

RETRIEVE/STUFF FIELDS

\$\$GET^DDSVVAL()

You can use this entry point only within a ScreenMan form, in all places where M code can be placed on the form.

This extrinsic function retrieves data from a data dictionary field. If the user has edited the field on the ScreenMan form, or if the form designer has modified the field with a PUT^DDSVVAL call, the function returns the new value, even if the user has not yet saved the change to the database. If the field has not been edited on the Screenman form, the function retrieves the data from the FileMan file/global.

Text for a word processing field is moved into a global array and \$\$GET^DDSVVAL returns the closed root of that array. The array has the same format as a FileMan word processing field.

Computed fields in FileMan files cannot be retrieved. To retrieve the value of a computed field defined on the form, use the \$\$GET^DDSVVALF function described below.

If, while a form is running, a call to \$\$GET^DDSVVAL fails, ScreenMan prints an error message in the Command Area.

Format

```
$$GET^DDSVVAL(FILE, [.]RECORD, FIELD, .ERROR, FLAGS)
```

Input Parameters

- | | |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FILE | (Required) The global root or number of the file or subfile. At the field level, the local variable DIE contains the current global root. |
| [.]RECORD | (Required) The internal entry number or an array of internal entry numbers. This parameter has the same form as the DA array. At the field level, the local array DA contains the current array of internal entry numbers. |
| FIELD | (Required) The field name or number or a relational expression that follows a forward pointer (e.g., POINTER:FIELD). The |

"Forward Pointers" section of the "ScreenMan Forms" chapter in this manual describes in detail the syntax accepted by ScreenMan to describe a relational jump via a DD field.

.ERROR (Optional) \$D(ERROR)>1, if the function call fails.

FLAGS (Optional) Controls whether the internal or external form is returned, as shown below (the I and E flags have no effect if FIELD is a word processing field):

I Return the **I**nternal form of the data. (Default)

E Return the **E**xternal form of the data.

Examples

Example 1

Retrieve the internal form of the .01 field of the record currently being edited:

```
S nmspNAME=$$GET^DDSVAl(DIE,.DA,.01)
```

Example 2

Retrieve the external form of field #20, record #362, in file #16000:

```
S nmspDATE=$$GET^DDSVAl("^DIZ(16000, ",362,20,"","E")
```

Example 3

Retrieve the text contained in a word processing field named DESCRIPTION:

```
S nmspWP=$$GET^DDSVAl(DIE,.DA,"DESCRIPTION")
```

The text of the DESCRIPTION field is moved to the array as follows:

```
@nmspWP@(0)=Header node of word processing field
@nmspWP@(1,0)=Line 1
@nmspWP@(2,0)=Line 2
...etc.
```

PUT^DDSVVAL()

You can use this entry point only within a ScreenMan form, in all places where M code can be placed on the form.

This procedure stuffs data into a data dictionary field as part of ScreenMan's transaction. The data passed to this procedure is filed in the database only when the user explicitly saves changes. Until then, it is stored in a temporary location.

If the specified field is a word processing field, the value passed to the procedure is the closed root of the array that contains the text.

If the specified field is a multiple field, the value passed is the subrecord first displayed to the user as a default at the multiple field. This value is a default for selection and is not actually filed.

Values cannot be stuffed into computed fields.

If, while a form is running, a call to PUT^DDSVVAL fails, ScreenMan prints an error message in the Command Area.

Format

```
PUT^DDSVVAL( FILE, [ . ]RECORD, FIELD, VALUE, .ERROR, FLAGS )
```

Input Parameters

- | | |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FILE | (Required) The global root or number of the file or subfile. At the field level, the local variable DIE contains the current global root. |
| [.]RECORD | (Required) The internal entry number or an array of internal entry numbers. This parameter has the same form as the DA array. At the field level, the local array DA contains the current array of record numbers. |
| FIELD | (Required) The field name or number. |
| VALUE | (Required) The value to stuff into the data dictionary field. If FLAGS (described below) does not contain an I, the value must be in the form of a valid, unambiguous user response. |

If FIELD is a word processing field, VALUE must be the closed root of the array that contains the text. The subscripts of the nodes below the root must be positive numbers, although they need not be integers, and there can be gaps in the sequence. The text must be in these nodes or in the 0 node descendent from these nodes.

If FIELD is a multiple field, VALUE determines the subrecord to display to the user as a default for selection. It is not a value that is ever filed. VALUE can be "FIRST", "LAST", or the specific internal entry number of the subrecord to display. "FIRST" indicates that the subrecord with the lowest internal entry number should be displayed and "LAST" indicates that the subrecord with the highest internal entry number should be displayed.

.ERROR (Optional) \$D(ERROR)>1, if the procedure call fails.

FLAGS Indicates whether VALUE is in internal or external form, as shown below:

A Append new word processing text to the current text. This flag can be used only when stuffing text into a word processing field. If the A flag is not sent, the current word processing text is completely erased before the new text is added.

I VALUE is in Internal form; it is not validated.

E VALUE is in External form. (Default)

The I and E flags have no effect when FIELD is a word processing field.

Examples

Example 1

Stuff the value 2940801 into a date field #20. The value passed is in internal form:

```
D PUT^DDSVAl(DIE, .DA, 20, 2940801, "", "I")
```

No data validation is performed.

Example 2

Stuff word processing text from an array into a word processing field named DESCRIPTION as shown below:

```
D PUT^DDSVAL(DIE,.DA,"DESCRIPTION", "^nmspWP("TEXT")")
```

The array that contains the text looks like:

```
^nmspWP("TEXT",1,0)=Line 1  
^nmspWP("TEXT",2,0)=Line 2  
...etc.
```

OR:

```
^nmspWP("TEXT",1)=Line 1  
^nmspWP("TEXT",2)=Line 2  
...etc.
```

HELP MESSAGES

HLP^DDSUTL()

You can use this entry point only within a ScreenMan form, at all places where M code can be placed on the form.

This procedure prints messages in the Command Area.

If you pass the string "\$\$EOP", then ScreenMan will issue the prompt "Press RETURN to continue" in the Command Area. This is useful if, for example, you want to print messages as part of the post action of a page, and need to pause to give the user a chance to read the messages before ScreenMan leaves that page.

Formats

1. HLP^DDSUTL(STRING)
2. HLP^DDSUTL(.STRING)

Input Variables

- STRING** (Required) The message to print in the Command Area.
- .STRING** (Required) An array of messages to print in the Command Area. STRING(1), STRING(2), ..., STRING(n) each contain a line of text.