

□□□-□□□□□□□□ □□□□□□□□ का उपयोग □□□□□□□□ □□□□□□□□ में

[illegible]

1. 00000000 00000000 00 00000 00 000 000000 00000000 00000000

```
0000000000 0000000000 (0.0., password="{aes}ABC123...") 00 000000 0000000000 000000 000000 0000. 00000 000000 000000
000000000000 00 00000 0000 00 00000 0000000000 00 00000000 0000, 00 0000000000 00 securityUtility 00000 0000 0000 0000000
0000.
```

□□□□□□ □□ □□□□□□□□:

```
# Liberty ke bin directory mein navigate karein
cd /path/to/wlp/bin
```

```
# securityUtility decode command run karein
```

```
./securityUtility decode \  
--server=<LibertyServerPath> \  
--key=<KeyAlias> \  
--keyPassword=<KeystorePassword> \  
--source="{aes}ABC123..."
```

```

00000000 00000000: - <LibertyServerPath>: 000000 00000000 00000000 00 0000 (0.0., usr/servers/defaultServer). -
<KeyAlias>: 0000000000 0000 0000 00 000000 (00000000: default). - <KeystorePassword>: 0000000000 00 0000000000
(00000000: KeystorePass 0000 00000000 0000 0000 0000). - {aes}ABC123...: server.xml 00 0000000000 0000000000.

```

2. □□□□□□□□ □□□□□□□□ □□ □□□□□□□□ □□□□ □□□□□□□□

1. 000000000 000 0000000 00 0000 0000000.
2. 0000 00000000000 000000 000000 0000000.
3. 000000000000 00000000000 00000:

- ```

0 00000000: 000000 00000000 0000 00000000 00000000 00000000.
0 0000: 00000000 00 server.xml 00 0000 000 000 00000000 (0.0., jdbc:oracle:thin:@//host:port/service).
0 0000000000: 00000000 00 <dataSource> 00 0000000000.
0 0000000000: 0000 1 00 0000 000000000000 0000000000 000000 00000000.

```

□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □

- 0 00000000 00000000: 000000 00000 0000000000 00000 (key.jks 00 key.p12) 000 00000 0000000000 00000 0000000000. 000000000,000 000000000 00 resources/security 00000000000 00000 000000 0000.
- 0 0000000000: 0000000 00000 00000 00000000000 0000000000 00 000000 0000 000000 00 00000000. 00000 00000000000 000000-0000 00000 0000000 00000000 00000 00000 00000 00000 00000 00000.
- 0 000000 00000: 00000 0000000 00000 00000000000 0000000000000000 00000 00000,00 000000000 00000000000000000000 00 00000000 00000000 00 00000000000 00 0000000000 000000 00 0000000000 000000 00 00000 0000000000 00000000.

□□□□□□□□ □□□□□□□□ (□□□ □□□□□□□□□□)

000000000000 00000000 0000 000, 00 server.xml 0000 0000000000 00000000 00 000000000000 000000-0000  
 0000000000 00 00000000 00000000 (00000000 00 00000 000000). 00000000 00 00000000 00000000 0000 000000-00000 00000-  
 00000 00 0000000000 00000 0000 00000000. 00000 00000 000000000000 0000000000 0000 0000000 00000000.

```
<!-- Example: Plain-text password (avoid in production) -->
<dataSource ...>
 <properties password="plaintextPassword" />
</dataSource>
```

[illegible]

0000000000 000 00000000 00 00000000 0000000000 00 000000 000000000 000000 00 00000, 00000000000 00000000 00000000  
 00 server.xml 00000 00000 000000000 00 00000 000000000000000000 00 0000 000000 0000, 0000000000000000 0000 00000000  
 00000 00000000 0000000000000 00000 000-000000000000 000000000 000000000 0000, 0000 0000000 0000000 00000000. 0000 000-  
 00000 server.xml 00000 00 0000000000000 000000000 00 000000000 000000, 000000000000 0000000000 00 00000000 000000, 0000  
 00000000000 0000 00 0000000000000000000 00000000000 0000000 000000000 000000 0000. 0000000 0000000000 000000 0000:

0000 1: server.xml 000000000000 00 00000000 00 0000

0000000000 00000000 00000000 0000, 0000000000 00000000000000 server.xml 0000 0000 <dataSource> 00000000 00  
 00000000 00000000 00 0000 0000 0000. 000 00000000 0000 00000000 000000000000 00 00000000 000000 000, 0000000000  
 00000000 0000000000 00 0000 00000000 <properties.oracle> 00000000 00 00000000. 00 00000000 000000000000000000 000  
 00 000000 000:

```

<dataSource jndiName="jdbc/myOracleDS">
 <jdbcDriver libraryRef="OracleLib"/>
 <properties.oracle url="jdbc:oracle:thin:@//localhost:1521/orcl" user="scott" password="{aes}encrypted_password">
</dataSource>
<library id="OracleLib">
 <fileset dir="${server.config.dir}/lib" includes="ojdbc6.jar"/>
</library>

```

000000: - url: 00000000 0000000000 00 0000 0000 0000 (0.0., jdbc:oracle:thin:@//localhost:1521/orcl). - user: 0000000000 0000000000 (0.0., scott). - password: 0000 00 0000000000 000000000000 0000000000, {aes} 00 000000 00000000 (0.0., {aes}encrypted\_password). - <jdbcDriver>: 00000000 000000 00000000 0000 000000 00 000000000000 000000 0000.

0000000000 0000 00 00000000000000 00000000 0000 0000 0000000000 0000000000-0000000000 000000 00000000 00 00000000 000000 0000 000000 (0.0., 000000 00000000 00 0000000000), 00000000 00000000 0000000000 0000000000 000000 000000 0000, 000000 000000000000 0000000000 00 0000 000000 0000.

000000 **2:** server.xml 00 00000000000000 00000000 00 00000000 00000000

000000 00000000 0000000000 00 0000000000000000 <dataSource> 00000000 00 server.xml 0000 0000 00000000 00000000. <properties.oracle> 0000000000 00 0000 00000000: - 000000 0000: url 000000000000 0000 000000 0000 (0.0., jdbc:oracle:thin:@//localhost:1521/orcl). - 000000000000: user 000000000000 0000 000000 0000 (0.0., scott). - 000000000000 0000000000: password 000000000000 0000 000000 0000 (0.0., {aes}encrypted\_password).

0000 0000 00000000 000000 0000 00 00000000 0000000000 00 000000 000000 00000000 00000000, 000000000000 00 00 00 00000000 000000: - jdbc:oracle:thin:@//hostname:port/service\_name (0000000000 000000 00 0000 000000 0000) - jdbc:oracle:thin:@hostname:port:SID (0000 00 0000 000000 0000)

000000 server.xml 00 000000 00000000 000000 0000 00 0000000000 000000 00 000000.

000000 **3:** 0000-000000000000 0000000000 00 00000000 00000000

server.xml 000000 000000000000 0000 00 0000000000 000000000000 0000, {aes} 00000000 00 000000 000000000000 000000 000000 0000. 000000000000 0000000000 0000000000 000000000000 00 0000000000 00 000000 0000000000 000000 0000, 00000000 0000000000 0000 00000000 000000 0000000000 00 000000000000 00000000 00 000000 0000000000 000000 0000. 00000000 000000000000 0000000000 00 000000:

1. 000000000000 00 securityUtility 000000 00 0000 00000000:

0 0000 000000 000000 000000000000 000000000000000000 00 000000 000000 0000, 000000000000 bin 000000000000 000000 (0.0., <liberty\_install\_dir>/bin/).

0 0000000000 00 0000000000 00000000 000000 bin 000000000000 00 0000 0000000000 0000 00000000:

securityUtility decode --encoding=aes <encrypted\_password>

<encrypted\_password> 00 00000000 000000000000 00000000 00 0000000000 00000000 password 000000000000 00 (0000 000000 {aes} 00 000000). 0000 0000000000:

```
securityUtility decode --encoding=aes encrypted_password
```

□ 0000 00000 0000 000000000 00 0000000 0000000.

## 2. 00000000000000:

□ 0000 00000 0000 0000000000 00000000 00000000000000 00 securityUtility 0000 00 0000000 0000 0000,  
00 000000 000000 0000 0000000000 00 000000 00000000 00000000000000 00 00 0000000 00 00000000 000000  
000000 000 00 00000 0000000 00 00000000000 0000 0000.

0000 0000 00000000 0000000000 000000000, 0000000 000000 0000 0000 000000000 000 00000 000 000000 000000 0000.

0000 4: 0000000 0000 0000000 00 0000000000 000 0000 0000000000 0000000

0000000000 000 00 0000000 0000 0000000 00 000000 000000000 000000 00 0000 000000000 0000. 000000 00000 0000000 000  
00000 000000000 00 server.xml 00 <library> 000000000 00000 00000000000 00000 00000 0000 (0.0., ojdbc6.jar).

## 1. 0000000 000 00 00000000 00000000:

□ server.xml 00 <fileset> 000000000 00000 000000000 00 00000 0000000 00000 0000000 0000 00000 00 0000000  
00000000 (0.0., ojdbc6.jar 00 \${server.config.dir}/lib).

□ 00000 000000 00000 00000 0000, 00 0000000 00 000000000 00 00000000000000 00000000 0000000000 00000000 (0.0.,  
ojdbc6.jar 00 ojdbc8.jar, 0000000000 000000 0000000000 000000000).

## 2. 0000000 00 0000000000 000 00000 000 0000000:

□ 0000000000 000 00 00000 00000000.

□ 00000 00000 000000000 0000 0000 000000 00000000.

□ + 00000000 000 000000 00000000 00000 00000000 0000 0000000 00 000000.

□ 00000000 00 000000000000 00000000:

- 00000: 00 00000 000000 00000000 (0.0., "00000000 00000 00000000").
- 0000000000 0000: 00 0000000 0000 000000 00000000 (0.0., jdbc:oracle:thin:@//localhost:1521/orcl).
- 0000000 000000: oracle.jdbc.OracleDriver 0000000 000000000.
- 0000000 000000 000000: 0000 0000 0000000 00000000, 00000 00000000 00000 00000000 0000 000000 00 00000000 0000  
00000000 00000000.

□ 00 0000 0000000 00000000 00000000 00 00000 000000 00 00000.

0000 5: 0000000000 0000 00000 00000000000 (0000000) 00000000 00000000

00, 000000000000 0000000000 00 0000 0000000 00000 00000000000000 0000000 00000000 00000000:

## 1. 00000 000000 000 00000000:

□ 0000000000 0000 00000 000000000 0000 0000 000000000.

□ + 00000000 000 000000 00000000 00000 0000000 0000 0000000 00 000000.

□ 000000 00 0000000000 0000000:

- 0000: 0000000000 00 0000 00 0000 000000 000000 (0.0., "000000 00 000 0000000000").
- 0000000: 00000 0000000000 00 000000 0000 0000000 0000000 0000000.
- 000: server.xml 00 <properties.oracle> 00000000 00 0000 000 000000 0000000 (0.0., jdbc:oracle:thin:@//loc
- 0000000000: server.xml 00 0000000000 000000 0000000 (0.0., scott).
- 0000000000: 0000 3 00 0000 00000000 000000 0000 0000000000 000000 0000000.

## 2. 0000000000: 000000000000 00000000000:

□ 0000 <properties.oracle> 00000000 0000 server.xml 0000 000000000000 000000000000 0000 (0.0., ssl="true" 00 connectionTimeout="30000"), 000000 0000000000000000 0000 000000000000 000 000 000000 00000000 000 0000 000-000000 000000 00 0000 0000 000 0000000.

## 3. 000000000000 00 0000 0000000:

□ 0000 000000000000 000 000000 0000000 0000000 000000 00 0000 00 0000000000 000 0000000000 00 0000000000 000 0000000000 000.

□ 0000 000000000000, 00 000 000000 0000000 000000 00 0000 000000 00 00000.

0000 000

0000000000 000 0000 000 000000 00 0000 000 server.xml 0000 0000 <properties.oracle> 00000000 00 url 000000000000 0000 0000000000 0000 0000 000. 000 00000000: - 0000 server.xml 0000 <properties.oracle url="jdbc:oracle:thin:@//localhost:1521/orcl" .../> 0000, 00 0000 00000000: jdbc:oracle:thin:@//localhost:1521/orcl 000 000, 0000000000 000 00000000 0000000000 00 000000, 0000000000 000 00 0000000 0000000000 00 00000000 000000 00000 0000.

000000

□ 000000000000 0000000000: 000 0000000000 000 0000 00000000000 0000000000 00 0000000000 000 0000 000 000000; 0000 000000 0000 0000 0000000 000000 000000 0000.

□ 000000000000 000000: server.xml 0000 <dataSource> 00000000000 0000000000 0000 000 000000 00 0000 00000000-000000 00 0000 0000000 0000 0000 000. 0000000000 000, 00 0000000000 0000000 000, 0000000 0000000 0000000-0000000000 00000000 00 00 0000000000 00 0000000 000000 0000.

□ 000 0000000000000000: 0000000 00 0000000 0000 0000000 000 000000000 000000 0000000000 00000000 00 000000 000000 00000 000000000000000000 00 00000.

000 000000 0000000 000000, 000 0000000000000000 0000000000 000 00 0000000 0000000000 00 000000000 000 000000 00000, 00 00000000000000 0000000 00 0000 000 000000000000 0000000000 000000000 00 server.xml 0000.

securityUtility 00000000 00 0000000000 00000000 000-0000000000 0000000000 (0.0., 000000 000-000000 0000 {aes} 00 0000 server.xml 0000) 00 0000 0000000 0000000 0000 00000000 000000, 000000 000-0000000000 0000000000 00 0000 000000 000000 0000 0000 00000000 000000 00 0000. 000000000000 0000000 com.ibm.websphere.crypto.PasswordUtil 000000 00 0000 000000 0000, 00 00000000 00000000 00 000000 0000 0000 0000000000 00 00000000 0000 000000 0000. 000000 00 00000000000 0000000000 0000 0000 0000000000 000000 00 00000:

0000000 **1:** 000 0 00000000000 000 00000000000000 (00000000000000)

00000000 0000000 0000 00 00000000 0000 00000000000000 00000000 000000, 0000 0000000000 000000 00 00000000 00000000-000000 00000 0000 0000 000000 00000, 00000 00000000 000000 0000 00 000000000 000000000000 0000 (0000000000 00 00000000, 00 server.xml 00000 00000000 00000 00000 0000) 00 00000000 0000.

0000000:

1. 00000000 0 0000 00000 00 00000 0000000000 00 decode.jsp 0000000 00000, 00 000000000 00 0000000:

```
<%@ page import="com.ibm.websphere.crypto.PasswordUtil" %>
<%
String encoded = request.getParameter("encoded");
if (encoded != null) {
 try {
 String decoded = PasswordUtil.decode(encoded);
 out.println("Decoded password: " + decoded);
 } catch (Exception e) {
 out.println("Error decoding password: " + e.getMessage());
 }
}
%>
```

2. 00000000 0000 0000

0 decode.jsp 00 00 0000 000000000000 000000000000 00000 000000 0000000, 000000 wlp/usr/servers/yourServer/apps/m  
0 00000 00000000000 0000, 00 0000000 0000 000000 0000000000 00 0000 00 0000000 0000 0000000000 000000 0000000000 00  
dropins 000000000000 00000 00000 0000000 00000000.

3. 00000000 0000 0000

0 000000000 00000000 00 0000000 00000000 (server start yourServer).  
0 000000000 00000 00000000 0000 00000000000 00000000: http://localhost:9080/myApp/decode.jsp?encoded={aes}your\_en  
{aes}your\_encrypted\_password 00 server.xml 00 00000000 0000000000000 00000000000 00 0000000000 00000000.

4. 00000000000 0000 0000000000 00000000000 00000 0000000000 0000000000 00 0000000000 00000000, 00 0000 0000 0000 000000  
00000 (0.0., 00000000000 0000 00000 00000000000 00 0000000000 000000 00 00000).

5. 0000000 000 000000000000 00000000 00000 00 0000, 000 00 000000 00 000000 00 00000000 000000 0000-00000000 000 00 0000.

000 0000 00000: 000000 0000 000 00000 000000 00000 000 00 PasswordUtil.decode() 0000 00000000000 000 (00000000 00 0000000, 00 wlp.password.encryption.key 00 00000000 server.xml 0000 00000000 0000 0000 0000) 00 000 000000 000 00 000000000 00 0000000 00000 00 0000 000 0000 0000 0000 000.

---

0000000 **2:** 000 0 00000000000 0000 00000000

0000 000 000000000000 0000000 000000 000000000 0000 000, 00 000 00 00000000000 0000 00000000 0000 000000 0000 000000 00000000 00000000 00 000000 0000000000 0000 000 0000000. 000 000000000 0000000 000 0000000 0000 00000000 0000000000 0000 00 000000000 0000 000, 00000000000 0000 0000000 000 000 00000 00000 000.

0000000 00000:

```
import com.ibm.websphere.crypto.PasswordUtil;

public class PasswordDecoder {
 public static void main(String[] args) {
 if (args.length < 1 || args.length > 2) {
 System.out.println("Usage: java PasswordDecoder <encoded_password> [crypto_key]");
 return;
 }
 String encoded = args[0];
 String cryptoKey = args.length == 2 ? args[1] : null;
 try {
 String decoded;
 if (cryptoKey != null) {
 decoded = PasswordUtil.decode(encoded, cryptoKey);
 } else {
 decoded = PasswordUtil.decode(encoded);
 }
 System.out.println("Decoded password: " + decoded);
 } catch (Exception e) {
 System.err.println("Error decoding password: " + e.getMessage());
 }
 }
}
```

000000:

### 1. 00000000 000 00000000

0 0000 00 PasswordDecoder.java 00 0000 0000 0000 0000000.

0 00000000 00 0000 00 000000 00000000 0000000:

```
javac -cp /path/to/wlp/lib/* PasswordDecoder.java
```

/path/to/wlp 00 000000 00000000 00000000000000 0000000000 00 00000000 0000000 (0.0., /opt/ibm/wlp).

### 2. 000 000 00000000

0 0000 000000000 00000000 000 00 00000000 0000000000 000:

```
java -cp /path/to/wlp/lib/*:. PasswordDecoder "{aes}your_encrypted_password"
```

0 0000 0000000 000 000 0000 0000 000 (0.0., server.xml 0000 0000000 0000 0000 000 00 <variable name="wlp.password.encryption.key" value="yourKey"/>):

```
java -cp /path/to/wlp/lib/*:. PasswordDecoder "{aes}your_encrypted_password" "yourKey"
```

### 3. 0000000 000 0000000 00000000 000000 0000 000000000 00 000000 000000 0000000 0000 000 00000000000 000.

000000:

0 00000000 0000 (0.0., 00 wlp/lib) 0000 com.ibm.websphere.crypto.PasswordUtil 000 0000 00000000000000 0000.

0 0000 0000000 000 000 0000 0000 000 000 0000 00000000 0000 00000, 00 000000000 0000 0000. server.xml 00 00000000 00 0000 0000000000000000 000000 0000 000 00 000000 0000000.

000 0000000000000000

0 00000000 00.000000 000:

- 0000 server.xml 0000 wlp.password.encryption.key 00000000 0000 0000 0000, 00 00000000 000 000 00000 000 000 PasswordUtil.decode(encoded) 0000000 00 000000 (0000000 2) 0000 000000000000 0000000-00000 00 0000 0000000.

- 0000 0000000 000 0000000 00000 0000 000, 00 00000 000000000000 000000 000000 0000 0000 00000000 00000 000000 000 (0000000 2) 00 0000000 00000000 000 0000 000000 000000 000 (0000000 1).

0 000000000: 00000000 000000000 00 00000000000 0000000 0000000, 0000000 000 000000 00000 0000 00000. 00000000 00 0000000000000000 0000000 000000 00 00000000000.

0 000000000000: 0000000 1 (000 0000000000000) 0000000000 000000 000000000 000 0000000 000 00000000 0000-000000000 0000 000 0000 000, 0000000000000000 000 000 00000000000000 00 0000000 000000 000.

000 000000 0000000 000000, 000 000000000000000 server.xml 00 000-000000000000 000000000 00 0000000 000 000000 00000, 00000000 securityUtility 0000 00000000 0000000 00 00000000 00.