GUI Date Input With the BBj INPUTD Control

By Jim Douglas

For many years, BBX® and BBj® have provided all of the language constructs necessary to control the output format of dates. The DATE() function formats a date based on any preferred output mask. The STBL("!DATE") global string defines the default date mask and the localized text for month and day names. The JUL() function converts a year, month, and day to a unique integer that corresponds to a particular date. In addition, BBj simplifies internationalization even more by automatically updating the STBL("!DATE") global string when the developer changes the locale. For example, setting STBL("!LOCALE") to "de_DE" sets the default date mask to DD.MM.YY and changes the month and day names to German.

Historically, date input was not as simple as date output. An application can prompt for a date using an INPUTE control with a mask such as "00/00/00" or "00/00/0000", but then it must interpret the date by assuming a particular format (usually MM/DD/YY or DD/MM/YY). This does not allow the user to type a date, for example, in the format "7/4/2004". The user instead must enter "070404", explicitly including the leading zeroes and allowing the INPUTE control to fill in the slashes. This adds unnecessary work and complexity for users.

Available in BBj 4.0, the INPUTD control uses the same date mask as the DATE() function, but it also allows the user to input a date in any logical format (based on the relative positions of the year, month, and day in the mask). For example, a user in Finland using a date mask of "%D.%M.%Y" can enter January 1, 2004 in the following ways:

"1/1/4"
"1/1/2004"
"01.01.2004"
"1-1-4"
"010104"
"01012004"

Moreover, if the current month is January 2004, then the user can enter "1", "1/1", or "0101"; the year defaults to 2004. After the user enters a date into an INPUTD control, the application can retrieve the text of the formatted date (e.g. "1/1/04") or the value of the Julian day number (2453006). With BBj 4.0, the complexity of interpreting the user's date input is transferred from the application programmer to the INPUTD control.

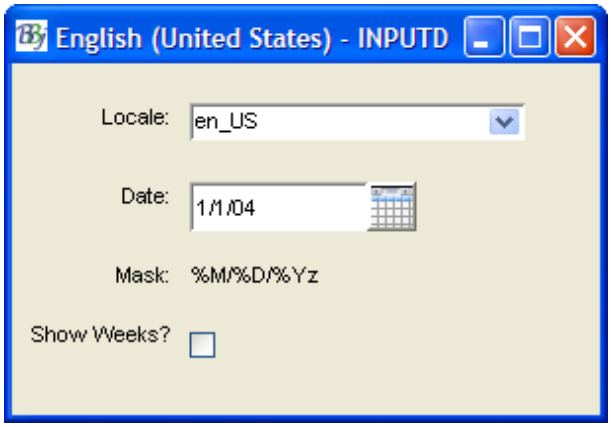

Figure 1. Date Format Changes.

Figure 1 demonstrates how the date format changes with different locales and different date masks.

This sample program uses a resource file, but a user can also create INPUTD controls using ***object syntax*** or the ***'INPUTD' mnemonic***.

Four-digit years are easier to understand, but many common date formats specify only 2-digit years and this often leads to ambiguity. For instance, is 7/4/50 in the past or the future? Because there is no standard way to resolve this ambiguity, BBj allows developers to customize the

continued...

YY-to-YYYY conversion rules using the STBL("!YYDATE") global string. The default rule is that 2-digit years fall in a range from 50 years before the current year to 49 years after the current year. Based on this rule, the INPUTD control assumes that a 2-digit year entered during 2003 falls between 1953 and 2052. See ***STBL("!YYDATE")*** for more information.

Date input controls usually provide a popup calendar so the user can click on a date and have it appear in the control. In this sample program, a user can invoke the popup calendar by pressing CTRL+P or the F2 key, or by clicking  , or by right-clicking in the date control.

In the example below, the popup calendar's format follows the rules for German (Germany):

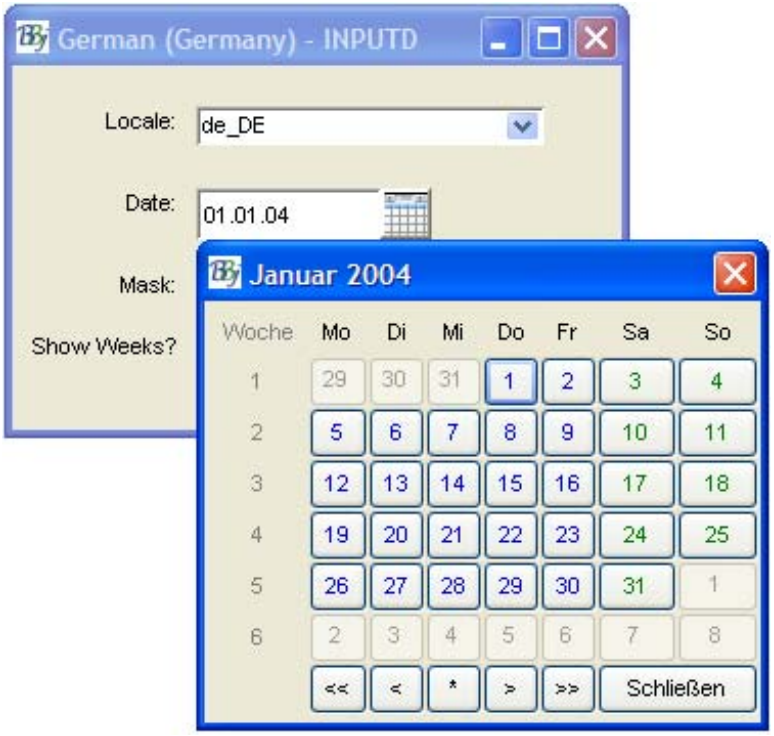

Figure 2. Popup Calendar.

The popup calendar appears in the language of the selected locale (German/Germany) and is formatted according to whether a week starts with Sunday (North America) or Monday (Europe).

continued...

Users can embed INPUTD controls in data-aware or standard grids, and data-aware grids can display dates that are stored in OEM date formats. The following ***example*** shows various OEM date formats embedded in a data-aware grid:

Figure 3. Data Aware Grid Showing OEM Date Formats.

In BBj 4.0, the JUL() function has also been enhanced to convert a human-readable date string to a Julian date. The date string is parsed according to the same rules used by the INPUTD control. For example:

```
0010 ENTER:
0020 INPUT "Enter a date: ",DATE$
0030 IF DATE$="" THEN STOP
0040 PRINT "You entered: ",DATE(JUL(DATE$,ERR=*NEXT)); GOTO ENTER
0050 PRINT 'RB',"Invalid date: ",DATE$; GOTO ENTER
0060 END
>RUN
Enter a date: 1231
You entered: 12/31/03
Enter a date: 123104
You entered: 12/31/04
Enter a date: 0229
Invalid date: 0229
Enter a date: 022904
You entered: 02/29/04
```

For more information about the INPUTD control and related features, see:

Date Input
Data Aware Grid Channels - BBj
JUL() – Get Julian Date - BBj

With the new INPUTD graphical control and enhanced JUL() functionality, BBj 4.0 now includes all of the features necessary for flexible, internationalized, CHUI or GUI date input.