

Chapter III-16 — Text Encodings

```
TextBox "α"
```

Also, Symbol font is no longer treated specially. Consequently this:

```
TextBox "\F'Symbol'a"
```

displays a "small letter a", not a "small Greek letter alpha". This is a departure from Igor6 and creates compatibility issues.

You can copy Symbol font characters to the clipboard as Unicode using the **Symbol Font Characters** table. You can insert Unicode characters by choosing Edit→Characters for commonly-used characters such as Greek letters and math symbols or Edit→Special Characters for other characters. In the Add Annotation dialog and in the Axis Label tab of the Modify Axis dialog, you can click Special and choose a character from the Character submenu.

Symbol Font Backward Compatibility

The use of Unicode greatly simplifies the display of Greek letters and other special characters. But it creates an incompatibility with Igor6. Without special handling, when Igor loads an Igor6 file, Symbol font text meant to display Greek letters would display Roman letters instead. This section explains how Igor deals with this issue to provide backward compatibility.

When loading Igor6-compatible procedure files, which use system text encoding, including the experiment recreation procedures that execute when you open an Igor6-compatible experiment file, Igor attempts to convert incoming Symbol font characters to Unicode. For compatibility with Igor6, the reverse is done on writing procedure files, including experiment recreation procedures, using system text encoding. The conversion is also done for Igor6-compatible formatted notebook files which use system text encoding. This conversion is done using a heuristic that involves scanning for certain patterns and consequently is not, and can not be, perfect.

There may be cases where this attempt at backward compatibility creates problems, for example, if you don't care about Igor6 compatibility. If you want to turn Symbol font compatibility off, you can execute

```
SetIgorOption EnableSymbolToUnicode=0
```

and reload the experiment.

If you encounter Symbol font problems and the information below does not provide the solution, please let us know, and provide the original Igor6 text that caused the problem.

Symbol Font Backward Compatibility Limitations

An important limitation is that, in the original Igor6 experiment, Symbol font specifications in annotations must not include anything other than the Symbol characters. No escape sequences, such as font size specifications, are allowed. For example, Igor will not handle this Igor6 text:

```
TextBox/C/N=text0/F=0/A=MC "\Z18A\Symbol'\Bq\F]0"
```

because the subscript escape code, "\B", is inside the Symbol font escape sequence and prevents Igor from recognizing this as a Symbol font sequence. As a result, you will see a "q" character instead of the intended small Greek letter theta character.

To enable Igor to recognize this as Symbol text, move the subscript escape code outside the Symbol font escape sequence, like this:

```
TextBox/C/N=text0/F=0/A=MC "\Z18A\B\Symbol'q\F]0"
```

When creating graphics for compatibility with Igor6 or for EPS export, use Unicode characters, like this:

```
TextBox/C/N=text0/F=0/A=MC "\Z18A\B\Symbol'θ\F]0"
```

When Igor writes this out to a procedure file, it will convert the Symbol font sequence into an Igor6-compatible sequence, by replacing the Unicode theta character with "q".