
Important Changes

The following changes have been made to Setup Factory 6.0 since the User's Guide was printed:

Expressions

1. You Must Escape Quotes and Backslashes in Expressions
2. You Don't Need to Quote Variables
3. You Can't Pass Expressions in Variables

Built-in Variables

1. Built-in Variable %PreventNextPage% Added

Expressions

1. You Must Escape Quotes and Backslashes in Expressions

Internal quotes and backslashes need to be escaped when they're used in literal strings inside an expression.

"Escaping" a character means replacing it with an *escape sequence* that represents that character. An escape sequence is a special series of characters that can be used to represent a single character. Escape sequences are usually used for characters that either can't be typed directly into a string, or can't be used on their own without breaking syntax rules.

In Setup Factory, the escape sequence for a quote is \" and the escape sequence for a backslash is \\.

Why is this necessary? In Setup Factory expressions, a quotation mark (") is used to indicate the beginning or end of a single value. In order to include a quotation mark inside a value, you need to use the escape sequence that represents it; otherwise, Setup Factory will see the quotation mark as the end of the value. So, for example, a string like:

She said "Goodbye" before she left

...would need to be written as "She said \"Goodbye\" before she left" when used in an expression. If you wrote it as "She said "Goodbye" before she left", Setup Factory would see the quotation mark before the word Goodbye as marking the end of the value "She said ".

Since a single backslash is now used as an escape character, marking the beginning of an escape sequence, we need a way to include backslash characters in expressions. This is done by using the \\ escape sequence wherever you want a single backslash to appear. For example, the path C:\temp\ now needs to be written as C:\\temp\\ when used in an expression.

IMPORTANT



You only have to escape quotation marks and backslashes in the values that you type directly into expressions. You don't have to escape them anywhere else. For example, strings that you store in variables don't need to be escaped—even if the variables are used in expressions.

2. You Don't Need to Quote Variables

You no longer have to put quotation marks around variables in expressions, even if the values stored in the variables contain spaces. You can still put quotes around the variables if you want...but you no longer need to.

Previously, variables were expanded *before* an expression was parsed into values and operators; now, variables are only expanded *after* an expression has been parsed into values and operators. Essentially, variables now always act as they would if you placed quotation marks around them.

3. You Can't Pass Expressions in Variables

The tip on page 228 of the User's Guide describes storing an expression in a variable and then evaluating the contents of that variable in another expression. This is no longer possible.

For example, if you store the string `5 * 2` in a variable named `%a%`, and then use that variable in the expression `%a% + 1`, the result will be `5 * 21`. This is because `"5 * 2"` is interpreted as a string, so the `+` operator performs concatenation, adding the two strings `"5 * 2"` and `"1"` together.

NOTE



The tip on page 228 has been removed from the electronic (PDF) version of the User's Guide.

Built-in Variables

1. Built-in Variable `%PreventNextPage%` Added

This is a special variable that can be set on a screen's Before or After action tabs. If this variable is set to `TRUE`, the installer will not allow the user to proceed to the next screen when the **Next** button is clicked; instead, the user will remain on the current screen. If set to `FALSE`, the installer will allow the user to proceed normally when they click **Next**.

This variable is useful if you want to validate the data entered by the user on a screen. If the data is incorrect, you could display an error message and set `%PreventNextPage%` to `"true"` to make the user go back and enter valid data.

The value of `%PreventNextPage%` is reset to `FALSE` every time a screen is displayed.