

Manual: FDO91 Manual

Chapter 15: Data Manager Protocol defines data manager protocol atoms and provides information for handling data extraction.

Last updated: January 1994

CHAPTER 15

Data Manager Protocol

The Data Manager protocol (protocol ID 3) consists of atoms that provide the ability to extract data from fields in forms for local processing or to be sent to the host. Various data types and extraction options are supported. There is also support for input field validation.

Data Manager Protocol Atoms

The Data Manager protocol atoms are described in alphabetical order in the rest of this chapter.

atom\$de_custom_data

11 (\$0B)

Description

atom\$de_custom_data is sent to the host from the data manager if an optional 32-bit value is passed with the **atom\$de_ez_send_form** atom. This atom is a host-bound atom only.

Syntax

```
atom$de_custom_data <32_bit_val>
```

<32_bit_val> Specifies a 32-bit custom value.

Return Value

None

Example

```
atom$de_custom_data <32_bit_val>
```

atom\$de_data

1 (\$01)

Description

atom\$de_data is sent to the host from the Data Manager Tool when it is sending extracted data. This atom is a host-bound atom only.

Syntax

atom\$de_data <data>

<data> Specifies variable length data.

Return Value

None

Example

atom\$de_data <data>

atom\$de_end_extraction

7 (\$07)

Description

atom\$de_end_extraction signifies the end of an extraction sequence.

Syntax

`atom$de_end_extraction`

Return Value

Unchanged

Example

`atom$de_end_extraction`

atom\$de_ez_send_field

15 (\$0F)

Description

atom\$de_ez_send_field is a shorthand atom that sends the data from the current object to the host.

Syntax

```
atom$de_ez_send_field <token>
```

<token> Specifies a token to send to host with data.

Return Value

Unchanged

Example

```
atom$de_ez_send_field <token>
```

atom\$de_ez_send_form

10 (\$0A)

Description

atom\$de_ez_send_form is a shorthand atom that sends the data from all the sendable children of the current object to the host.

Syntax

```
atom$de_ez_send_form <token> [<32_bit_val>]
```

<token> Specifies a token to send to host.

<32_bit_val> Specifies an optional 32-bit custom value.

Return Value

Unchanged

Example

```
atom$de_ez_send_form <token> [<32_bit_val>]
```

atom\$de_ez_send_list_index

14 (\$0E)

Description

atom\$de_ez_send_list_index is a shorthand atom that sends the index of the selected item of the current list box or the list box specified by the optional relative ID parameter.

Syntax

```
atom$de_ez_send_list_index <token> [<relative_ID>]
```

<token> Specifies a token to send to host.

<relative_ID> Specifies an optional relative ID.

Return Value

Unchanged

Example

```
atom$de_ez_send_list_index <token> [<relative_ID>]
```


atom\$de_ez_send_list_text

12 (\$0C)

Description

atom\$de_ez_send_list_text is a shorthand atom that sends the text of the currently selected item from the current list box or the list box specified by the optional relative ID parameter.

Syntax

```
atom$de_ez_send_list_text <token> [<relative_ID>]
```

<token> Specifies a token to send to host.

<relative_ID> Specifies an optional relative ID.

Return Value

Unchanged

Example

```
atom$de_ez_send_list_text <token> [<relative_ID>]
```

atom\$de_get_data

6 (\$06)

Description

atom\$de_get_data gets data according to the preset state of the Data Manager Tool and sends it to the host.

Syntax

```
atom$de_get_data [<extract_method>]
```

<extract_method> Specifies an optional extraction method. Values are as follows:

0	EM_DEFAULT
1	EM_CURRENT_OBJECT
2	EM_SELECTED_CHILDREN
3	EM_CUR_OBJ_SELECTED_CHILDREN
4	EM_ALL_CHILDREN
5	EM_CUR_OBJ_ALL_CHILDREN

Return Value

Unchanged

Example

```
atom$de_get_data 1
```

atom\$de_get_data_pointer

9 (\$09)

Description

atom\$de_get_data_pointer gets data and returns a pointer to the data according to the preset state of the Data Manager Tool.

Syntax

```
atom$de_get_data_pointer [<extract_method>]
```

<extract_method> Specifies an optional extraction method. Values as follows:

0	EM_DEFAULT
1	EM_CURRENT_OBJECT
2	EM_SELECTED_CHILDREN
3	EM_CUR_OBJ_SELECTED_CHILDREN
4	EM_ALL_CHILDREN
5	EM_CUR_OBJ_ALL_CHILDREN

Return Value

A pointer to the extracted data.

Example

```
atom$de_get_data_pointer 4
```

atom\$de_get_data_value

8 (\$08)

Description

atom\$de_get_data_value gets data and returns the value of the data according to the preset state of the Data Manager Tool.

Syntax

```
atom$de_get_data_value [<extract_method>]
```

<extract_method> Specifies an optional extraction method values as follows:

0	EM_DEFAULT
1	EM_CURRENT_OBJECT
2	EM_SELECTED_CHILDREN
3	EM_CUR_OBJ_SELECTED_CHILDREN
4	EM_ALL_CHILDREN
5	EM_CUR_OBJ_ALL_CHILDREN

Return Value

The extracted value.

Example

```
atom$de_get_data_value 1
```

atom\$de_set_data_type

3 (\$03)

Description

atom\$de_set_data_type sets the data type to be extracted by the next **atom\$de_get_data** command.

Syntax

atom\$de_set_data_type <data_type>

<data_type>	Specifies a data type. Values are as follows:
0	DATATYPE_DEFAULT
1	DATATYPE_TEXT
2	DATATYPE_VAR
3	DATATYPE_BOOLEAN
4	DATATYPE_GLOBAL_ID
5	DATATYPE_RELATIVE_ID
6	DATATYPE_INDEX
7	DATATYPE_CHILD_COUNT
8	DATATYPE_OBJPTR
9	DATATYPE_VALUE
10	DATATYPE_RAW

Return Value

Unchanged

Example

```
atom$de_set_data_type 1
```

atom\$de_set_extraction_type

0 (\$00)

Description

atom\$de_set_extraction_type is sent to the host on an **atom\$de_get_data** to tell the host the extraction method used.

This atom is a host-bound atom only.

Syntax

```
atom$de_set_extraction_type <extract_method>
```

<extract_method> Specifies an extraction method. Values are as follows:

0	EM_DEFAULT
1	EM_CURRENT_OBJECT
2	EM_SELECTED_CHILDREN
3	EM_CUR_OBJ_SELECTED_CHILDREN
4	EM_ALL_CHILDREN
5	EM_CUR_OBJ_ALL_CHILDREN

Return Value

Unchanged

Example

```
atom$de_set_extraction_type 1
```

atom\$de_set_text_column

5 (\$05)

Description

atom\$de_set_text_column sets the column (tab stop) from which text is extracted on the next **atom\$de_get_data** command.

Syntax

```
atom$de_set_text_column <number>
```

<number> Specifies a zero-based column number.

Return Value

Unchanged

Example

```
atom$de_set_text_column <number>
```


atom\$de_set_variable_id

4 (\$04)

Description

atom\$de_set_variable_id sets the ID of the variable to be extracted on the next **atom\$de_get_data** command.

Parameters

```
atom$de_set_variable_id <var_id>
```

<var_id> Specifies the variable ID.

Return Value

Unchanged

Example

```
atom$de_set_variable_id <var_ID>
```

atom\$de_start_extraction

2 (\$02)

Description

atom\$de_start_extraction initializes the Data Manager Tool for the current stream. This atom is always used to start extraction.

Syntax

```
atom$de_start_extraction [<buf_flag>]
```

<buf_flag> Specifies optional buffer flags. For flag values, see the **atom\$buf_set_flag** atom in Chapter 6, "The Buffer Tool." Default flags are for sending data to the host. A zero value can be used for local operations.

Return Value

Unchanged

Example

```
atom$de_start_extraction <buf_flag>
```

atom\$de_validate

16 (\$10)

Description

atom\$de_validate causes the member input fields in the current object tree to be validated.

Syntax

```
atom$de_validate <action_flags>
```

<action_flags> Specifies one or more optional actions to be taken when a field that is not valid is encountered. The following flags can be combined:

DVF_DISPLAY_MESSAGE (01) - Displays a message box informing the member that a field in error was detected.

DVF_TERMINATE (02) - Performs a DE_END_EXTRACTION and terminates the current stream.

Return Value

0 (zero) if the input is valid, non-zero otherwise.

Example

```
atom$de_validate <action_flags>
```