

Manual: FDO91 Manual

Appendix B: Describes the FDO91 internationalization (i18n) atoms.

Last updated: May 1996

A P P E N D I X B

Internationalization (i18n) Atoms

The FDO91 internationalization atoms are used to localize data sent from the host system to international clients of the online service. These atoms localize text, date/time formats, and data types.

Data sent from the host is localized according to time zone and language preferences specified within the member preferences area of the online service. Host data is also localized according to date and time settings within the control panel(s) of the client computer operating system, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

The FDO91 internationalization atoms are described in alphabetical order in the rest of this appendix.

atom\$async_alert_add_date_time

33 (\$21)

Description

atom\$async_alert_add_date_time converts a host time stamp to a localized date/time string and adds the string to the alert message being built. This atom is sent by the host. The date and time format depends on the date and time settings specified by the client computer, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

Syntax

```
atom$async_alert_add_date_time [<flags>]<timestamp>[<timezone>]
```

The format of the argument to this atom is determined by the argument length, as follows:

4 bytes [<flags>]

5 bytes [<flags>] <timestamp>

6 bytes [<flags>] <timestamp> [<timezone>]

[<flags>] Specifies the display format of the date and time. This is a 1-byte argument whose bits are specified as follows:

Bit 0 The time will appear. If this bit is not specified, bits 1, 2, and 3 are ignored.

Bit 1 Seconds will be included in the time display, e.g., 08:30:01.

Bit 2 The time zone will be included in the time display, e.g., 08:30:01 EDT.

Bit 3 The time will appear before the date, e.g., 08:30:01 05-30-96.

Bit 4 The date will appear. If this bit is not specified, bits 5, 6, and 7 are ignored.

Bit 5 Displays the date in abbreviated date format, e.g., Thur, May 30, 1996. (Macintosh only)

Bit 6 Displays the date in long date format, e.g., Thursday, May 30, 1996.

Bit 7 Not used.

<timestamp> Specifies that the host time stamp will be used as the date/time string for the current object. The host time stamp is in Stratus time format.

[<timezone>] Host-specified time zone will appear. Otherwise, the client-specified time zone will be used.

Return Value

Unchanged.

Example

The following is an example of how to attach a localized date/time string to the end of the alert message being built:

```
atom$uni_start_stream
.
.
.
atom$async_alert_start "The following error occurred"
↔ atom$async_alert_add_date_time <508741200>
atom$async_alert_add_text "This is not a known user."
atom$async_alert_end
.
.
.
atom$uni_end_stream
```

atom\$async_alert_add_text

32 (\$20)

Description

atom\$async_alert_add_text inserts text into the alert message being built. If a text argument is provided with **atom\$async_alert_start**, the text argument for this atom is concatenated to the end of the alert message. This atom is sent by the host.

Syntax

```
atom$async_alert_add_text [<text>]
```

[<text>] Text inserted into the alert message being built.

Return Value

Unchanged.

Example

The following is an example of how to attach text to the end of the alert message being built:

```
atom$uni_start_stream
.
.
.
atom$async_alert_start "The following error occurred"
atom$async_alert_add_date_time <508741200>
↪ atom$async_alert_add_text "This is not a known user."
atom$async_alert_end
.
.
.
atom$uni_end_stream
```

atom\$async_alert_end

34 (\$22)

Description

atom\$async_alert_end specifies the end of an alert message definition. If a text argument is provided with this atom, the text is attached to the end of the alert message being built. This atom is sent by the host.

Syntax

```
atom$async_alert_end [<text>]
```

[<text>] The text of the alert message definition.

Return Value

Unchanged.

Example

The following is an example of how to specify the end of an alert message definition:

```
atom$uni_start_stream
.
.
.
atom$async_alert_start "The following error occurred"
atom$async_alert_add_date_time <508741200>
atom$async_alert_add_text "This is not a known user."
⇨ atom$async_alert_end
.
.
.
atom$uni_end_stream
```

atom\$async_alert_start

31 (\$1F)

Description

atom\$async_alert_start specifies the start of an alert message definition. If a text argument is provided with this atom, the text is attached to the beginning of the alert message being built. This atom is sent by the host.

Syntax

```
atom$async_alert_start [<text>]
```

[<text>] Text attached to the beginning of the message within the alert message being built.

Return Value

Unchanged.

Example

The following is an example of how to specify the beginning of an alert message definition:

```
atom$uni_start_stream
.
.
.
⇒ atom$async_alert_start "The following error occurred"
atom$async_alert_add_date_time <508741200>
atom$async_alert_add_text "This is not a known user."
atom$async_alert_end
.
.
.
atom$uni_end_stream
```

atom\$async_invoke_language_pref

36 (\$24)

Description

atom\$async_invoke_language_pref invokes the language preferences dialog box. Language preferences allow the member to specify a language from a list box of up to eight languages.

Syntax

```
atom$async_invoke_language_pref
```

Return Value

Unchanged.

Example

The following is an example of how to invoke the language preferences dialog box:

```
atom$uni_start_stream  
atom$uni_set_context_relative <1>  
⇨ atom$async_invoke_language_pref  
atom$man_end_context  
atom$uni_end_stream
```


atom\$async_invoke_timezone_pref

35 (\$23)

Description

atom\$async_invoke_timezone_pref invokes the time zone preferences dialog box. Time zone preferences allow members to specify the local time zone from a list of time zones. This atom is not currently intended for use with Macintosh clients of the online service.

Syntax

```
atom$async_invoke_timezone_pref
```

Return Value

Unchanged.

Example

The following is an example of how to invoke the time zone preferences dialog box:

```
atom$uni_start_stream  
atom$uni_set_context_relative <1>  
↔ atom$async_invoke_timezone_pref  
atom$man_end_context  
atom$uni_end_stream
```

atom\$lm_add_entry_date_time

36 (\$24)

Description

atom\$lm_add_entry_date_time converts a host time stamp to a localized date/time string and adds the string to the list entry being built. This atom is sent by the host. The date and time format also depends on the date and time settings specified by the client computer, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

Syntax

```
atom$lm_add_entry_date_time [<flags>] <timestamp> [<timezone>]
```

The format of the argument to this atom is determined by the argument length, as follows:

4 bytes [<flags>]

5 bytes [<flags>] <timestamp>

6 bytes [<flags>] <timestamp> [<timezone>]

[<flags>] Specifies the display format of the date and time. This is a 1-byte argument whose bits are specified as follows:

Bit 0 The time will appear. If this bit is not specified, bits 1, 2, and 3 are ignored.

Bit 1 Seconds will be included in the time display, e.g., 08:30:01.

Bit 2 The time zone will be included in the time display, e.g., 08:30:01 EDT.

Bit 3 The time will appear before the date, e.g., 08:30:01 05-30-96.

Bit 4 The date will appear. If this bit is not specified, bits 5, 6, and 7 are ignored.

Bit 5	Displays the date in abbreviated date format, e.g., Thur, May 30, 1996. (Macintosh only)
Bit 6	Displays the date in long date format, e.g., Thursday, May 30, 1996.
Bit 7	Not used.
<timestamp>	Specifies that the host time stamp will be used as the date/time string for the current object. The host time stamp is in Stratus time format.
[<timezone>]	Host-specified time zone will appear. Otherwise, the client-specified time zone will be used.

Return Value

Unchanged.

Example

The following is an example of how to attach a localized date/time string to the "New Mail" list entry being built:

```
atom$uni_start_stream
.
.
.
atom$lm_start_list_entry
↔ atom$lm_add_entry_date_time <270452900>
atom$lm_add_entry_tab
atom$lm_add_entry_text "Akoontz"
atom$lm_add_entry_tab
atom$lm_add_entry_text "No Subject"
atom$lm_end_list_entry
.
.
.
atom$uni_end_stream
```

atom\$lm_add_entry_tab

35 (\$23)

Description

atom\$lm_add_entry_tab attaches a tab to the list entry being built. This atom is sent by the host and is used within an **atom\$lm_start_list_entry** / **atom\$lm_end_list_entry** stream.

Syntax

```
atom$lm_add_entry_tab
```

Return Value

Unchanged.

Example

The following is an example of how to attach a tab to the "New Mail" list entry being built:

```
atom$uni_start_stream
.
.
.
atom$lm_start_list_entry
atom$lm_add_date_time <270452900>
⇨ atom$lm_add_entry_tab
atom$lm_add_entry_text "Akoontz"
⇨ atom$lm_add_entry_tab
atom$lm_add_entry_text "No Subject"
atom$lm_end_list_entry
.
.
.
atom$uni_end_stream
```

atom\$lm_add_entry_text

34 (\$22)

Description

atom\$lm_add_entry_text attaches text to the end of the list entry being built. This atom is sent by the host and is used within an **atom\$lm_start_list_entry** / **atom\$lm_end_list_entry** stream.

Syntax

```
atom$lm_add_entry_text <text>
```

<text> The text of the list entry being built.

Return Value

Unchanged.

Example

The following is an example of text attached to the "New Mail" list entry being built:

```
atom$uni_start_stream
.
.
.
atom$lm_start_list_entry
atom$lm_add_date_time <270452900>
atom$lm_add_entry_tab
↔ atom$lm_add_entry_text "Akoontz"
atom$lm_add_entry_tab
↔ atom$lm_add_entry_text "No Subject"
atom$lm_end_list_entry
.
.
.
atom$uni_end_stream
```

atom\$man_add_date_time

84 (\$54)

Description

atom\$man_add_date_time converts a host time stamp to a localized date/time string and adds the string to the current object. This atom is sent by the host. The date and time format also depends on the date and time settings specified by the client computer, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

Syntax

```
atom$man_add_date_time [<flags>] <timestamp> [<timezone>]
```

The format of the argument to this atom is determined by the argument length, as follows:

4 bytes [<flags>]

5 bytes [<flags>] <timestamp>

6 bytes [<flags>] <timestamp> [<timezone>]

[<flags>] Specifies the display format of the date and time. This is a 1-byte argument whose bits are specified as follows:

Bit 0 The time will appear. If this bit is not specified, bits 1, 2, and 3 are ignored.

Bit 1 Seconds will be included in the time display, e.g., 08:30:01.

Bit 2 The time zone will be included in the time display, e.g., 08:30:01 EDT.

Bit 3 The time will appear before the date, e.g., 08:30:01 05-30-96.

Bit 4 The date will appear. If this bit is not specified, bits 5, 6, and 7 are ignored.

Bit 5	Displays the date in abbreviated date format, e.g., Thur, May 30, 1996. (Macintosh only)
Bit 6	Displays the date in long date format, e.g., Thursday, May 30, 1996.
Bit 7	Not used.
<timestamp>	Specifies that the host time stamp will be used as the date/time string for the current object. The host time stamp is in Stratus time format.
[<timezone>]	Host-specified time zone will appear. Otherwise, the client-specified time zone will be used.

Return Value

Unchanged.

Example

The following is an example of how to attach a localized date/time string to the end of the current object:

```
atom$man_add_date_time <270452900>
```

atom\$man_add_title_date_time

88 (\$58)

Description

atom\$man_add_title_date_time converts a host time stamp to a localized date/time string and adds the string to the end of the title being built. This atom is sent by the host. The date and time format also depends on the date and time settings specified by the client computer, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

Syntax

```
atom$man_add_title_date_time [<flags>] <timestamp> [<timezone>]
```

The format of the argument to this atom is determined by the argument length, as follows:

4 bytes [<flags>]

5 bytes [<flags>] <timestamp>

6 bytes [<flags>] <timestamp> [<timezone>]

[<flags>] Specifies the display format of the date and time. This is a 1-byte argument whose bits are specified as follows:

Bit 0 The time will appear. If this bit is not specified, bits 1, 2, and 3 are ignored.

Bit 1 Seconds will be included in the time display, e.g., 08:30:01.

Bit 2 The time zone will be included in the time display, e.g., 08:30:01 EDT.

Bit 3 The time will appear before the date, e.g., 08:30:01 05-30-96.

Bit 4 The date will appear. If this bit is not specified, bits 5, 6, and 7 are ignored.

Bit 5	Displays the date in abbreviated date format, e.g., Thur, May 30, 1996. (Macintosh only)
Bit 6	Displays the date in long date format, e.g., Thursday, May 30, 1996.
Bit 7	Not used.
<timestamp>	Specifies that the host time stamp will be used as the date/time string for the current object. The host time stamp is in Stratus time format.
[<timezone>]	Host-specified time zone will appear. Otherwise, the client-specified time zone will be used.

Return Value

Unchanged.

Example

The following is an example of how to attach a localized date/time string to the end of the title being built:

```
atom$man_add_title_date_time
```

atom\$man_add_title_tab

87 (\$57)

Description

atom\$man_add_title_tab attaches a tab to the end of a title being built. This atom is sent by the host.

Syntax

```
atom$man_add_title_tab
```

Return Value

Unchanged.

Example

The following is an example of how to attach a tab to the title being built:

```
atom$man_add_title_tab
```

atom\$man_add_title_text

86 (\$56)

Description

atom\$man_add_title_text attaches text to the end of the title being built. This atom is sent by the host.

Syntax

```
atom$man_add_title_text <text>
```

<text> Text to be attached to the end of the title.

Return Value

Unchanged.

Example

The following is an example of how to attach text to the title being built:

```
atom$uni_start_stream
.
.
.
atom$man_start_title " "
↪ atom$man_add_title_text "Sonics Sweep Rockets"
atom$man_end_title
.
.
.
atom$uni_end_stream
```

atom\$man_end_title

89 (\$59)

Description

atom\$man_end_title specifies the end of a title definition. If a text argument is provided, it is attached to the end of the title. This atom is sent by the host.

Syntax

```
atom$man_end_title [<text>]
```

[<text>] Text to be attached to the end of the title.

Return Value

Unchanged.

Example

The following is an example of how to specify the end of a title definition:

```
atom$uni_start_stream
.
.
.
atom$man_start_title " "
atom$man_add_title_text "Sonics Sweep Rockets"
↪ atom$man_end_title
.
.
.
atom$uni_end_stream
```

atom\$man_start_title

85 (\$55)

Description

atom\$man_start_title specifies the start of a title definition. If a text argument is provided with this atom, the text is attached to the beginning of the title being built. This atom is sent by the host.

Syntax

```
atom$man_start_title [<text>]
```

[<text>] Text to be attached to the beginning of the title.

Return Value

Unchanged.

Example

The following is an example of how to specify the beginning of a title definition:

```
atom$uni_start_stream
.
.
.
↪ atom$man_start_title " "
atom$man_add_title_text "Sonics Sweep Rockets"
atom$man_end_title
.
.
.
atom$uni_end_stream
```

atom\$mat_bool_encode_unicode

134 (\$86)

Description

atom\$mat_bool_encode_unicode specifies that Unicode encoding will be used as the encode type when the contents of the associated text field are sent to the host. This atom is sent by the host.

Syntax

```
atom$mat_bool_encode_unicode (<yes>|<no>)
```

<yes> Unicode encoding will be used.

<no> Unicode encoding will not be used. (Default)

Return Value

Unchanged.

Example

The following is an example of how to specify that Unicode encoding will be used as the encode type:

```
atom$mat_bool_encode_unicode <yes>
```

atom\$mat_bool_first_script

106 (\$6A)

Description

atom\$mat_bool_first_script specifies whether the associated text field or list box is limited to characters from the script language type of the first character entered. This atom is sent by the host.

Syntax

```
atom$mat_bool_first_script (<yes>|<no>)
```

<yes> The associated text field or list box is limited to characters from the script language type of the first character entered.

<no> The associated text field or list box is not limited to characters from the script language type of the first character entered. (Default)

Return Value

Unchanged.

Example

The following is an example of how to specify that the associated text field or list box be limited to characters from the script language type of the first character entered:

```
atom$mat_bool_first_script <yes>
```

atom\$mat_bool_language_popup

117 (\$75)

Description

atom\$mat_bool_language_popup populates the language preferences list box within the member preferences/language preferences area of the online service. This atom is sent by the host.

Syntax

```
atom$mat_bool_language_popup (<yes>|<no>)
```

<yes>	Populates the language preferences list box.
<no>	Does not populate the language preferences list box. (Default)

Return Value

Unchanged.

Example

The following is an example of how to populate the language preferences list box:

```
atom$mat_start_object <dss_list>  
atom$mat_orientation <hcc>  
↔ atom$mat_bool_language_popup <yes>  
atom$mat_title_position 01x  
atom$mat_bool_invisible <yes>  
atom$mat_start_object
```


atom\$mat_field_script

104 (\$68)

Description

atom\$mat_field_script specifies the allowable script language type (and associated ASCII type) for the associated input field. This atom is sent by the host.

Syntax

```
atom$mat_field_script <script>
```

<script> Allowable script for the associated input field. Valid values are as follows:

- | | |
|-----|-----------------------|
| 127 | ScriptASCII (Default) |
| 128 | ScriptAnyScript |
| 0 | ScriptLatin1 |
| 1 | ScriptJapanese |
| 2 | ScriptTradChinese |
| 3 | ScriptKorean |
| 4 | ScriptArabic |
| 5 | ScriptHebrew |
| 6 | ScriptGreek |
| 7 | ScriptCyrillic |
| 8 | ScriptDevangari |
| 9 | ScriptGurmaukhi |
| 10 | ScriptGujarati |
| 12 | ScriptOriya |

| | |
|----|--------------------|
| 13 | ScriptBengali |
| 14 | ScriptTamil |
| 15 | ScriptTelugu |
| 16 | ScriptKannada |
| 17 | ScriptMalayalam |
| 18 | ScriptSinhalese |
| 19 | ScriptBurmese |
| 20 | ScriptKhmer |
| 21 | ScriptThai |
| 22 | ScriptLaotian |
| 23 | ScriptGeorgian |
| 24 | ScriptArmenian |
| 25 | ScriptSimpChinese |
| 26 | ScriptTibetan |
| 27 | ScriptMongolian |
| 28 | ScriptGeez |
| 29 | ScriptEastEurRoman |
| 30 | ScriptVietnamese |
| 31 | ScriptExtArabic |

Return Value

Unchanged.

Example

The following is an example of how to specify the allowable script language type for the Keyword input field of the online service:

```
atom$man_append_data "Enter word(s):"  
atom$man_start_sibling <edit_view>  
↔ atom$mat_field_script 00x
```

atom\$mip_message_date

16 (\$10)

Description

atom\$mip_message_date converts a host time stamp to a localized date/time string and attaches the string to the message being built. This atom is sent by the host and should be used in place of **atom\$mip_message_datestr** for international clients. The date and time format also depends on the date and time settings specified by the client computer, i.e., Control Panels/Date & Time (Macintosh) and Control Panel/Regional Settings (Windows).

Syntax

```
atom$mip_message_date [<flags>] <timestamp> [<timezone>]
```

The format of the argument to this atom is determined by the argument length, as follows:

4 bytes [<flags>]

5 bytes [<flags>] <timestamp>

6 bytes [<flags>] <timestamp> [<timezone>]

[<flags>] Specifies the display format of the date and time. This is a 1-byte argument whose bits are specified as follows:

Bit 0 The time will appear. If this bit is not specified, bits 1, 2, and 3 are ignored.

Bit 1 Seconds will be included in the time display, e.g., 08:30:01.

Bit 2 The time zone will be included in the time display, e.g., 08:30:01 EDT.

Bit 3 The time will appear before the date, e.g., 08:30:01 05-30-96.

Bit 4 The date will appear. If this bit is not specified, bits 5, 6, and 7 are ignored.

| | |
|--------------|--|
| Bit 5 | Displays the date in abbreviated date format, e.g., Thur, May 30, 1996. (Macintosh only) |
| Bit 6 | Displays the date in long date format, e.g., Thursday, May 30, 1996. |
| Bit 7 | Not used. |
| <timestamp> | Specifies that the host time stamp will be used as the date/time string for the current object. The host time stamp is in Stratus time format. |
| [<timezone>] | Host-specified time zone will appear. Otherwise, the client-specified time zone will be used. |

Return Value

Unchanged.

Example

The following is an example of how to attach a localized date/time string to the message being built:

```
atom$mip_message_date <943587328>
```

atom\$uni_end_typed_data

24 (\$18)

Description

atom\$uni_end_typed_data terminates the current data type specified by **atom\$uni_start_typed_data** and resets the data type to ASCII. This atom can be sent by the host or the client. **atom\$uni_start_typed_data / atom\$uni_end_typed_data** pairs cannot be nested.

Syntax

```
atom$uni_end_typed_data
```

Return Value

Unchanged.

Example

The following is an example of how to terminate the current data type in effect:

```
atom$uni_start_typed_data <$0104>
.
.
.
↵ atom$uni_end_typed_data
```


Example

The following is an example of how to specify the data type of the text argument of the subsequent atom:

```
atom$uni_start_typed_data <$0000>
.
.
.
↪ atom$uni_next_atom_typed <$0104>
atom$async_play_sound "A bientôt"
.
.
atom$uni_end_typed_data
```


atom\$uni_start_typed_data

23 (\$17)

Description

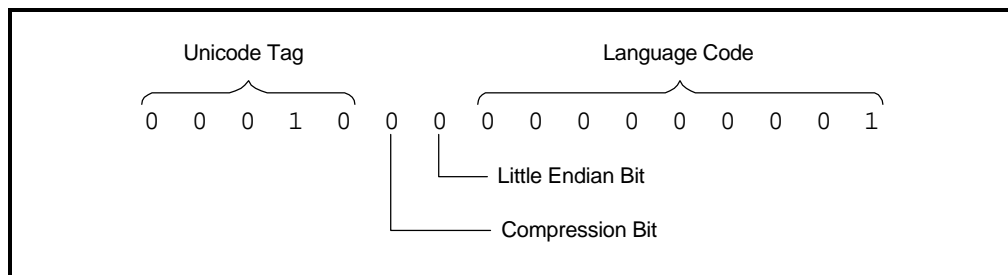
atom\$uni_start_typed_data specifies the data type for the text arguments of all subsequent atoms in the stream. **atom\$uni_start_typed_data / atom\$uni_end_typed_data** pairs cannot be nested.

Syntax

```
atom$uni_start_typed_data <data_type>
```

| | | | |
|-------------|--|-----------------|-----------------|
| <data_type> | The data type identifier for the text argument of all subsequent atoms. Valid values are as follows: | | |
| | 0 | (\$0000) | ASCII (Default) |
| | 260 | (\$0104) | Latin1 |
| | 4096-6143 | (\$1000-\$17FF) | Unicode |

The data type identifier is a two-byte field that defines how the data is interpreted and how it affects the target field. The example below specifies that the data type is Unicode, compression is off, big endian format is used, and the language is French.



Return Value

Unchanged.

Example

The following is an example of how to specify the data type of the text argument of all subsequent atoms:

```
⇒ atom$uni_start_typed_data <$0104>
.
.
.
atom$uni_end_typed_data
```