDE Using Different Search Parameters**

```
>D GET^DDE("ZZZ NEW PERSON",,"SMITH",1),^%G

For help on global specifications DO HELP^%G
Global ^TMP("DDE GET",$J
^TMP("DDE GET",3282,0)=7
                     1)="<NewPerson><IEN>11895</IEN><Source>500</Source><Name>SMI
TH,DEAN W</Name><Gender>MALE</Gender><LastSignOn>20090720</LastSignOn></NewPerso
n>"
                     2)="<NewPerson><IEN>11275</IEN><Source>500</Source><Name>SMI
TH,JOHN</Name><State>NY</State><PhoneNumbers><HomePhone>555-555-1212</HomePhone>
</PhoneNumbers></NewPerson>"
                     3)="<NewPerson><IEN>11653</IEN><Source>500</Source><Name>SMI
TH,JOHN A</Name></NewPerson>"
                     4)="<NewPerson><IEN>11599</IEN><Source>500</Source><Name>SMI
TH,LUTHER</Name><State>DC</State></NewPerson>"
                     5)="<NewPerson><IEN>11442</IEN><Source>500</Source><Name>SMI
TH,MICHAEL C</Name></NewPerson>"
                     6)="<NewPerson><IEN>1611</IEN><Source>500</Source><Name>SMIT
H,ROBERT</Name><Gender>MALE</Gender></NewPerson>"
                     7)="<NewPerson><IEN>11828</IEN><Source>500</Source><Name>SMI
TH,TONY</Name><Gender>MALE</Gender></NewPerson>"

Global ^
```

**At this point, the user presses the <Enter> key to exit.**

```
>
```

End of Exercise 7.2.

## 8.3 Exercise 7.3: Custom Query Routine

DDE also supports the use of a custom lookup or query routine, if DIC is *not* sufficient. The only requirement is that this code returns the **DLIST** array.

### 8.3.1 Step 1. Create Search Routine

For this exercise, rather than filtering by name you will look for users that hold the Provider key. Use your preferred editor to create a new routine that loops on the **AK.PROVIDER** cross-reference of the NEW PERSON (#200) file. Figure 55 is a possible sample:

**Figure 55: Exercise 7.3—Creating a Search Routine**

```
ZZZDEMO  ;OIT/STAFF - DDE DEMO
         ;;DEMO
FIND     ;demo query routine
         N NUM,NAME,IEN
         S NUM=0
         S NAME="" F  S NAME=$O(^VA(200,"AK.PROVIDER",NAME)) Q:NAME=""  D
         . S IEN=0 F  S IEN=$O(^VA(200,"AK.PROVIDER",NAME,IEN)) Q:'IEN  D
         .. S NUM=NUM+1,DLIST(NUM)=IEN
         Q
```

This code *must* return the **DLIST(#)** array, where the # is simply a sequence number and each node is set to a record identifier in the format expected by the Entity. **ZZZ NEW PERSON** expects a pointer to the NEW PERSON (#200) file.

### 8.3.2 Step 2. Test Routine in Programmer Mode

**Figure 56: Exercise 7.3—Testing Routine in Programmer Mode**

```
>D FIND^ZZZDEMO ZW DLIST
DLIST(1)=11295
DLIST(2)=11960
DLIST(3)=11962
…

>
```

### 8.3.3 Step 3. Open ZZZ NEW PERSON Entity

**Figure 57: Exercise 7.3—Opening ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity.

### 8.3.4 Step 4. Add Query Routine

**Figure 58: Exercise 7.3—Adding Query Routine**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                  Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ NEW PERSON
  DISPLAY NAME: NewPerson

  DEFAULT FILE: 200

       SORT BY: B                                 DATA MODEL:
     FILTER BY: ZZZ                               READ ONLY:
        SCREEN:

QUERY ROUTINE: FIND^ZZZDEMO

   DESCRIPTION: +
```

Use the arrow keys to get to the **QUERY ROUTINE** field, and then enter the tag and name of your search routine. The FIND^DIC parameters can be left defined, as the **QUERY ROUTINE** will take precedence.

Use the arrow keys to drop down to the Command prompt, **Save** your changes, and **Exit** the form.

### 8.3.5 Step 5. Test Changes

Exit to programmer mode and run GET^DDE again to view the results.

**Figure 59: Exercise 7.3—Testing Changes**

```
>D GET^DDE("ZZZ NEW PERSON"),^%G

Global ^TMP("DDE GET",$J
^TMP("DDE GET",3282,0)=170
                    1)="{""IEN"":11295, ""Source"":500,
""Name"":""ABNERATHY,GEORGE"", ""Gender"":""MALE""}"
                    2)="{""IEN"":11960, ""Source"":500,
""Name"":""ACHARYA,SHIVA"", ""Gender"":""FEMALE"", ""LastSignOn"":20190916}"
…
```

**NOTE:** This exercise could also have been accomplished by changing the **Sort By** value to "**AK.PROVIDER**" and removing the value for **Filter By**.

End of Exercise 7.3.

## 8.4 Exercise 7.4: General List Property

Sometimes, a list property is needed that *cannot* easily be built using the methods already discussed in this tutorial. A generic list type is available that can be constructed like a query.

### 8.4.1 Step 1. Create Search Routine

For this exercise, you will add a property to list the user's security keys. Use your preferred editor to add another bit of code to your **QUERY ROUTINE** that will loop on the **^XUSEC** global. Figure 60 is a possible sample:

**Figure 60: Exercise 7.4—Creating a Search Routine**

```
KEYS(USER) ;find user's keys
          N NUM,KEY
          S NUM=0,USER=+$G(USER)
          S KEY="" F  S KEY=$O(^XUSEC(KEY)) Q:KEY=""  I $D(^XUSEC(KEY,USER)) S
NUM
=NUM+1,DLIST(NUM)=KEY
          Q
```

Like the query, this code *must* return the **DLIST(#)** array where the # is simply a sequence number. However, here each node should be set to the value, in this case the key name, that is added to the results.

### 8.4.2 Step 2. Test Routine in Programmer Mode

**Figure 61: Exercise 7.4—Testing the Routine in Programmer Mode**

```
>D KEYS^ZZZDEMO(DUZ) ZW DLIST
DLIST(1)="DGPF ASSIGNMENT"
DLIST(2)="ORES"
DLIST(3)="PROVIDER"
DLIST(4)="XUPROG"
DLIST(5)="XUPROGMODE"

>
```

### 8.4.3 Step 3. Open ZZZ NEW PERSON Entity

**Figure 62: Exercise 7.4—Opening the ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity. Use the arrow keys on Page 1 to go right to the Command prompt, then enter **N** (**Next**) to go to Page 2.

### 8.4.4 Step 4. Add Keys Property

Use the arrow keys to drop down to the bottom of the item list. Enter a new property name of **Keys** in the **Item** column and respond **YES** when asked if adding a new Item.

**Figure 63: Exercise 7.4—Page 2: Adding Keys Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq   Type  Field       Sub/File
Gender                            4       S   4           200
HomePhone                                 S   .131        200
IEN                               1       I
Language                          6       E   200.07      200
LastSignOn                        5       S   202         200
Name                              3       S   .01         200
OfficePhone                               S   .132        200
PhoneNumbers                      8       C
Source                            2       F
State                             7       S   .115        200
Keys                              10      L
```

Again, press **<Enter>** to jump to **Seq** and enter the number **10**. Press **<Enter>** to jump to the **Type** and enter the letter **L**; pressing **<Enter>** one more time opens the "List" dialog (Figure 64).

### 8.4.5 Step 5. Define List

For this exercise, you will be building a list from scratch, so select **ARRAY** for the **LIST TYPE** field. This type is built entirely from code in the **GET ACTION** field, so there is no list type popup dialog to define the source.

**Figure 64: Exercise 7.4—Page 2 "List" Dialog: Defining List**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: ARRAY                                                           .
.                                                                            .
.GET ACTION: D KEYS^ZZZDEMO(DIEN)                                            .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY:                                    FIELD#:                      .
.                                              EXT PTR:                      .
.                                              INT VAL:                      .
.                                                                            .
.   XML TAG: Key                                                             .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Like a Query, the **GET ACTION** field *must* create the **DLIST(#)** array. Each node should be set to a single value that is returned and added to the results; alternatively, the list can return identifiers for passing into a specified Entity. Remember that the local **DIEN** variable holds the record ID and can be referenced. (For this demo Entity, **DIEN** is the **#200** pointer to a user.)

The field attributes on the right-hand side of the screen are *not* used with this type of list. Enter the singular name "**Key**" for the **XML TAG** field to identify each instance, when using XML format.

Press **<Enter>** to get to the Command prompt, **Close** the dialog, and return to Page 2 of the form.

### 8.4.6 Step 6. Save Changes and Exit Form

### 8.4.7 Step 7. Test Changes

Exit to programmer mode and run $$GET1^DDE again to view the results.

**Figure 65: Exercise 7.4—Test Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":"PROGRAMMER,ZZZ ONE",
"Gender":"FEMALE", "LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG",
"Name":"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309",
"OfficePhone":"555-123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER",
"Default":"Yes"}, {"Name":"ALBANY OPC"}], "Keys":["DGPF ASSIGNMENT", "ORES",
"PROVIDER", "XUPROG", "XUPROGMODE"]}
>
```

# 9 Lesson 8: Advanced Features [Optional]

Sometimes, a little creativity is needed to translate VistA into the desired data model. In this optional lesson, you will learn about other features available in the DDE utility for handling special situations.

## 9.1 Exercise 8.1: Set of Codes as an Entity

A nested Entity is not only used to expand a pointer into Multiple fields from the target file. Both internal and external forms of a value are often needed for SET OF CODE fields as well. An Entity can be created to return both forms, but the variables holding the source file and field numbers are **NEW**ed when entities are nested and called recursively. DDE provides the **DATA** variable to save any information from the property that needs to be available inside the nested Entity.

### 9.1.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 66: Exercise 8.1—Opening ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity. Use the arrow keys to get to the Command prompt and enter **N** (**Next**) to go to the next page.

### 9.1.2 Step 2. Change Gender Property to an Entity

Use the arrow keys to drop down to the **Gender** property.

**Figure 67: Exercise 8.1—Page 2: Changing Gender Property to an Entity**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L               200.02
Gender                          4       E   4           200
HomePhone                               S   .131        200
IEN                             1       I
Keys                            10      L
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
_______________________________________________________________________________
```

Press **<Enter> twice** to move to the **Type** column. Change the type to **E** and press **<Enter>**, which opens the "Entity Item" dialog (Figure 68).

### 9.1.3 Step 3. Update Item

**Figure 68: Exercise 8.1—Page 2 "Entity Item" Dialog: Updating Item**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.          FIELD#: 4                          FILE#: 200                      .
.    EXT PTR LKUP:                                                           .
.  INTERNAL VALUE: YES                                                       .
.                                                                            .
.      GET ACTION:                                                           .
.                                                                            .
.OUTPUT TRANSFORM:                                                           .
. INPUT TRANSFORM:                                                           .
.                                                                            .
.          ENTITY: ZZZ CODED ELEMENT                                         .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
------------------------------------------------------------------------------
  Located in the Z (Local) namespace.
  Are you adding 'ZZZ CODED ELEMENT' as a new ENTITY (the 284TH)? No// Y
```

Set the **INTERNAL VALUE** flag to **YES**, to return **VALUE** as the code instead of the name.

Notice that the "ENTITY:" prompt allows adding a new entry (Learn As You Go [LAYGO]). This only creates a new entry in the ENTITY (#1.5) file; the new Entity *must* be edited as per usual to add any needed attributes and properties.

**Close** this dialog and return to the form. **Save** your changes and **Exit** the form.

### 9.1.4 Step 4. Flesh Out New ZZZ CODED ELEMENT Entity

At the "Select ENTITY:" prompt, enter **ZZZ CODED ELEMENT**.

**Figure 69: Exercise 8.1—Page 1: Fleshing Out New ZZZ CODED ELEMENT Entity**

```
                                  Edit Entity
NAME: ZZZ CODED ELEMENT                                             Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ CODED ELEMENT
 DISPLAY NAME: CodeSet

 DEFAULT FILE:
```

Only the **DISPLAY NAME** field *must* be completed here, leave the **DEFAULT FILE** field and query fields blank, so this Entity can be used for any SET OF CODES field in any file. The property name is used for tags when this Entity is nested inside another, so this text just provides a default value as well as a brief description of its purpose.

It is *highly recommended* to save a description for utility entities that have no defined source file, such as shown in Figure 70:

**Figure 70: Exercise 8.1—Adding a Description for the Utility Entities**

```
  1>This Entity returns a generic coded element for a Code, and its name in the
  2>variable DATA. It is intended for use as a nested Entity.
  3><Enter>
EDIT Option:
```

Press **<Enter>** to get to the Command line and enter **P** to go to Page 3 next this time.

**Figure 71: Exercise 8.1—Page 3: Entering Code in the GET ENTRY ACTION Field**

```
                                Edit Entity
NAME: ZZZ CODED ELEMENT                                          Page 3 of 3
------------------------------------------------------------------------------

GET ENTRY ACTION: S DATA=$$EXTERNAL^DILFD(FILE,FIELD,,VALUE)
 GET EXIT ACTION:
   GET ID ACTION:
```

The **DATA** variable can be set to the external form of **VALUE** to make it available inside the nested Entity. The **FILE** and **FIELD** variables hold the current file and field number respectively and can be referenced in the property's "Action" fields. **DATA** is defined for the scope of the property, which includes the nested Entity, and is automatically cleaned up when the property in the main Entity has completed.

When an Entity is nested its **GET ENTRY ACTION** field is executed by the calling property once the input **VALUE** has been determined so **FILE** and **FIELD** still reflect the **Gender** property definition. **DILFD** *must* be called here as these variables will be **NEW**ed when the nested Entity is processed. The results of the nested Entity is returned as the property's **VALUE**. Finally, the nested Entity's **GET EXIT ACTION** is executed.

Press **<Enter>** to get to the Command line and enter **P** (**Previous**) again to move to Page 2.

**Figure 72: Exercise 8.1—Page 2: Creating Two New Properties**

```
                                Edit Entity
NAME: ZZZ CODED ELEMENT                                          Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type  Field       Sub/File
Code                            1        I
Name                            2        F
```

Create the two properties shown in Figure 72:

- **Code**
- **Name**

For **Name**, put code in its **GET ACTION** field to retrieve the value from **DATA** (Figure 73).

**Figure 73: Exercise 8.1—Page 2: Entering Code in the GET ACTION Field**

```
                                Edit Entity
NAME: ZZZ CODED ELEMENT                                          Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                             .
.FIXED RESPONSE:                                                              .
.                                                                             .
.                                                                             .
.     GET ACTION: S VALUE=$G(DATA)                                            .
.                                                                             .
.                                                                             .
.                                                                             .
.                                                                             .
.                                                                             .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

**Close** this dialog, **Save** your changes, and **Exit** the form.

### 9.1.5 Step 5. Test Changes

Exit to programmer mode and run $$GET1^DDE again, noting the change to the **Gender** property:

**Figure 74: Exercise 8.1—Testing Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":"PROGRAMMER,ZZZ ONE",
"Gender":{"Code":"F", "Name":"FEMALE"}, "LastSignOn":"20220908111819-0500",
"Language":{"Code":"ENG", "Name":"ENGLISH"}, "State":"UT",
"PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePhone":"555-123-4567"},
"Divisions":[{"Name":"MEDICAL CENTER", "Default":"Yes"}, {"Name":"ALBANY OPC"}],
"Keys":["DGPF ASSIGNMENT", "ORES", "PROVIDER", "XUPROG", "XUPROGMODE"],
"Addresses":[{"Street":"123 MAIN ST", "City":"MIDVALE", "State":"UT",
"Zip":84047}, {"Street":"600 OAK ST", "City":"LOGAN", "State":"UT",
"Zip":84321}]}
>
```

End of Exercise 8.1.

## 9.2 Exercise 8.2: String as an Entity

An Entity usually acts on a pointer or code, expanding it into multiple attributes, but it can accept any string as its input value and act on it. A default source file is *not* required, but then all properties returned are generated by action code instead of field definitions.

### 9.2.1 Step 1. Create ZZZ NAME Entity

**Figure 75: Exercise 8.2—Creating ZZZ NAME Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NAME
  Located in the Z (Local) namespace.
  Are you adding 'ZZZ NAME' as a new ENTITY (the 285TH)? No// Y <Enter> (Yes)
Enter 'Name' for the Display Name but leave the Default File and query fields
empty. Save a Description such as:
  1>This Entity accepts a standard formatted name string (LNAME,FNAME MI)
  2>as the ID and returns the components using Kernel's XLFNAME utilities.
EDIT Option:
```

Press **<Enter>** to exit the editor, then enter **P** (**Previous**) to go to the previous page (Page 3) next.

**Figure 76: Exercise 8.2—Page 3: Entering Code in the GET EXIT ACTION and GET ID ACTION Fields**

```
                                Edit Entity
NAME: ZZZ NAME                                                    Page 3 of 3
------------------------------------------------------------------------------

GET ENTRY ACTION:
 GET EXIT ACTION: K ZZZNAME
   GET ID ACTION: K ZZZNAME S ZZZNAME=$G(DIEN) D NAMECOMP^XLFNAME(.ZZZNAME)
```

Remember that the **DIEN** variable holds the input value for the Entity; in this case it is the VistA name string. Enter a call to the Kernel XLFNAME utility in the **GET ID ACTION** field to parse the string into its components. Be sure to **KILL** the local variable used in the **GET EXIT ACTION** field.

Press **<Enter>** to get to the Command line and enter **P** (**Previous**) to go to Page 2 next.

**Figure 77: Exercise 8.2—Page 2: Creating Four Fixed String Properties**

```
                                Edit Entity
NAME: ZZZ NAME                                                    Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq    Type  Field       Sub/File
First                             1        F
Middle                            2        F
Last                              3        F
Suffix                            4        F
```

Create the four Fixed String (**F**) properties shown in Figure 77. Enter code into the **GET ACTION** field for each item to set **VALUE** to the appropriate node in the **ZZZNAME** array, such as:

- First: **S VALUE=$G(ZZZNAME("GIVEN"))**
- Middle: **S VALUE=$G(ZZZNAME("MIDDLE"))**
- Last: **S VALUE=$G(ZZZNAME("FAMILY"))**
- Suffix: **S VALUE=$G(ZZZNAME("SUFFIX"))**

**Save** your changes and **Exit** the form.

### 9.2.2 Step 2. Modify Name Property of ZZZ NEW PERSON

At the "Select ENTITY:" prompt enter "**ZZZ NEW PERSON**" and go to Page 2 of the form.

**Figure 78: Exercise 8.2—Page 2: Modifying Name Property of ZZZ NEW PERSON**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Divisions                       9       L                 200.02
Gender                          4       E   4             200
HomePhone                               S   .131         200
IEN                             1       I
Keys                            10      L
Language                        6       E   200.07        200
LastSignOn                      5       S   202           200
Name                            3       S   .01           200
OfficePhone                             S   .132         200
PhoneNumbers                    8       C
Source                          2       F
_______________________________________________________________________________
```

Use the arrow keys to drop down to the **Name** property. Press **<Enter> twice** to move to the **Type** column and change the current **S** value to **E**; press **<Enter>** again to open the "Entity" dialog (Figure 79).

**Figure 79: Exercise 8.2—Page 2 "Entity" Dialog: Entering Entity Name**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.          FIELD#: .01                      FILE#: 200                        .
.    EXT PTR LKUP:                                                           .
.  INTERNAL VALUE:                                                           .
.                                                                            .
.      GET ACTION:                                                           .
.                                                                            .
.OUTPUT TRANSFORM:                                                           .
. INPUT TRANSFORM:                                                           .
.                                                                            .
.          ENTITY: ZZZ NAME                                                  .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Enter "**ZZZ NAME**" at the **ENTITY** field. **Close** the dialog, **Save** your changes, and **Exit** the form.

### 9.2.3 Step 3. Test Changes

Exit to programmer mode and run $$GET1^DDE again, noting the changes to the **Name** property:

**Figure 80: Exercise 8.2—Testing Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":{"First":"ZZZ", Middle":"ONE",
"Last":"PROGRAMMER"}, "Gender":{"Code":"F", "Name":"FEMALE"},
"LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG", "Name":"ENGLISH"},
"State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePhone":"555-
123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER", "Default":"Yes"},
{"Name":"ALBANY OPC"}], "Keys":["DGPF ASSIGNMENT", "ORES", "PROVIDER", "XUPROG",
"XUPROGMODE"], "Addresses":[{"Street":"123 MAIN ST", "City":"MIDVALE",
"State":"UT", "Zip":84047}, {"Street":"600 OAK ST", "City":"LOGAN",
"State":"UT", "Zip":84321}]}
>
```

End of Exercise 8.2.

## 9.3 Exercise 8.3: Complex Group as an Entity

A nested Entity can also be used to return a group of independent fields from the default source file instead of using a complex group. An Entity can be preferred if many fields are needed, or if the group could be re-used in other entities. Its input value is the same as the primary Entity's record identifier.

### 9.3.1 Step 1. Create ZZZ ADDRESS Entity

**Figure 81: Exercise 8.3—Creating the ZZZ ADDRESS Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ ADDRESS
  Are you adding 'ZZZ ADDRESS' as a new ENTITY (the 282TH)? No// Y <Enter> (Yes)
```

Repeat the exercises in Lesson 1 to create a new Entity. Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option to create another Entity to pull from the NEW PERSON (#200) file.

**Figure 82: Exercise 8.3—Page 1: Editing ZZZ ADDRESS Entity**

```
                                Edit Entity
NAME: ZZZ ADDRESS                                                  Page 1 of 3
------------------------------------------------------------------------------

         NAME: ZZZ ADDRESS
 DISPLAY NAME: Address

 DEFAULT FILE: 200
```

Enter a description, then enter **N** (**Next**) at the Command prompt to continue to Page 2 of the form.

### 9.3.2 Step 2. Add Four Simple Properties

Create the following properties; refer to Exercise 2.3 in Lesson 2 if needed:

- **Street**
- **City**
- **State**
- **Zip**

**Figure 83: Exercise 8.3—Page 2: Adding Four Simple Properties**

```
                                Edit Entity
NAME: ZZZ ADDRESS                                                  Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type   Field        Sub/File
Street                          1        S    .111         200
City                            2        S    .114         200
State                           3        S    .115         200
Zip                             4        S    .116         200
```

Use the arrow keys to drop down to the Command prompt, then enter **S** to **Save** your work. Enter **E** to **Exit** the edit form for this Entity.

### 9.3.3 Step 3. Create Address Property

At the "Select ENTITY:" prompt, choose the **ZZZ NEW PERSON** Entity and go to Page 2 of the edit form.

**Figure 84: Exercise 8.3—Page 2: Creating Address Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type  Field       Sub/File
Gender                          4        S   4           200
HomePhone                                S   .131        200
IEN                             1        I
Keys                            10       L
Language                        6        E   200.07      200
LastSignOn                      5        S   202         200
Name                            3        S   .01         200
OfficePhone                              S   .132        200
PhoneNumbers                    8        C
Source                          2        F
State                           7        S   .115        200
Address                         11       E
_______________________________________________________________________________
```

Use the arrow keys to go to the bottom of the list, then add the **Address** property; assign a **Seq** of **11** and enter **E** for the **Type**. Press **<Enter>** to open the "Entity" dialog (Figure 85).

### 9.3.4 Step 4. Complete Item Attributes

**Figure 85: Exercise 8.3—Page 2 "Entity" Dialog: Completing Item Attributes**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.          FIELD#:                          FILE#: 200                        .
.    EXT PTR LKUP:                                                           .
.  INTERNAL VALUE:                                                           .
.                                                                            .
.      GET ACTION: S VALUE=DIEN                                              .
.                                                                            .
.OUTPUT TRANSFORM:                                                           .
. INPUT TRANSFORM:                                                           .
.                                                                            .
.          ENTITY: ZZZ ADDRESS                                               .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Use the arrow keys or press **<Enter>** to skip over the field values and go right to the "GET ACTION:" prompt. This time the value needed for the nested Entity is *not* a specific field value but the same record ID already in use; so, rather than defining a field value, you simply set the **VALUE** variable to the current record number (**DIEN**).

Enter "**ZZZ ADDRESS**" at the "ENTITY:" prompt. **Close** the dialog and return to Page 2 of the form. **Save** your changes and **Exit**.

### 9.3.5 Step 5. Test Changes

Exit to programmer mode and run $$GET1^DDE again to view the results.

**Figure 86: Exercise 8.3—Testing Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":"PROGRAMMER,ZZZ ONE",
"Gender":"FEMALE", "LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG",
"Name":"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309",
"OfficePhone":"555-123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER",
"Default":"Yes"}, {"Name":"ALBANY OPC"}], "Keys":["DGPF ASSIGNMENT", "ORES",
"PROVIDER", "XUPROG", "XUPROGMODE"], "Address":{"Street":"123 MAIN ST",
"City":"MIDVALE", "State":"UT", "Zip":84047}}}
>
```

End of Exercise 8.3.

## 9.4 Exercise 8.4: Complex Group as a List

Sometimes a data model may expect values to be returned as a list that are stored as separate independent fields in VistA. In XML, lists and groups look the same, but JSON uses different brackets for lists vs. groups, so a nested Entity will *not* produce the desired results. DDE supports a List type of **Complex** for this purpose.

### 9.4.1 Step 1. Create ZZZ TEMP ADDRESS Entity

**Figure 87: Exercise 8.4—Creating the ZZZ TEMP ADDRESS Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ TEMP ADDRESS
  Are you adding 'ZZZ TEMP ADDRESS' as a new ENTITY (the 283TH)? No// Y <Enter>
(Yes)
```

Repeat Exercise 8.1 to create another address Entity. Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option to create an Entity to pull the temporary address fields from the NEW PERSON (#200) file.

**Figure 88: Exercise 8.4—Page 1: Editing the ZZZ TEMP ADDRESS Entity**

```
                                Edit Entity
NAME: ZZZ TEMP ADDRESS                                          Page 1 of 3
------------------------------------------------------------------------------

          NAME: ZZZ TEMP ADDRESS
 DISPLAY NAME: Address

 DEFAULT FILE: 200
```

Use the same **DISPLAY NAME** of "**Address**" as before. Enter a description, then enter **N** (**Next**) at the Command prompt to continue to Page 2 of the form.

### 9.4.2 Step 2. Add Four Simple Properties

Create the same properties as **ZZZ ADDRESS**, but this time assign the Temporary Address field numbers.

**Figure 89: Exercise 8.4—Page 2: Adding Four Simple Properties**

```
                                Edit Entity
NAME: ZZZ TEMP ADDRESS                                          Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq    Type   Field        Sub/File
Street                            1        S    .1211        200
City                              2        S    .1214        200
State                             3        S    .1215        200
Zip                               4        S    .1216        200
```

Use the arrow keys to drop down to the Command prompt, then enter **S** to **Save** your work. Enter **E** to **Exit** the edit form for this Entity.

### 9.4.3 Step 3. Modify Address Property in ZZZ NEW PERSON

At the "Select ENTITY:" prompt, choose the **ZZZ NEW PERSON** Entity and go to Page 2 of the edit form.

**Figure 90: Exercise 8.4—Modifying Address Property in the ZZZ NEW PERSON Entity**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                            Page 2 of 3
------------------------------------------------------------------------------

Item                              Seq    Type   Field        Sub/File
PermanentAddress                         E                   200
```

The cursor should appear on the **Address** property in the first line. Change its name to "**PermanentAddress**" as shown in Figure 90. Press **<Enter>** to jump to the **Seq** column, then enter an **at-sign** (**@**) to remove the value. This item will become part of the new **Addresses** list group in a later step (refer to Exercise 4.3 if needed).

Use the arrow keys to move to the Command line and enter **S** to **Save** your changes.

### 9.4.4 Step 4. Add Temporary Address Property

From the Command line, use the up arrow to move to the end of the **Item** list. Create a new property called "**TemporaryAddress**" as shown in Figure 91, again omitting a **Seq** number.

**Figure 91: Exercise 8.4—Page 2: Adding Temporary Address Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
Gender                          4       S   4           200
HomePhone                               S   .131        200
IEN                             1       I
Keys                            10      L
Language                        6       E   200.07      200
LastSignOn                      5       S   202         200
Name                            3       S   .01         200
OfficePhone                             S   .132        200
PhoneNumbers                    8       C
Source                          2       F
State                           7       S   .115        200
TemporaryAddress                        E
_______________________________________________________________________________
```

Enter **E** for the **Type** and press **<Enter>** to open the "Entity" dialog (Figure 92).

**Figure 92: Exercise 8.4—Page 2 "Entity" Dialog: Selecting ZZZ TEMP ADDRESS Entity**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.          FIELD#:                          FILE#: 200                        .
.    EXT PTR LKUP:                                                           .
.  INTERNAL VALUE:                                                           .
.                                                                            .
.      GET ACTION: S VALUE=DIEN                                              .
.                                                                            .
.OUTPUT TRANSFORM:                                                           .
. INPUT TRANSFORM:                                                           .
.                                                                            .
.          ENTITY: ZZZ TEMP ADDRESS                                          .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Enter the same code into the **GET ACTION** field as with **ZZZ ADDRESS** but select **ZZZ TEMP ADDRESS** for the Entity.

**Close** the dialog to return to Page 2 of the form.

### 9.4.5 Step 5. Add Addresses List Group

The cursor should be at the bottom of the item list. Create a new list property called "**Addresses**" (Figure 93).

**Figure 93: Exercise 8.4—Adding Addresses List Group**

```
TemporaryAddress                        E              200
Addresses                    11         L
```

Assign a **Seq** of **11** and a **Type** of **L**. Press **<Enter>** to open the "List" dialog (Figure 94).

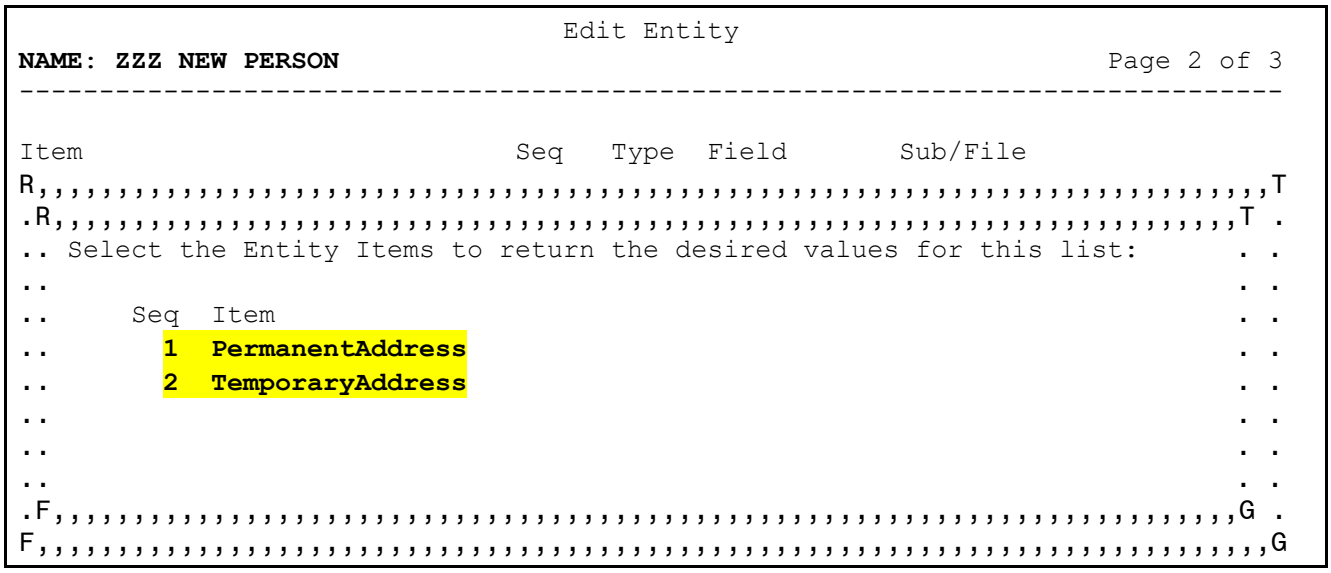
**Figure 94: Exercise 8.4—Page 2 "List" Dialog: Selecting COMPLEX for the LIST TYPE**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                           Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: COMPLEX                                                         .
.                                                                            .
.GET ACTION:                                                                 .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY:                                    FIELD#:                      .
.                                              EXT PTR:                      .
.                                              INT VAL:                      .
.                                                                            .
.   XML TAG:                                                                 .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

Select **COMPLEX** for the **LIST TYPE**. This opens the "Group Items" dialog.

**Figure 95: Exercise 8.4—Page 2 "Group Items" Dialog: Entering Address Properties**




Enter the following two address properties:

- **PermanentAddress**
- **TemporaryAddress**

Assign a **Seq** number inside the group for each property, answering **YES** when asked if adding a new Complex Type, and then enter the name of the item when prompted for the Complex Item Name. **Close** this dialog when finished.

**Figure 96: Exercise 8.4—Page 2 "List" Dialog : Editing COMPLEX List Type**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                          Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
. LIST TYPE: COMPLEX                                                         .
.                                                                            .
.GET ACTION:                                                                 .
.                                                                            .
. Select the Entity or Field to be returned for each record:                 .
.    ENTITY:                                   FIELD#:                       .
.                                             EXT PTR:                       .
.                                             INT VAL:                       .
.                                                                            .
.   XML TAG: Address                                                         .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

No additional attributes are needed for a Complex group list in the "List" dialog. If using XML, you can define the **XML TAG** property to ensure all instances use the same tags. (Refer to Exercise 5.1 if needed.)

**Close** this dialog to return to the "Edit" form. **Save** your changes and **Exit** the form.

## 9.4.6 Step 6. Test Entity

Exit VA FileMan to programmer mode, and use $$GET1^DDE to test your changes.

**Figure 97: Exercise 8.4—Testing Entity: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":"PROGRAMMER,ZZZ ONE",
"Gender":"FEMALE", "LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG",
"Name":"ENGLISH"}, "State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309",
"OfficePhone":"555-123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER",
"Default":"Yes"}, {"Name":"ALBANY OPC"}], "Keys":["DGPF ASSIGNMENT", "ORES",
"PROVIDER", "XUPROG", "XUPROGMODE"], "Addresses":[{"Street":"123 MAIN ST",
"City":"MIDVALE", "State":"UT", "Zip":84047}, {"Street":"600 OAK ST",
"City":"LOGAN", "State":"UT", "Zip":84321
}]}
>
```

End of Exercise 8.4.

## 9.5 Exercise 8.5: External Field

Sometimes a field needs to be returned that is *not* in the source file or its pointed-to files. DDE allows any field in any file to be returned in an Entity but including a field outside of the default source file requires changing the record identifier temporarily.

### 9.5.1 Step 1. Open ZZZ NEW PERSON Entity

**Figure 98: Exercise 8.5—Opening the ZZZ NEW PERSON Entity**

```
VA FileMan 22.2


Select OPTION: OTHER OPTIONS
Select OTHER OPTION: DATA MAPPING
Select DATA MAPPING OPTION: ENTER/EDIT AN ENTITY
Select ENTITY: ZZZ NEW PERSON
```

Use the **Enter/Edit an Entity** [DDE ENTITY ENTER/EDIT] option again and select the **ZZZ NEW PERSON** Entity. Use the arrow keys to get to the Command prompt and enter **N** (**Next**) to go to the next page.

### 9.5.2 Step 2. Create DefaultLocation Property

Use the arrow keys to go to the bottom of the "Item" list and enter "**DefaultLocation**"; answer **YES** when prompted if adding a new Item.

**Figure 99: Exercise 8.5—Page 2: Creating DefaultLocation Property**

```
                              Edit Entity
NAME: ZZZ NEW PERSON                                              Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type  Field       Sub/File
Keys                            10       L
Language                        6        E   200.07      200
LastSignOn                      5        S   202         200
Name                            3        E   .01         200
OfficePhone                              S   .132        200
PermanentAddress                         E               200
PhoneNumbers                    8        C
Source                          2        F
State                           7        S   .115        200
TemporaryAddress                         E               200
DefaultLocation                 12       S
```

Press **<Enter>** and assign a **Seq** of **12**; press **<Enter>** again to jump to the **Type** column and enter **S**. Press **<Enter>** a third time to open the "Simple Field" dialog (Figure 100).

### 9.5.3 Step 3. Complete DefaultLocation Property

**Figure 100: Exercise 8.5—Page 2 "Simple Field" Dialog: Completing DefaultLocation Property**

```
                                  Edit Entity
NAME: ZZZ NEW PERSON                                                  Page 2 of 3
------------------------------------------------------------------------------

Item                             Seq   Type  Field       Sub/File
R,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,T
.                                                                              .
.           FIELD#: .02                      FILE#: 8926                       .
.     EXT PTR LKUP:                                                            .
.   INTERNAL VALUE:                                                            .
.                                                                              .
.       GET ACTION: S IEN=+$O(^TIU(8926,"B",DIEN,0)) I IEN<1 S DDEOUT=1        .
.                                                                              .
.OUTPUT TRANSFORM:                                                             .
. INPUT TRANSFORM:                                                             .
.                                                                              .
F,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,G
```

The cursor is initially placed at the "FILE#:" prompt; this field gets a default value from the Default File Number of the Entity, but it is editable. For this example, you will pull the user's Default Location from the Text Integration Utilities (TIU) Personal Preferences file, so change the value to **8926**. Press **<Enter>** to jump to the "FIELD#" prompt and enter **.02** for the Default Location field.

Since this file is *not* DINUM'd to the NEW PERSON (#200) file, you need to tell DDE which record to use. The **IEN** variable holds the current record number when processing each property; it is defaulted to the value of **DIEN** but can be changed if a record in an external file needs to be accessed. It is scoped for a single property, so its value returns to match **DIEN** for the next property in sequence.

It is *recommended* to verify that **IEN** has a valid value, and if not, set **DDEOUT** to quit the property.

**Close** the dialog and return to Page 2 of the form (Figure 101).

**Figure 101: Exercise 8.5—Page 2: Completed DefaultLocation Property**

```
                                Edit Entity
NAME: ZZZ NEW PERSON                                                Page 2 of 3
------------------------------------------------------------------------------

Item                            Seq    Type   Field       Sub/File
Keys                            10       L
Language                        6        E    200.07      200
LastSignOn                      5        S    202         200
Name                            3        E    .01         200
OfficePhone                              S    .132        200
PermanentAddress                         E                200
PhoneNumbers                    8        C
Source                          2        F
State                           7        S    .115        200
TemporaryAddress                         E                200
DefaultLocation                 12       S    .02         8926
```

You will notice the Field and File numbers are now displayed for the **DefaultLocation** prompt.

If there are multiple instances of the desired value in an external file, for example if a subfile was moved out to a file to flatten the data dictionary, then a **List** property of type **File** can be created. These files often include an indexed field that points to the primary file; that index name can be saved as the list's cross-reference, and the search value would be the current record identifier in the primary file. which is in the **DIEN** variable. You can save a variable name in the **Filter By** instead of a static string. Complete the rest of the list property as shown in Lesson 4 exercises.

## 9.5.4 Step 4. Test Changes

Exit VA FileMan to programmer mode and use $$GET1^DDE to test your changes, noting the new property.

**Figure 102: Exercise 8.5—Testing Changes: Running $$GET1^DDE**

```
>W $$GET1^DDE("ZZZ NEW PERSON",DUZ)
"NewPerson":{"IEN":11948, "Source":500, "Name":{"First":"ZZZ", Middle":"ONE",
"Last":"PROGRAMMER"}, "Gender":{"Code":"F", "Name":"FEMALE"},
"LastSignOn":"20220908111819-0500", "Language":{"Code":"ENG", "Name":"ENGLISH"},
"State":"UT", "PhoneNumbers":{"HomePhone":"555-867-5309", "OfficePhone":"555-
123-4567"}, "Divisions":[{"Name":"MEDICAL CENTER", "Default":"Yes"},
{"Name":"ALBANY OPC"}], "Keys":["DGPF ASSIGNMENT", "ORES", "PROVIDER", "XUPROG",
"XUPROGMODE"], "Addresses":[{"Street":"123 MAIN ST", "City":"MIDVALE",
"State":"UT", "Zip":84047}, {"Street":"600 OAK ST", "City":"LOGAN",
"State":"UT", "Zip":84321}], "DefaultLocation":"GENERAL MEDICINE"}
>
```

End of Exercise 8.5.

## 9.6 Exercise 8.6: Dynamic Query Parameters

Entities can be set up to find specific defined records or provide a more general solution. The GET^DDE call accepts a list parameter by reference of dynamic search filters. Some standard filters are provided, but a developer can pass in any filter that the Entity has been coded to accept.

### 9.6.1 Step 1. Edit Query Routine to Accept a Key

GET^DDE can accept a list of search parameters in the form **FILTER("name")=value**, which is passed by reference into the third input parameter. Filters, such as "**start**" and "**stop**", are provided by DDE, and their values can be referenced in the **DSTRT** and **DSTOP** variables respectively in the Query Routine.

**REF:** For full details on how to use the query parameters, see the *VA FileMan Developers Guide*.

Any filter value can be passed into DDE; however, those filter values *must* be coded in the Entity's **QUERY ROUTINE**. At the time the query routine is executed, the parameters are available in the **FILTER** array.

Use your preferred editor to modify the FIND^ZZZDEMO query routine to look for a key name in **FILTER("key")** to use in place of PROVIDER. Figure 103 shows a possible solution:

**Figure 103: Exercise 8.6—Editing Query Routine to Accept a Key**

```
ZZZDEMO  ;OIT/STAFF - DDE DEMO
         ;;DEMO
FIND     ;demo query routine
         N NUM,NAME,IEN,KEY
         S NUM=0,KEY=$G(FILTER("key"),"PROVIDER")
         S NAME="" F  S NAME=$O(^VA(200,"AK."_KEY,NAME)) Q:NAME=""  D
         . S IEN=0 F  S IEN=$O(^VA(200,"AK."_KEY,NAME,IEN)) Q:'IEN  D
         .. S NUM=NUM+1,DLIST(NUM)=IEN
         Q
```

This code looks for the desired key to use to find a list of users, and defaults to PROVIDER if no key is passed into it.

### 9.6.2 Step 2. Test Changes in Programmer Mode

Run your query using different values for the key. Remember that this code is intended to create the **DLIST** array and leave it defined for DDE, so be sure to **KILL** it between test runs.

**Figure 104: Exercise 8.6—Testing Changes in Programmer Mode**

```
>K DLIST D FIND^ZZZDEMO ZW DLIST ;default=PROVIDER
DLIST(1)=11295
DLIST(2)=11960
…
DLIST(170)=1006

>K DLIST S FILTER("key")="HLOMGR" D FIND^ZZZDEMO ZW DLIST
DLIST(1)=11968
DLIST(2)=11964
DLIST(3)=11908
DLIST(4)=11927
DLIST(5)=11967

>
```

### 9.6.3 Step 3. Test Entity

Run GET^DDE in programmer mode, with and without a specified search key.

**Figure 105: Exercise 8.6—Testing Entity: Running GET^DDE**

```
>D GET^DDE("ZZZ NEW PERSON"),^%G

For help on global specifications DO HELP^%G
Global ^TMP("DDE GET",$J
^TMP("DDE GET",29759,0)=170
                    1)="{""IEN"":11295, ""Source"":500, ""Name"":{""First"":""G
EORGE"", ""Last"":""ABNERATHY""}, ""Gender"":{""Code"":""F"", ""Name"":""FEMALE"
"}, ""Keys"":[""PROVIDER""], ""DefaultLocation"":""GENERAL MEDICINE""}"
                    2)="{""IEN"":11960, ""Source"":500, ""Name"":{""First"":""S
HIVA"", ""Last"":""ACHARYA""}, ""Gender"":{""Code"":""F"", ""Name"":""FEMALE""},
 ""LastSignOn"":20190916, ""Keys"":[""PROVIDER""], ""DefaultLocation"":""GENERAL
 MEDICINE""}"
…
^TMP("DDE GET",29759,170)="{""IEN"":1006, ""Source"":500, ""Name"":{""First"":""
PERRY"", ""Last"":""ZZTONSILS""}, ""Gender"":{""Code"":""M"", ""Name"":""MALE""}
, ""Keys"":[""PROVIDER""], ""DefaultLocation"":""GENERAL MEDICINE""}"

Global ^
```

**At this point, the user presses the <Enter> key to exit.**

```
>S QUERY("key")="HLOMGR"

>D GET^DDE("ZZZ NEW PERSON",,.QUERY),^%G

For help on global specifications DO HELP^%G
Global ^TMP("DDE GET",$J
^TMP("DDE GET",29759,0)=5
                    1)="{""IEN"":11968, ""Source"":500, ""Name"":{""First"":""O
RLANDO"", ""Middle"":""E"", ""Last"":""CRUZ""}, ""Gender"":{""Code"":""M"", ""Na
me"":""MALE""}, ""LastSignOn"":""20200909114252-0500"", ""Keys"":[""HLOMAIN"", "
"HLOMGR"", ""IB AUTHORIZE"", ""IB EDIT"", ""IB EDIT PAY-TO"", ""IB EDIT PAY-TO T
C"", ""XUMGR"", ""XUPROG"", ""XUPROGMODE""], ""DefaultLocation"":""GENERAL MEDIC
INE""}"
…

Global ^
```

**At this point, the user presses the <Enter> key to exit.**

```
>
```

End of Exercise 8.6.