

An Insight into Adobe Document Services (ADS)



Applies to:

This article applies to SAP NetWeaver, WAS (Web Application Server), Web Dynpro Java/ABAP, Adobe Interactive Form development.

Summary

Developers who work with Adobe Interactive Forms in their Web Dynpro Applications need to install credentials and configure Adobe Document Services (ADS). If, “What is ADS exactly?”, is the question in your mind then indeed this article has the answers.

In this article we will learn a bit about what is Adobe Document Services (ADS) and that what does it do for us and when exactly, an overview of how ADS fits into the Web Application Server (WAS) architecture and the scenarios in which it plays an active role.

Author: Amol Gupta

Company: Wipro Technologies

Created on: 8 April 2008

Author Bio



I am Amol Gupta. I have been working with Wipro Technologies on Web Dynpro programming, Enterprise Portal Content Management, Java API for accessing MDM, MDM Console, MDM Data Manager, Interactive forms for Web Dynpro Java . I also know a bit of Java (SCJP) and ABAP programming.

Table of Contents

Prerequisites	3
Outcomes.....	3
Adobe Document Services (ADS) is a Web Service	4
Why do I need the Adobe Document Services (ADS)?	4
What is Adobe Document Services (ADS)?	4
Full integration with SAP Netweaver	5
Adobe Solution	6
The Better View	6
Overview - Runtime	7
Online Interactive Form Scenario- Architecture.....	8
Offline Interactive Form Scenario- Architecture.....	9
ADS and related services	10
What are XDC files	10
What is "ERROR.pdf"	10
Adobe Document Services (ADS) related transactions on the Web Application Server (WAS)	11
1) Installing the credentials and configuring ADS	11
2) Checking if the ADS service is running alright.....	11
3) Viewing the logs.....	11
4) Activating the trace for Adobe Document Services	11
Related Content.....	12
Disclaimer and Liability Notice.....	13

Prerequisites

You should have access to Portal and WAS.
(That implicitly means that the WAS is installed already and running)

It would be to your advantage if you have brushed up with ADS somewhere already, for example having an idea of installing Credentials and Configuring the ADS already would definitely help. But even if you are starting with this document itself, I will suggest that you get a rough/vague idea about the ADS.

For your reference you can go through the following articles:-

[Tutorial on how to install credential and configure ADS.](#)

[How to check if the ADS is running fine](#)

Ideally the target audience will be developers who have already touched upon ADS somewhere but don't have a clear idea about it.

Outcomes

After going through this article you'll have a good understanding of ADS. Questions like

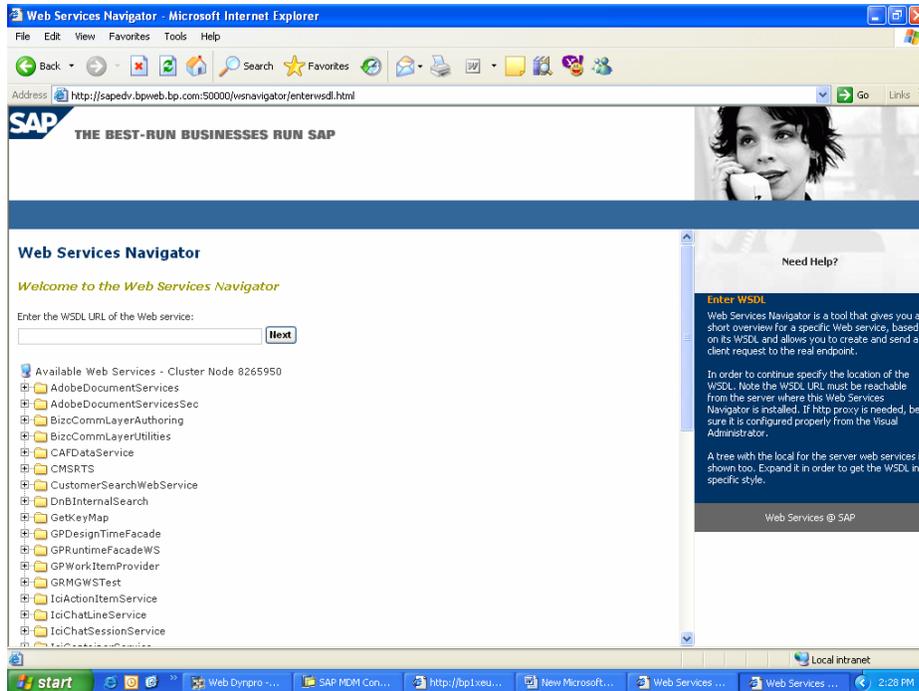
- ✓ What is ADS?
- ✓ How it works?
- ✓ When it works?
- ✓ How does it fit into the WAS architecture?
- ✓ When does the ADS come into the picture?
- ✓ What related transactions one can do in WAS?

will be answered satisfactorily. You will have a sound foundation of basic knowledge on which you can build on when you come across related topics from time to time.

Adobe Document Services (ADS) is a Web Service

I'll begin this article with the information that the Adobe Document Services (ADS) is actually a Web Service that can be viewed in the Web Service navigator which is the Web Service homepage of your WAS (Web Application Server).

Go to link "http://<YourServerName>:50000/wsnavigator". You will see all the Web Services Available on your WAS, ADS being one of them.



Why do I need the Adobe Document Services (ADS)?

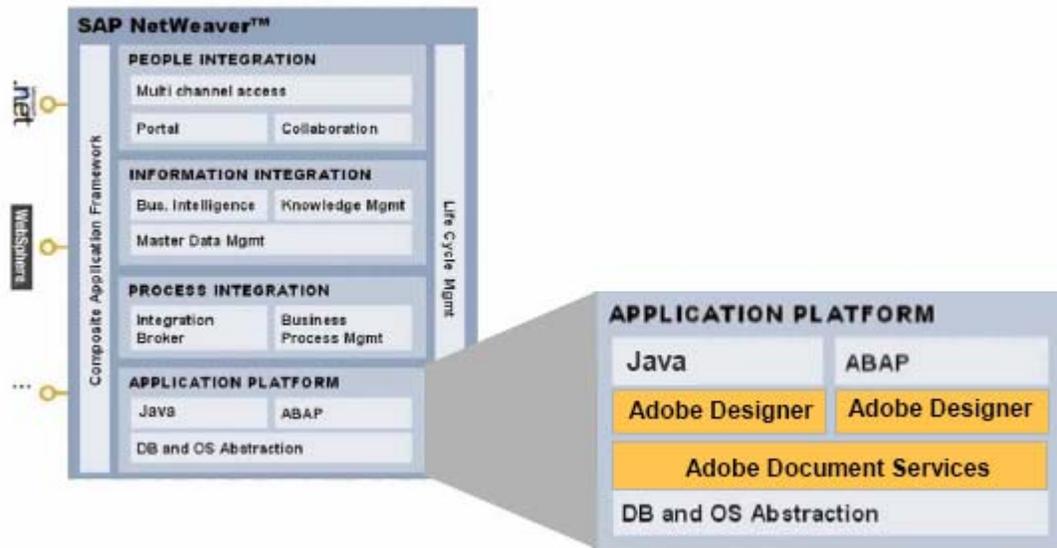
The document services are the run-time component of Interactive Forms and have two fundamental tasks:

- ✓ Creating a PDF file (and other output formats in print scenarios) by merging a form template with current system data for output
- ✓ Extracting user-entered data (stored in XML format inside the PDF) from interactive forms and return the data to the system

What is Adobe Document Services (ADS)?

- ✓ ADS is implemented as Web Service on SAP Netweaver 04
- ✓ It generates a pdf based on
 - Template (XML Form Template - XFT) : layout description
 - Data (XML Form Data - XFD): data
- ✓ ADS applies Adobe Reader Rights to the PDF
- ✓ It Extracts data from a PDF
- ✓ Additional Features
 - Encryption
 - Digital Signatures
 - Certificates

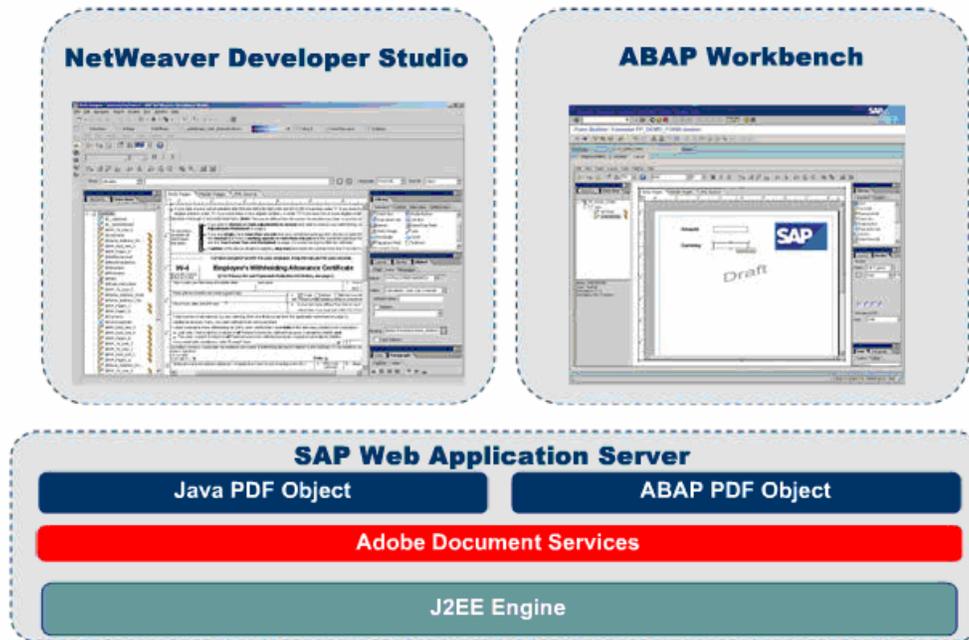
Full integration with SAP NetWeaver



Both Adobe Designer and Adobe document services are fully integrated into SAP Web Application Server.

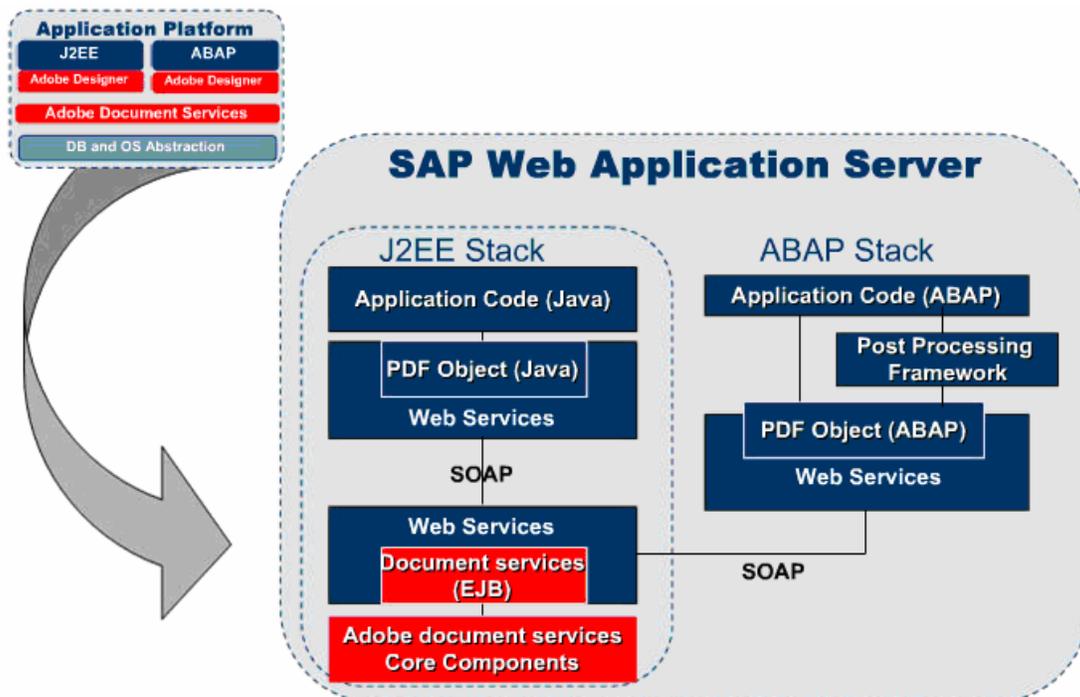
- ✓ **Adobe Designer**, the user-friendly, state-of-the-art form design tool, is integrated into the ABAP Workbench and the SAP NetWeaver Developer Studio, SAP's development environment for Java. When you work in either environment, you can access Designer in place, that is you do not need to leave your workspace to be able to design your form.
You use Designer to create the layout of your form and the binding of the required system fields against the layout objects on the form.
- ✓ **Adobe document services**, the run-time component of the Adobe integration, are Web services deployed on the J2EE Engine of SAP, which is the Java stack of SAP Web Application Server. The services render the form template created in Designer and the current system data as well as all the interactive elements of your form into PDF format (as well as other formats such as PS, PCL or ZPL in the case of high-volume print scenarios).
- ✓ **WebDynpro applications** can easily be integrated in your SAP Enterprise Portal framework, the people integration part of SAP NetWeaver.

Adobe Solution

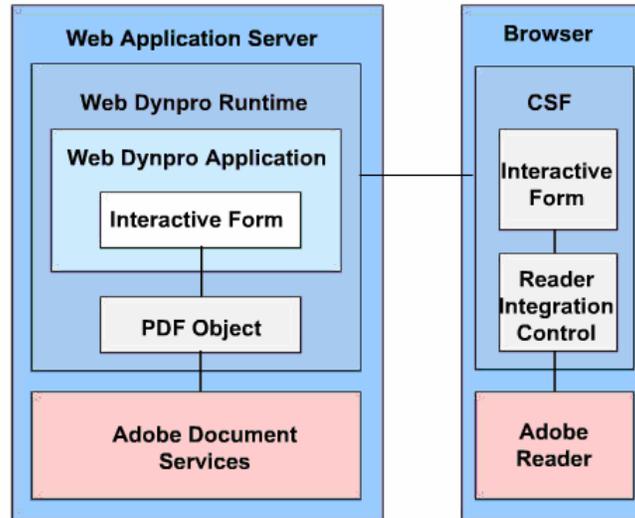


Note: Integration has been provided for both the Java as well as ABAP stack. Note that there is a PDF Object in both ABAP stack and Java stack. The PDF Object calls the ADS service. The PDF object (one for Java and one for ABAP) communicates with the Adobe document services at run time. The PDF Object encapsulates the functions and features made available by the services. PDF Object, which offers an easy to use API for Adobe document services calls

The Better View

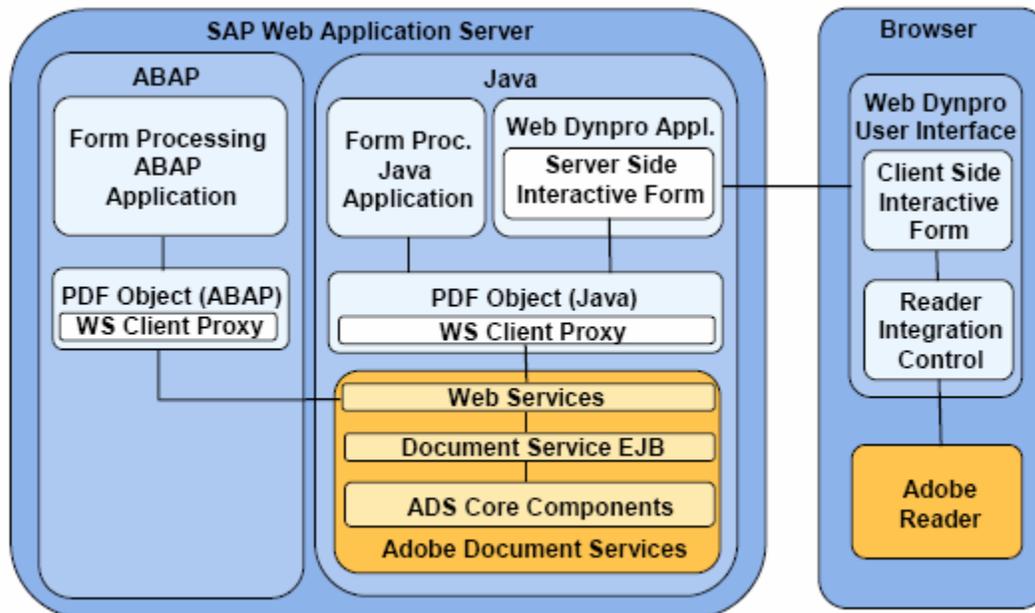


Overview - Runtime



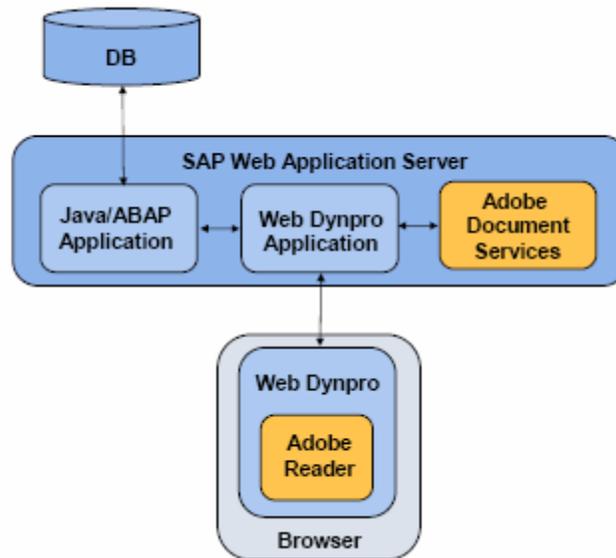
To make the application developer's job even easier, SAP provides the so-called *PDF Object* (one for Java and one for ABAP) for communicating with the Adobe document services at run time. The PDF Object encapsulates the functions and features made available by the services. PDF Object, which offers an easy to use API for Adobe document services calls.

The browser is there at the client. There is a Adobe Reader installed at the client as a plugin and that's what displays the PDF. On the server side you have a Web Dynpro Runtime and the interactive form is running within the Web Dynpro Application so whenever it encounters a PDF it use the PDF Object interface to make a call to the Adobe Document Services which serves out the PDF in the Web Dynpro application which is served to the browser and the user sees in the adobe reader.



Note: The Adobe Document Services are deployed on the Web Application Server and offers services for generating PDFs based on the layout description and data or for the extraction of data from a PDF. ADS is written in Java. So they need to install the Java Stack on the SAP Web AS. Therefore the ABAP side communicates with the Java side through a Web Service Client Proxy. The Document Service EJB talks with the underlying ADS.

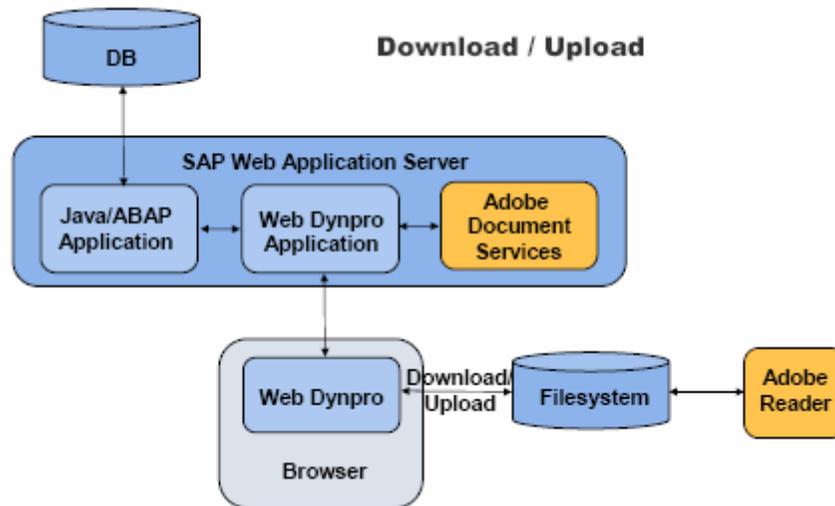
Online Interactive Form Scenario- Architecture



On this figure, you see the architecture of an online Interactive Form scenario.

- ✓ An Interactive Form is displayed and changed in the Adobe Reader plug-in during the execution of a Web Dynpro application in a browser.
- ✓ It is an online Interactive Form scenario, because the user is connected to the server while he enters the data in the form.
- ✓ *Using the Adobe document services, the Web Dynpro application generates an InteractiveForm or extracts data from an Interactive Form.*
- ✓ The Web Dynpro application gets the data it displays from or writes the data it receives to the backend using the business logic, which is realized in an ABAP or J2EE application.
- ✓ The business logic communicates with a database in order to read or write the data.
- ✓ You can only develop scenarios using Interactive Forms in the Web Dynpro for Java environment as well as with Web Dynpro for ABAP.

Offline Interactive Form Scenario- Architecture



In this figure, you see the architecture of an offline Interactive Form scenario based on down-/upload functionality.

- ✓ You can download an Interactive Form to your local file system using a Web Dynpro application.
- ✓ You can then display and change the Interactive Form locally in Adobe Reader.
- ✓ It is an offline Interactive Form scenario, because the user is not connected to the server while he enters the data in the form.
- ✓ After entering or changing the data, you upload the Interactive Form from your local file system using the Web Dynpro application.
- ✓ *Using the Adobe document services, the Web Dynpro application generates an InteractiveForm or extracts data from an Interactive Form.*
- ✓ The Web Dynpro application gets the data it displays from or writes the data it receives to the backend using the business logic, which is realized in an ABAP or J2EE application.
- ✓ The business logic communicates with a database in order to read or write the data.
- ✓ You can develop scenarios using Interactive Forms in the Web Dynpro for Java environment as well as ABAP environment.

ADS and related services

Adobe document services consist of the Adobe document services Web service and these SAP J2EE services:

- Document Services Data Manager
- Document Services Font Manager
- Document Services License Service
- Document Services Trust Manager Service
- PDF Manipulation Module (High Encryption or Low Encryption)
- XML Form Module

These services are installed together with SAP Web AS and must all be running to enable Adobe document services to operate correctly. In the event of problems, you can verify the state of these services using the Visual Administrator.

What are XDC files

An XDC file is a printer description in XML format. Adobe document services require this file to create the print files. PDF-based forms can only be printed on printers whose SAP device type have an XDC file in the system.

When Adobe document services are deployed to the SAP Web AS the XDC files are located in this directory: /usr/sap/[systemID]/sys/global/AdobeDocumentServices/lib.

In some cases, it may be necessary to make changes to these files. You upload XDC files to the server in the following cases:

- You want to install a new XDC file.
- You want to install a corrected XDC file.
- You want to install an XDC file that you have modified.

What is “ERROR.pdf”

When the rendering of a form fails, Adobe document services create a file that contains detailed information about the error, called Error.PDF. This file is written on the server that hosts the Adobe document services in the directory <DIR_GLOBAL>\AdobeDocumentServices\renderErrorLog\ErrorFiles. The file name is <Date+Time+ApplicationName+Username>.pdf. The default maximum size of the error file directory is 100 MB. After it creates the Error.PDF, the system examines the directory to determine the total size of files in the directory. If that size is more than the maximum allowed, it begins deleting files (oldest first) until the directory size is below the maximum allowed or until only one error file is left in the directory.

You can change the Maximum Size for the Storage of the ERROR.PDF File

Adobe Document Services (ADS) related transactions on the Web Application Server (WAS)

1) Installing the credentials and configuring ADS

For this you can check the following article. [Tutorial on how to install credential and configure ADS](#)

2) Checking if the ADS service is running alright

For this you can check the following article. [How to check if the ADS is running fine](#)

3) Viewing the logs

All of the components of Adobe document services work together and record events to the logs, including any service errors. Security-related messages are logged in the security.log file of the J2EE Engine. This log file contains information on the user and the actions he performed like certifying or signing a form and the used credentials. An easy way to find the appropriate information is to search for the location com.adobe.AdobeDocumentServices. This filters out entries that match the following pattern:

Date, time: User: <name>action that was performedwith credential alias <alias>

4) Activating the trace for Adobe Document Services

To analyze a problem with Adobe document services, switch on the trace and reproduce the problem. The trace entries then provide you with detailed control flow information. If you activate the trace for Adobe document services, all business data to be printed out or displayed in a PDF, will be recorded as Base 64-encoded data in the trace file. To avoid the storage of productive or security-related data, do not activate the trace.

Related Content

[Tutorial on how to install credential and configure ADS](#)

[How to check if the ADS is running fine](#)

[Adobe Forms for WebDynpro Java](#)

[Interactive Forms Overview Presentation](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.