

Processing Write Pro documents (faceless)

By Vance Villanueva, Technical Services Engineer, 4D Inc.

Technical Note 19-01

Table of Contents

Table of Contents.....	2
Introduction	3
Overview.....	3
What are expressions?	3
How are expressions used in 4D Write?	4
4D Write Pro containing evaluated expressions	5
Implementation for faceless processing of 4D Write Pro documents	6
Encapsulating the expression	6
EVALUATION_METHOD.....	7
PROCESS_EXPRESSIONS	7
Sample database	9
Printing from the 4D Write/Write Pro document	9
Printing from the 4D Write Pro object from the object field	10
Conclusion	10
Reference	10

Introduction

4D Write Pro's predecessor had the ability to process documents without the presence of the form object by creating an offscreen area using the command **WR New offscreen area**. But in 4D Write Pro, it is an object that can be manipulated with the 4D Write Pro command suite without calling a command to create an "offscreen area". A common feature that is used in 4D Write/Write Pro documents are expressions which can automate data that are contained in a 4D database tables or in a form of a formula. The need to do this process in the backend without a 4D Write Pro form object as a UI is beneficial. This Technical document will demonstrate a process without the presence of a UI on 4D Write Pro documents containing expressions.

Overview

4D Write Pro documents can simply be opened with the 4D Write Pro commands whether there is a form object present or not. What is also beneficial is that 4D Write Pro documents can be manipulated with certain 4D Write pro commands in a faceless/backend manner. An example would be to do batch jobs such as printing and/or sending emails. Data that is contained in 4D database can be automated with using embedded expressions in a 4D Write Pro document. This Technical document focuses on 4D Write Pro documents that contain expressions where they can be processed very easily without a 4D Write Pro form object reference.

What are expressions?

It is a combination of one or more constants, variables, operators, and functions that the programming language interprets and computes to a value. Here are some sample expressions:

Expressions
1 + 1
X + 1
Sin(45)
Current Date (4D)

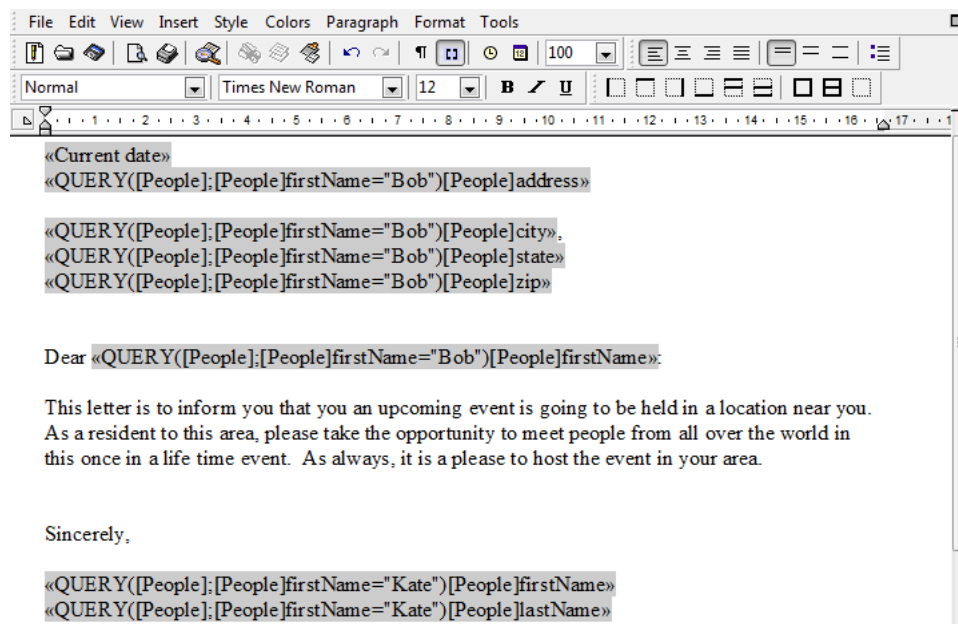
These expressions can be evaluated later at runtime of a program. The evaluation of the expression can be the following:

Expressions evaluated
2
(If X=3) -> 4
0.8509035
01/03/19

The power in expressions is that it can be evaluated later in a 4D Write document. It is like placing a formula in the document which will be displayed later. With this concept, the next sections will explain how it is used in 4D Write and 4D Write Pro.

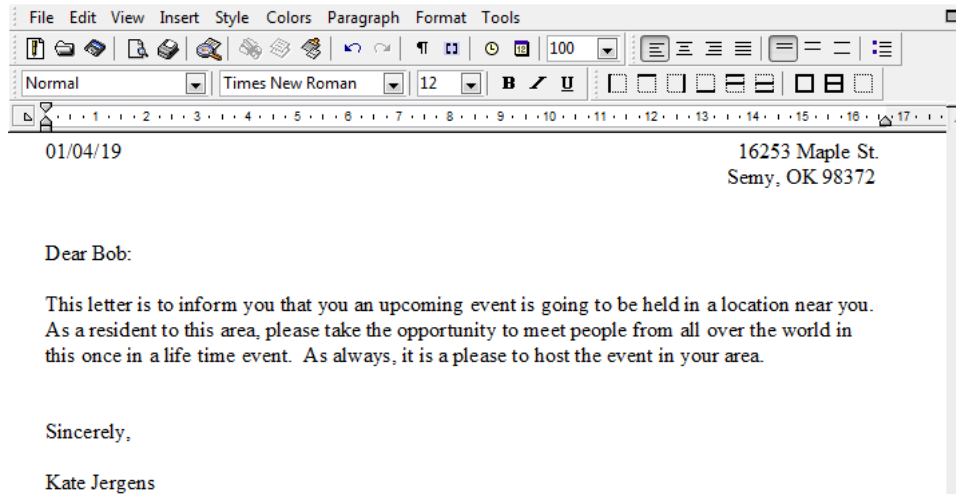
How are expressions used in 4D Write?

4D code expression written as text is placed in the 4D Write document. It is done either by using **WR INSERT EXPRESSION** programmatically or in the menu bar under Insert->4D expressions. An expression that can be used for example is querying a table in a database to place the record result as the following:



E.g. - QUERY([PEOPLE];[People]firstName="Bob") [People]address

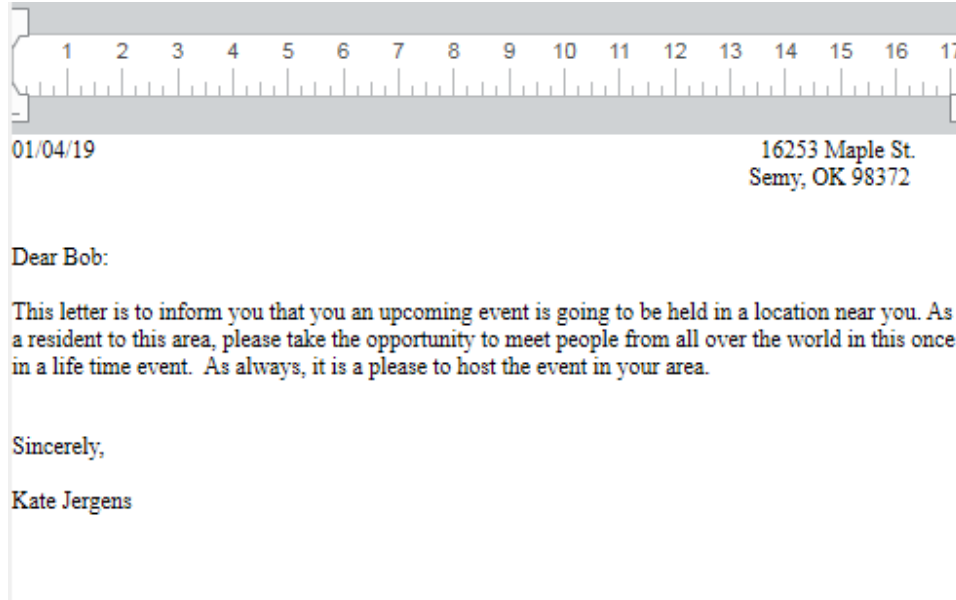
The expressions inserted above are the **Current date** and **Query** commands to make a generic letter in a 4D Write document which are denoted with "<<" and ">>". When the expressions are evaluated, it will look like the following:



This is a simple example of how a 4D Write document can be used with expressions. It is also similar in 4D Write Pro.

4D Write Pro containing evaluated expressions

In v17, 4D Write will not be available in 64-bit. Therefore, documents that are in 4D Write will need to be converted to 4D Write Pro. The same document that is opened in 4D Write Pro with the evaluated expressions show the following:



The evaluated expression is similar to what 4D Write had produced. But keep in mind that there is a process which will be discussed later in this document on how it becomes similar. Given the two examples above, this gives an idea of how the document is processed with the presence of a form object. The expressions are evaluated and displayed in the appropriate manner. Like 4D Write's offscreen area, the goal of this Technical document is

to demonstrate the same result but faceless with a 4D Write Pro document/object. It can be then verified by printing to a PDF through a sample database which is in the later part of the document.

Note: In order to keep the compatibility of the converted 4D Write document to 4D Write Pro that contains expressions, the same database that is either upgraded or not MUST be used. The reason is because 4D Write internally references the table and fields with unique IDs when written in the expression. Using another database with the same table and field names will not function as expected in the document.

Implementation for faceless processing of 4D Write Pro documents

As displayed in the previous section, the sample documents that contain expressions will be evaluated. But how is this done in a non-UI process for 4D Write Pro? This section will explain the process. In a 4D Write Pro document, every character that is placed in the document has a location which represents one single position. But for an expression, the amount of characters used will only represent one single position. There may be situations where an expression in a 4D Write Pro document could possibly not be evaluated properly. As a safeguard, encapsulating the expression with something that 4D Write Pro can evaluate will be used.

Encapsulating the expression

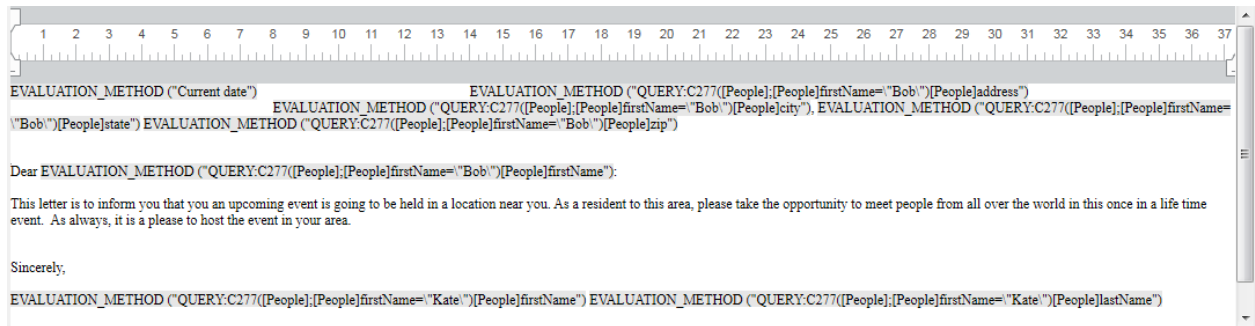
In 4D Write Pro, expressions can also be inserted and recognized in the document. An expression that 4D Write Pro can evaluate is 4D methods. By extracting the expression at the location of the 4D Write Pro document, it will be used as a parameter to a project method that can be seen by 4D Write Pro. For example, if the expression is as text:

```
"Query([People];[People]firstName="\Bob\")"
```

It will be encapsulated in a project method such as the following:

```
EVALUATION_METHOD("Query([People];[People]firstName="\Bob\")"
```

This will then be inserted back as an expression in text to the same location where it was found in the 4D Write Pro document. The resulting expressions in the document will look like the following:



The depiction looks quite messy, but it is in the correct locations as it was found in the document. In order to accomplish this transition, two methods are required:

- `EVALUATION_METHOD`
- `PROCESS_EXPRESSIONS`

`EVALUATION_METHOD`

This method takes the “text” parameter which is the found expression in a 4D Write Pro document that will evaluate the expression at run time with **PROCESS 4D TAGS** as shown below:

```
//-----
// Name: EVALUATION_METHOD
// Description: Method will evaluate the expression from 4D Write documents that
//              contain an expression
// Parameters:
// $1 (TEXT) - Expression in TEXT
//
// Output:
// $0 (TEXT) - Output of the PROCESS 4D TAGS command
// -----
C_TEXT ($1;$exp_t)
C_TEXT ($0)
If (Count parameters>=1)
    $exp_t:="<!--4DEVAL "+$1+" -->"
    PROCESS 4D TAGS ($exp_t;$0)
End if
```

The **`EVALUATION_METHOD`** needs to be registered in order for 4D Write Pro to use the method. The next method does the registering using **SET ALLOWED METHODS** and finds expressions the document.

`PROCESS_EXPRESSIONS`

This method will take in the 4D Write Pro object and traverse through the document to find every expression and create a text of **`EVALUATION_METHOD`**(“expression”) and insert back as the new expression and return a 4D Write Pro object evaluated as shown below:

```

//-----
// Name: PROCESS_EXPRESSIONS
// Description: Method will go through the Write Pro document to search for
//              expressions and replace them with the "EVALUATION METHOD" that
//              will be executed at run time.
// Parameters:
// $1 (POINTER) - Pointer to the Write Pro Object
//
// Output:
// $0 (OBJECT) - Updated Write Pro Object
// -----
C_POINTER($1;$fileLoc_obj)
C_TEXT($loc)
C_OBJECT($0;$obj)
C_LONGINT($length)
ARRAY TEXT($methodArray;0)

$fileLoc_obj:=$1
APPEND TO ARRAY($methodArray;"EVALUATION METHOD")
APPEND TO ARRAY($methodArray;"PROCESS_EXPRESSION")
SET ALLOWED METHODS($methodArray)

If (Type($fileLoc_obj->)=Is text)
  If (Test path name($fileLoc_obj->)=Is a document)
    $obj:=WP Import document($fileLoc_obj->)
  End if
Else
  $obj:=$fileLoc_obj->
End if

$length:=Length(ST Get text($obj;ST Start text;ST End text))

For ($i;1;$length)
  $type:=ST Get content type($obj;$i;$i)

  If ($type=2) // found expression
    $text:=ST Get expression($obj;$i;$i)
    $text:=Replace string($text;"\"";"\\")
    $Newtext:="EVALUATION METHOD ("+"\""+$text+"\""+") "
    ST INSERT EXPRESSION($obj;$Newtext;$i;$i+1)
  End if
End for

$0:=$obj

```

Here is an example of how the command will be used in 4D Write Pro object:

```

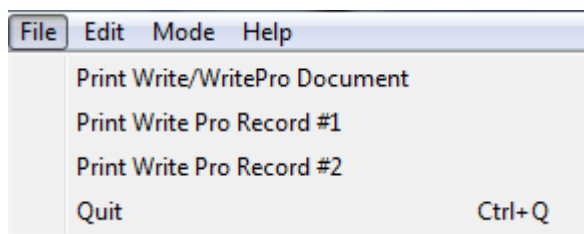
C_OBJECT($wpObj)
$wpObj:=PROCESS_EXPRESSIONS(->$wpObj)

```

With these two methods, a 4D Write Pro document can easily evaluate expressions and save it to a new 4D Write Pro object in a faceless manner. The next section will discuss a sample database demonstration.

Sample database

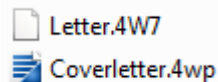
To give an idea of how this implementation works from the previous section, a sample database is provided which has menu options to open a file or field object record. The selected menu item will be printed with the evaluated expressions to a PDF without a 4D form present containing 4D Write Pro form object.



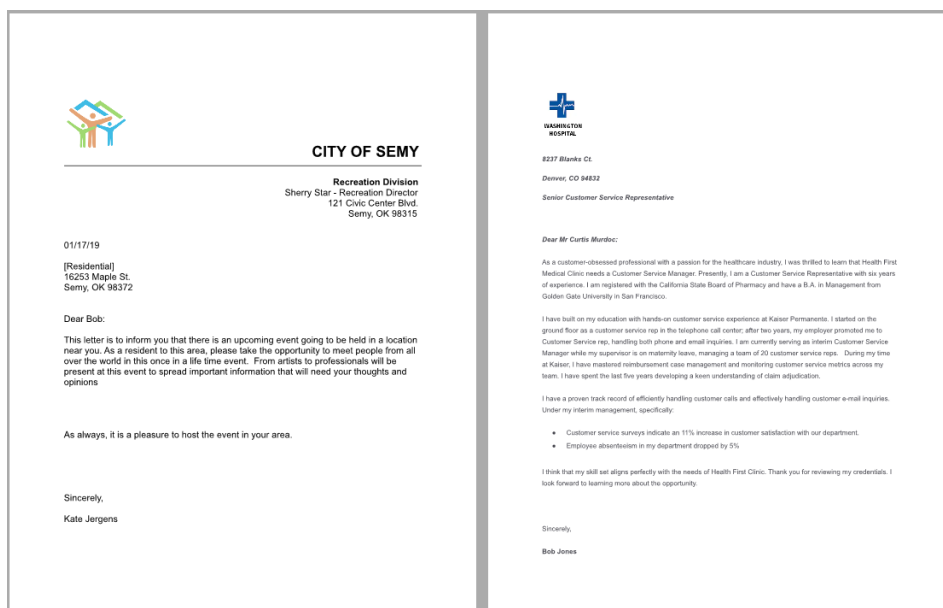
The selected menu items will be executed with the **PRINT_OFFSCREEN** method that determines if a 4D Write/Pro document or a 4D Write Pro object will be printed.

Printing from the 4D Write/Write Pro document

When selecting the first item of the menu, it will open a window that will display two files “Letter.4W7” and “Coverletter.4wp” which is 4D Write and 4D Write Pro respectively.




Opening either of these files will automatically process the file and preview the printed document:



Printing from the 4D Write Pro object from the object field

Selecting the next two menu items will read the record from the object field that contains a 4D Write Pro document. What will be printed are rental receipts from two customers:

 **RENTAL RECEIPT**

Date: 01/17/19

Received from: Curtis Murdoc

The Sum of: \$5500

As rent for the month of: January

For the rental property located at:
8237 Blanks Ct. Denver, CO 94832

Landlord: Bob Murdoc

Signature: _____

These menu item examples give an idea of how the process can be done without using a 4D Write Pro UI.

Conclusion

This Technical Note demonstrated the ability to process 4D Write Pro documents containing expressions in a faceless manner. 4D extends the functionality of 4D Write Pro by allowing the manipulation of the object programmatically similar to 4D Write's offscreen area. The Technical document examines that certain 4D commands such as ST "Style Text" as well as 4D Write Pro does not require UI intervention. The result of the processed document is then demonstrated by printing to PDF. The concept of "faceless" operations is very suitable for backend server-side operations which can be used in stored procedures and other jobs. As 4D Write Pro evolves, a document can completely be done programmatically.

Reference

Filter expressions contained in a 4D Write Pro document

<http://livedoc.4d.com/What-s-new/4D-Write-Pro-Reference-17/Filter-expressions-contained-in-a-4D-Write-Pro-document.300-3726276.en.html>