

Bug Fixes

ID : 1268**Fixed in version :** 3.1.8**Short Description:** Unable to print PDF

Full Description: A customer of ours says his customers can't print the PDF documents he sends. The PDF's are generated with PDFDevice 3.0.0
Is there anything you notice with these document which may explain this behaviour ?
The security settings allow for printing.
Maybe some font used ?

Comments : This problem is related to the kDevPdfConvLinks option and occurred as the report contained a web site address followed by a closing bracket. PDFDevice mistakenly included the closing bracket ')' as part of the link when it annotated the link. PDFDevice has been changed to consider punctuation characters such as brackets and excludes them from any links that are encountered.

ID : 1293**Fixed in version :** 3.1.6**Short Description:** Printing a Report Omnis Server Crashes

Full Description: pdfdevice v3.1.5 causes Omnis Server to crash. The reason for that isn't to much data. It seems that it is one or more non visual characters.

Here is an example of the text which causes problems:

"﻿﻿Ziegeleistrasse 17/A"

In front of the character Z are two non visual characters. I hope they don't get lost when submitting the request.

I attached two Omnis Disk images (server_print.rep and dev_print.rep) in the print_to_disk.zip which Michael asked for. server_print.rep was printed with a Omnis Server and dev_print.rep with Omnis Develop.

Comments : Unfortunately, we have been unable to reproduce a crash. So the only immediate solution that we can offer is that we handle the characters that are causing the issue. From now on PDFDevice will strip out byte order characters U+FFFE and U+FEFF as these character do not perform any useful function within the context of a PDF file.

This issue has now been identified, please see release notes for case 1299.

ID : 1299**Fixed in version :** 3.1.7**Short Description:** When printing to PDF Device Omnis crashes

Full Description: this problem is like the one of "Case ID 1293" except it seems to be a different character which causes the problem.
The character
"  " U+F04A
seems to cause the problem. When i remove the character it prints fine.

Like Case ID 1293 it only crashes when Omnis runs as a service and it also crashes when Omnis Server runs in window mode.

Comments : It appears that when Omnis runs as a windowless service, Omnis does not initialise the window's COM object library. Without this initialisation PDFDevice fails to fetch the multi language support object which allows PDFDevice to map characters that are not supported in one font to a font where they are supported. This interface was assumed to always be accessible and a vital test was missing which resulted in a crash in a windowless service application when unusual characters were encountered.

We believe the reason the COM library is not initialised in this case is because Omnis does not directly initialise the COM library and it is only initialised in a windowed application when Omnis loads OLE support which initialises the COM library indirectly. There may also be other windows functionality that Omnis utilises which will initialise the COM library when called.

Having studied the requirements for initialising the COM object library we have found it somewhat ambiguous in relation to what has to be done from within a plug in external DLL when the main application code does not initialise the COM library by default but may attempt so later. It is made ambiguous in two ways. Firstly, there are two ways of initialising the library and choosing the correct one is vital for the operation of the entire application. Secondly, our DLL may or may not be executed on the main thread. If the COM library is initialised incorrectly within the context of Omnis some parts of Omnis that require the COM interface may stop working.

Based on the information available to us, we have implemented a mode of initialisation that is compatible with OLE and which is executed when Omnis failed to initialise the COM library first. We were unable to fully resolve potential ambiguities relating to different thread execution and the impact on COM library initialisation. We therefore recommend that, if PDFDevice is used in a web server and is run as a service, PDFDevice is stress tested by forcing multiple thread execution, i.e. at least two simultaneous requests that produce PDF files. Unfortunately, COM object initialisation appears to be very library dependent (i.e. what functionality your library utilises and in what order) so we were unable to devise a simple test that would cover all eventualities.

However a failure by Omnis or PDFDevice to initialise the COM library will no longer result in a crash. At worst, there will be a small performance overhead when PDFDevice attempts to manually map a suitable font or if that is not possible characters that cannot be mapped to a font may appear as invalid characters within the PDF output. This can take on the appearance of a square where the invalid character is located.