



ATOL PROBLEM RESOLUTION :
CONCAT in Entity with comment (ATOLBAS)
And Commented+ (FOUNDATION+)
PLEX Version greater or equal than 4.0b.
ATOL V4.0 Level 275
ATOL V4.5 Level 279
(09/01/00)



I. PURPOSE :	3
II. RESOLUTION FOR ATOLBAS/ENTITY WITH COMMENT :	3
1. STEP 1 :	4
2. STEP 2 :	4
3. STEP 3 :	4
4. STEP 4 :	5
III. RESOLUTION FOR FOUNDAT+/COMMENTED+ :	6
1. STEP 1 :	7
2. STEP 2 :	7
3. STEP 3 :	8
4. STEP 4 :	8
5. NOTES :	9



I.PURPOSE :

This document describes how to manage the problem concerning the modification around the Plex concat function in ATOLBAS/Entity with comment (ATOLBAS) and Commented+(FOUNDATION+).

In fact the CONCAT function has changed since Plex tool version 4.0b. Now the concat function concat two strings but right trims the two string before concat. For example in version 4.0 (first edition) String1 = "toto ", String2 = "titi " => String1 CONCAT String2 = "toto titi".

Since Plex 4.0 b the result is String1 CONCAT String2 = "tototiti".

For more information see article in

<http://cool2.sterling.com/support/plex2e/Plex/ConcatWincBehaviour.htm>

So you should have problems in ATOLBAS/Entity With Comment.Edit function and FOUNDAT+/Commented+.Comment+.ClientFunctions.Edit+ functions. When you try to save the new comments, it will be impossible because the database controller (Plex function) won't leave you to change this comment (Old values have changed).

To solve this problem, use the next steps, first for ATOLBAS/Entity With Comment and FOUNDAT+/Commented+.Comment+.ClientFunctions.Edit+ depending on the technology you use.

II.RESOLUTION FOR ATOLBAS/ENTITY WITH COMMENT :

The modification consists in adding a workaround which avoids the concat function in the ATOLBAS/Entity With Comment.Comment.Edit function.

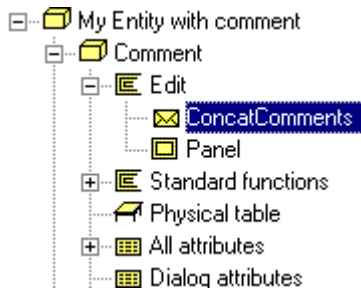
 Post Point 1 Build edit multiline

```
Set Details<_Edit multiline> = Details<_Edit multiline> CONCAT Base<_String>
```

These modifications have to be done in your ATOLBAS customisation layout, then all your comment function will be changed. Consider that you Entity With comment entity is customised by **My Entity with comment**.

1. STEP 1 :

Add a message "ConcatComments" to the **My Entity with comment.Comment.Edit** function.



2. STEP 2 :

Add the following parameter to your message :

My Entity with comment.Comment.Edit.ConcatComments	parameter	ATOLBAS/_Edit multiline ATOLBAS/_String
--	-----------	--

Fill it with the following statement :

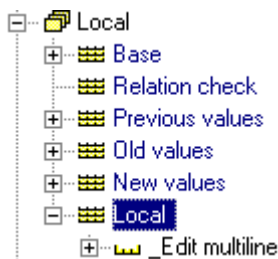
```
&(1:)&(2:)
```

3. STEP 3 :

Declare a new variable <Local> in the **My Entity with comment.Comment.Edit** function.

My Entity with comment.Comment.Edit	message	ConcatComment
	variable	OBJECTS/Local
	...as	Local

Add a ATOLBAS/_Edit multiline field in that variable.



4. STEP 4 :

Add the following lines in the **My Entity with comment.Comment.Edit** function (subroutine set panel field values)

```
Function My Entity with comment.Comment.Edit
  Pre Point 0 Set panel field values
  Set Local<_Edit multiline> = <_Edit multiline.Nul>

  Edit Point 1 Build edit multiline
  to avoid concat problem use format message method
  Format Message Message: My Entity with comment.Comment.Edit.ConcatComment, Local<_Edit multiline>

  Pre Point 1 Set panel field values
  +If Variant: ATOLBAS/Windows/AS400
  API Call Source code: ATOLBAS/_Components.String.Trim right.API

  Edit Point 1 Set panel field values
  strap Concat method
  Set Details<_Edit multiline> = Local<_Edit multiline>
  refresh oldvalues
  Set Old values<_Edit multiline> = Details<_Edit multiline>
```

Map parameter for FormatMessage as followed :

Type	To	From
I	<_Edit multiline>	Local<_Edit multiline>
I	<_String>	Base<_String>

Map parameter For Trim Right API as followed :

To	From
<_String>	Local<_Edit multiline>

III.RESOLUTION FOR FOUNDAT+/COMMENTED+ :

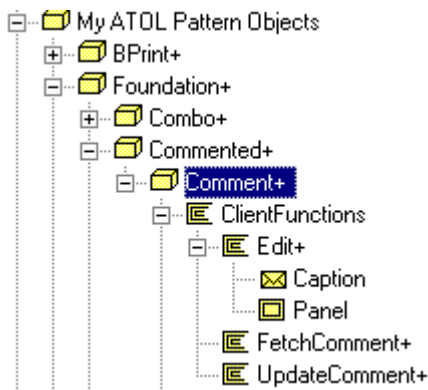
The modification consists in adding a workaround which avoids the concat function in the FOUNDAT+/Commented+.Comment+.ClientFunctions.FetchComment+ function.

```

While FetchL<RowsFetched> >= FetchL<Current>
  Set FetchL<UseRow> = <UseRow.Yes>
  Seq Comment
  Pre Point
  Edit Point Set data+ ?
  Post Point
  If FetchL<UseRow> == <UseRow.Yes>
    Pre Point
    Edit Point Process data fetched+
    Set Output<CommentText+> = Output<CommentText+> CONCAT StringL+<String+>
    Post Point Process data fetched+
  Set FetchL<Current> = FetchL<Current> + <Current.*One>

```

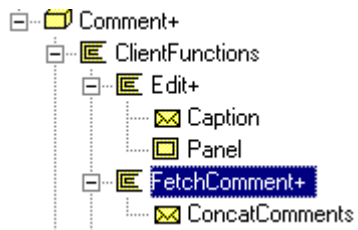
These modifications have to be done in your FOUNDAT+ customisation layout, then all your comment function will be changed. Consider that Commented+ entity is customised by **My ATOL Pattern Object.Foundation+.Commented+..**





1. STEP 1 :

Add a message "ConcatComments" to the ...Comment+.ClientFunctions.FetchComment+ function.



2. STEP 2 :

Add the following parameter to your message :

My ATOL Pattern Objects.Foundation+.Commented+.Comment+.ClientF	parameter	UISTYLE+ /CommentText+
		UISTYLE+ /String+

Fill it with the following statement :

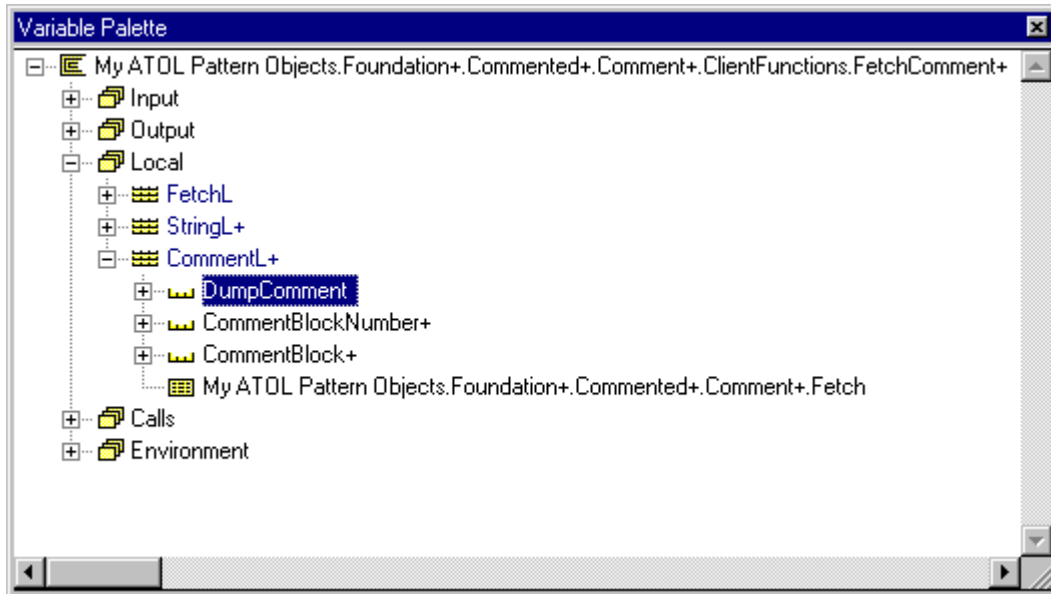
```
&(1);&(2)
```

3. STEP 3 :

Declare a new field Which inherits from CommentText+, for example DumpComment.

DumpComment is a UISTYLE+/CommentText+.

Add this field in the CommentL+ variable of your ...Comment+.ClientFunctions.FetchComment+ function.



4. STEP 4 :

Add the following lines in the your ...Comment+.ClientFunctions.FetchComment+ function (subroutine CallBlockFetch+)

```
Function My ATOL Pattern Objects.Foundation+.Commented+.Comment+.ClientFunctions.FetchComment+
  Pre Point Process data fetched+
  strap concat method
   Format Message Message: My ATOL Pattern Objects.Foundation+.Commented+.Comment+.ClientFunctions.FetchComment+.ConcatComments, CommentL+<DumpComment>
  Post Point Process data fetched+
  concat line before is ignored
  Set Output<CommentText+> = CommentL+<DumpComment>
```



Map parameter for FormatMessage as followed :

I	<CommentText+>	Output<CommentText+>
I	<String+>	StringL+<String+>

5. NOTES :

As the ...Comment+.ClientFunctions.FetchComment+ inherits from UISTYLE+/FetchComment+. If you have a customisation layout for UISTYLE+ funcios you should do these modification in that customised function.