

# SAP Light Debugger



SAP Light Debugger .....	3
Using SAP Light Debugger .....	5
Reference.....	7



## SAP Light Debugger

SAP Light Debugger is a supportability tool designed as enhanced means of analyses and troubleshooting of SAP NetWeaver Application Server Java (AS Java) and applications running on top. SAP Light Debugger offers features that enhance the debug capabilities of the standard debugger provided with the Java Development Kit (JDK) and allows for efficient debugging to be performed using a simple GUI on customer site where neither a complete integrated development environment (IDE) is available, nor the source code of SAP modules is in place. The tool can be used by SAP support or any Java developer who has difficulties connecting their local SAP NetWeaver Developer Studio to AS Java due to firewall restrictions or other reasons.

### Prerequisites

- The problem should be reproducible with concrete steps and you need to know how (or when) it occurs.
- The Java process which will be debugged should be started in debug mode. SAP JVM allows enabling debugging on the fly without restart of the process.

### Features

SAP Light Debugger can be used in complex Java scenarios that would take too much time with the standard troubleshooting approach, or, if the default logging/tracing shipped with the product does not provide sufficient information for analysis. SAP Light Debugger is also suitable when reproducing the problem in an SAP landscape is not possible for example due to network specifics of the landscape or due to the fact that the problem is data-driven. The most efficient and at times the only way to find a program error in such a case is to directly investigate on the system where the problem occurs and to examine the data on the fly using SAP Light Debugger.

### Activities

Besides the typical debugging capabilities available for Java debuggers, SAP Light Debugger also exposes a plugin framework that allows you to execute actions specific to AS Java for even more efficient and comprehensive troubleshooting. Using the plugins of SAP Light Debugger, you can:

- Export a thread dump to a text file
- Export a variables dump to a text file
- Trace HTTP traffic
- Convert the time in the log files (presented as integer in milliseconds) to human readable date and time

- Display information about a given class which is loaded on the server and open it in the source tab area
- Find meta information about the jars loaded in the server
- Extract the System Information from the jars MANIFEST.MF and SAPMANIFEST.MF file
- Trace all exceptions that occur on the debugging VM
- Open jar/ear/zip/sda/sca archive files from the file system.

Some features of SAP Light Debugger, such as extracting the source code of a particular class by connecting to the SAP Perforce Depot, are not available outside SAP Network.

### **More Information**

[Using SAP Light Debugger](#)

[Reference](#)

## Procedure

### Starting the tool

1. Go to the SAP Light Debugger installation directory. It is located in *<AS Java Installation Dir>/support/lightdebugger*.
2. Start SAP Light Debugger, depending on the OS of the system to be debugged:
  - Windows : Execute the *start.bat* which is located in the installation directory.
  - Unix : Local debugging directly with GUI on the Unix host
    1. Make sure a graphical desktop environment is enabled on the system.
    2. Execute the *start.sh* script which is located in *<AS Java Installation Dir>/support/lightdebugger directory*
  - Unix: Local debugging with GUI on a Windows host
    1. Make sure XServer is installed and enabled on the Windows host.
    2. Export the DISPLAY variable on the Unix host to point to the XServer on the Windows host.
    3. Execute the *start.sh* script which is located in *<AS Java Installation Dir>/support/lightdebugger directory*.

### Connecting to a system

1. Make sure that the Java process which will be debugged is started in debug mode:
  - a. In SAP MC, select the *AS Java* node and choose *Process Table*.
  - b. In the result view, choose the server process on which you want to open a debug port and, from the context menu, choose *All tasks -> Enable debugging*.
2. After you have started SAP Light Debugger, a “*Connect to*” dialog appears where you specify the host and port of the Java process to which you want to connect the debugger and choose “*Yes*”.
3. Once the debugger is connected, you can see a list of all threads currently running in the server process.

## **Debugging**

After you have connected to the system, you can proceed with the debugging activities as with any other standard debugger. SAP Light Debugger provides all the standard debugging functions, including step execution, setting breakpoints (including conditional ones) and values, inspecting variables and values, and suspending and resuming threads.

While debugging, each thread in your program appears as a node in the tree. If the thread is suspended, its stack frames are shown as child elements.

SAP Light debugger also offers some additional capabilities. For more information, see [Reference](#).

 **Reference**

Here is a list of the features and plugins available in SAP Light Debugger that come on top of the debug capabilities of a standard debug tool.

<b>Feature</b>	<b>Description</b>
<b>Info</b>	<p>Gives information about what is read from the MANIFEST.MF and SAP_MANIFEST.MF of the jar file in which the selected class is located. In case the debugger and the debugged program are on different hosts, this functionality will work if:</p> <ul style="list-style-type: none"><li>▪ the remote host is mapped as a drive to the local host</li><li>▪ the following properties, which are located under “Additional -&gt;Options”, are configured as follows:<ul style="list-style-type: none"><li>○ PATH_MAPPING_FROM – set the path which you want to replace at runtime with the path in which the jars will be searched</li><li>○ PATH_MAPPING_TO – set the corresponding part of the path to which the path will be replaced.</li></ul></li></ul> <p>For example, if the debugger is on host ABC and the debugged server is on host DEF on drive d, set the above properties as follows:</p> <pre>PATH_MAPPING_FROM = d:\ PATH_MAPPING_TO = \\DEF\d\\$</pre>
<b>Check</b>	Compares the source file which is in the selected tab with the executable source which is loaded in the debugged VM.
<b>Perforce</b>	Opens a new frame with the selected class fingerprint. With this fingerprint, the source of the class as it is in Perforce, can be retrieved by one service which is exposed internally.
<b>Decompile</b>	Opens a new frame with the selected class byte code. With this

	byte code passing it to one service which is exposed internally you can retrieve the source code.
<b>Compile</b>	<p>Enables the following activities:</p> <ul style="list-style-type: none"> <li>○ Add Jar/Directory – adds a new archive or directory from the file system of the debugger to the compilation classpath</li> <li>○ Add Jars Recursively - sets a directory as a parameter and all jars which are in it and its subfolders will be added recursively</li> <li>○ Remove - removes the selected entries from the classpath list</li> <li>○ Filter classpath - allows you to exclude applications or directories from the classpath</li> <li>○ Compile class - saves the content of the tab inside a source file with the same name in the src directory and compiles it against the current classpath</li> <li>○ Hotswap Class - enabled only after the compilation of the class. It replaces the selected class byte code which was loaded inside the server VM with the class byte code which was just compiled. This allows you to test the changes made directly on the customer system.</li> <li>○ Tested with ok? - after you have hotswaped the the class and everything is working fine, choose this button to indicate that the fix is tested and can be patched inside a jar.</li> <li>○ Patch Jar - finds the original jar from which this class has been loaded and patches the jar inside the src directory which was previously created.</li> <li>○ Replace Jar - backs up the original jar via renaming it with appending an index to it and pastes the patched jar instead.</li> </ul>
<b>Save All</b>	Saves the content of all currently opened tabs inside a new file with the same names as the tab itself and puts them inside the src directory



<b>Save</b>	Saves the content of the currently selected tab inside a new file with the same name as the tab and puts it in the src directory.
<b>Date</b>	Converts a timestamp to a human readable date. It is useful when you have to look at some traces and the timestamps in the logs are not meaningful values.
<b>Dump Threads</b>	Creates a thread dump in a text file.
<b>Dump Variables</b>	Creates a variables dump in a text file.
<b>HTTP Watch</b>	Collects and displays information about HTTP traffic. Use it when the issue you want to investigate is related to HTTP traffic and you want to know what is sent and received via HTTP protocol.
<b>Log Time Converter</b>	Converts the time in the log files presented as integer in milliseconds to human readable date and time.
<b>Tracing Exceptions</b>	Traces all exceptions that occur on the debugging VM. It is useful when exceptions are thrown but not logged in the default trace or their stack trace is missing there.
<b>Zip/Jar</b>	Opens jar/ear/zip/sda/sca archive files from the file system. This can be used when the system has no archiving utility capable to unpack formats like SDA and SCA.

## Copyright

© Copyright 2012 SAP AG. All rights reserved.

SAP Library document classification: PUBLIC

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, PowerPoint, Silverlight, and Visual Studio are registered trademarks of Microsoft Corporation

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, z10, z/VM, z/OS, OS/390, zEnterprise, PowerVM, Power Architecture, Power Systems, POWER7, POWER6+, POWER6, POWER, PowerHA, pureScale, PowerPC, BladeCenter, System Storage, Storwize, XIV, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, AIX, Intelligent Miner, WebSphere, Tivoli, Informix, and Smarter Planet are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are trademarks or registered trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems Inc.

HTML, XML, XHTML, and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Apple, App Store, iBooks, iPad, iPhone, iPhoto, iPod, iTunes, Multi-Touch, Objective-C, Retina, Safari, Siri, and Xcode are trademarks or registered trademarks of Apple Inc.

IOS is a registered trademark of Cisco Systems Inc.

RIM, BlackBerry, BBM, BlackBerry Curve, BlackBerry Bold, BlackBerry Pearl, BlackBerry Torch, BlackBerry Storm, BlackBerry Storm2, BlackBerry PlayBook, and BlackBerry App World are trademarks or registered trademarks of Research in Motion Limited.

Google App Engine, Google Apps, Google Checkout, Google Data API, Google Maps, Google Mobile Ads, Google Mobile Updater, Google Mobile, Google Store, Google Sync, Google Updater, Google Voice, Google Mail, Gmail, YouTube, Dalvik and Android are trademarks or registered trademarks of Google Inc.

INTERMEC is a registered trademark of Intermec Technologies Corporation.

Wi-Fi is a registered trademark of Wi-Fi Alliance.

Bluetooth is a registered trademark of Bluetooth SIG Inc.

Motorola is a registered trademark of Motorola Trademark Holdings LLC.

Computop is a registered trademark of Computop Wirtschaftsinformatik GmbH.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

## Typographic Conventions

Type Style	Description
<i>Example text</i>	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.
<b>Example text</b>	Cross-references to other documentation. Emphasized words or phrases in body text, graphic titles, and table titles.
EXAMPLE TEXT	Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.
Example text	Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
<b>Example text</b>	Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.
<Example text>	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
EXAMPLE TEXT	Keys on the keyboard, for example, F2 or ENTER.

