



ORACLE®

Agile9 SDK Cache

Jie Chen
May 2010



You will learn

- Information of Tmp @ Server & Client
- The principle of cache working
- Simple ways to validate Cache related issues



Tmp @ Server

Where

- <http://server/Agile/ServerAPIProperties>
%tmpdir%\sdk.extensions.libs\
- Not a real cache

```
# java.io.tmpdir=C:\DOCUME~1\jichen\LOCALS~1\Temp\  
# java.io.tmpdir.readable=true  
# java.io.tmpdir.writable=true  
# sdk.extensions=D:/Agile/Agile93/integration/sdk/extensions  
# sdk.extensions.readable=true  
# sdk.extensions.writable=true  
# cookie.domain=.sl.agilesoft.com
```



Tmp @ Server

- **What**

JARs' copy deployed in Agile Extension

- i. Common PX
- ii. Event PX
- iii. Third-party JARs

- **When**

The time of any PX invoked
or the Actual Jar has newer timestmaps

05/20/2010 11:11 PM	1,509 TestNullResult-0.139911.jar
05/23/2010 12:27 AM	1,509 TestNullResult-0.18166.jar
05/29/2010 12:30 AM	1,510 TestNullResult-0.18168.jar



Tmp @ Server

- **Quizz**

Customer developed PX *TestNullResult.jar* and deployed. Several days later he modified the code and redeployed by overwriting, and once again one week later. Now he sees three similar Jars deployed in Tmp folder and is upset if Agile will invoke the wrong one.

05/20/2010 11:11 PM	1,509 TestNullResult-0.139911.jar
05/23/2010 12:27 AM	1,509 TestNullResult-0.18166.jar
05/29/2010 12:30 AM	1,510 TestNullResult-0.18168.jar

Agile dynamically loads the correct version of Jar by comparing the last modified timestamp.

Cache @ Client

- **How**

- I. First time to run SDK client, classes loaded from server to local Tmp
- II. Further running will load classes from local cache directly

JVM: -Dncl.printload=true -Dncl.printfind=true

```
Loading 'com.agile.api.pc.Session' ... loaded (total bytes: 39154)
Loading 'com.agile.api.common.IResourceBundleHolder' ... loaded (total bytes: 39357)
Loading 'com.agile.api.pc.APIObject' ... loaded (total bytes: 46607)
```

```
Loading 'com.agile.api.pc.Session' ... loaded from cache
Loading 'com.agile.api.common.IResourceBundleHolder' ... loaded from cache
Loading 'com.agile.api.pc.APIObject' ... loaded from cache
```



Cache @ Client

- **Where**

```
System.out.println(System.getProperty("java.io.tmpdir"));  
%tmpdir%\AgileSDK.cache\
```

- **Cached Files**

<host-port>_Agile_.cache

containing all the downloaded classes (binary file)

<host-port>_Agile_.properties

containing the file size and index to the file pointer in the file *<host-port>_Agile_.cache*

slag9301w3-1.sl.agilesoft.com-1_Agile_.cache

slag9301w3-1.sl.agilesoft.com-1_Agile_.properties



Cache @ Client

- **SDK API Version**

current-server-version

current-impl-version

- **Server:**

`http://server/Agile/ServerAPIProperties`

- **Client:**

`<host-port>_Agile_.properties`



Case Study 1

- 3-1640106631

Context

Customer was on 9301 and had SDK client running fine. Later one HF was applied.

The HF includes Agile API fix. When customer ran PX again it fails with below error:

Error of Local Class Incompatible: Stream Classdesc SerialVersionUID

Analysis

SDK client not modified

Reviewed API code, no introduced issue found

SDK client uses that API fix

Check SDK cache, existing!! API versions do not match

Conclusion

Different class Serial Version between Cached class and remote server class.

Solution: Purge local cache



Case Study 2

- 3-1510903003

Context

Customer has PX running fine in 9300. Client developer modified class in the PX Jar and redeployed. But they found Agile always called the old function of Jar. That is to say, the newly modified class does not take effect.

Server OS: Redhat Linux

Analysis

Customer re-build the jar in Linux. Somehow the jar still shows the old timestamps.

Conclusion

Cause Unknown, possible JAR issue in Linux

Workaround: Use Linux command “touch” to activate



ORACLE®

Thanks & Q/A