

# CellConv

A cell's validation and conversion settings.

## Usage

```
Set_Property(OLECtrlEntID, "OLE.CellConv[field; record]", Array)
```

## Values

Array has the following structure:

Pos	Name	Type	Description	Default
<1>	Validation Pattern	Text	The cell's validation pattern	"None"
<2>	Conversion Pattern	Text	The cell's output conversion pattern	"None"
<3>	Validation Message	Text	The cell's message to display when data is invalid	""
<4>	Show Message Flag	Boolean	Determines if the cell should display the validation message or simply pass it on to the <a href="#">OnInvalidData</a> event	1
<5>	Highlight Text on Cancel Flag	Boolean	Determines if the invalid text should be highlighted instead of just leaving the cursor where it is	1
<6>	Disable Validation	Boolean	Determines if the table's automated validation is disabled	0

## Indices

Index	Description
field	Index to an existing field
record	Index to an existing record

## Remarks

The CellConv property attempts to simplify the process of validating and converting data in an Edit Table. Manual validation and conversion requires a lot of event processing, but adding your settings here reduces the amount of work required.

***If RowsToRecord is not zero, then a validation/conversion setting for any cell is applied to the entire column of that cell.***

Here is a breakdown of this property's fields:

### Validation Pattern

The validation pattern can be any OI pattern acceptable to the built-in OI function IConv. If you wish to remove a pattern, set this field to "None". Also, all user defined patterns must be written in ALL CAPS and encased in square brackets, e.g., "[PHONE\_FORMAT]"

***Some user defined conversions show message boxes internally. The only way to suppress this message boxes is to write your own version that does not display message boxes.***

### Conversion Pattern

The conversion pattern **must be a custom conversion**. Simply passing built-in conversions does not work due to technical limitations. The SRP EditTable Controls calls the FMT routine via RevCAPI, and for reasons unknown to us, built-in conversions do not work when called from outside BASIC+. However, custom conversions--conversion within square brackets--work just fine. As with Validation Patterns, these must be written in ALL CAPS and encased in square brackets. We recommend creating a custom converter that is a simple wrapper around OConv. For example:

```

Compile subroutine Cell_Conv(Conv, Ans, Branch, ReturnData)

If Conv EQ "OCONV" then
  If Branch[1, 1] EQ "D" then
    If Num(Ans) then
      // The data is already in internal format, so just OConv it
      ReturnData = OConv(Ans, Branch)
    end else
      // The data is an some date format, so IConv it first, then OConv it
      ReturnData = OConv(IConv(Ans, "D"), Branch)
    end
  end else
    // For all non-date types, IConv first (just in case), then OConv it
    ReturnData = OConv(IConv(Ans, Branch), Branch)
  end
end else
  // IConv as usual
  ReturnData = IConv(Ans, Branch)
end

Return

```

Of course, you can use any custom converter you desire, just so long as you are aware that this is the best way to get built-in conversions working automatically for you in the SRP EditTable Control.

## Validation Message

The validation message is the text to be presented to the end user when the data is recognized to be invalid. If there is no validation pattern, then this field is ignored. Furthermore, if this field is empty, no message will display. This text is always passed via the Message parameter of the [OnInvalidData](#) function for convenience, but you can have the Edit Table show the message for you (see below).

The validation message string accepts one placeholder string called @DATA, which will be replaced with the invalid data so long as you have specified that the edit table show its own message. For example, "@DATA is not a valid phone number" will become "ASDF is not a valid phone number".

***The @DATA string is only replaced when the edit table shows its own message. If you opt to capture the [OnInvalidData](#) event instead, then the @DATA string is not replaced.***

## Show Message Flag

This flag is where you choose how to handle invalid data. If you set this to 1, which is the default, then the edit table will show a standard message box with the message you supplied above. If no message was supplied or if you set this field to 0, then the [OnInvalidData](#) event fires instead.

## Highlight Text on Cancel Flag

This flag allows you to specify the behavior of the cursor when data is invalid. If you set this to 1, which is the default, then the entire cell is selected. If you set this to 0, then the cursor remains in its previous position. Note that this flag is ignored if the Show Message flag is turned off. If the [OnInvalidData](#) event is used to display error messages then use the [Cancel](#) property to control the behavior of the cursor.

## Disable Validation

By default, the table will always attempt to validate the cell's data against the Validation Pattern. If, however, you merely want to store the validation in this property but perform the validation check yourself, you can set this flag to 1 to disable the automated validation. In this case, you would want to use the [BeforeUpdate](#) event to perform the validation manually.

## Example

```

// Validate by date (note: OnInvalidData will fire on invalid data)
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D")

// Validate by date and convert using custom conversion (note: OnInvalidData will fire on invalid data)
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D":@FM:"[CELL_CONV,D2-]")

// Validate by date and show a message box instead of firing OnInvalidData
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D":@FM:"[CELL_CONV,D2-]":@FM:"@DATA" is not a
valid date.')

// Validate by date, set a custom message, still fire OnInvalidData instead
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D":@FM:"[CELL_CONV,D2-]":@FM:"@DATA" is not a
valid date.':@FM:0)

// Validate by date and show a message box, but don't select the invalid text
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D":@FM:"[CELL_CONV,D2-]":@FM:"@DATA" is not a
valid date.':@FM:@FM:0)

// Validate by date, but do the validation manually
Set_Property(@Window:".OLE_EDITTABLE", "OLE.CellConv[1; 1]", "D":@FM:"[CELL_CONV,D2-]":@FM:@FM:@FM:@FM:1)

```