

Advanced Interface Methods

ORACLE

Copyright © Oracle Corporation, 2004. All rights reserved.

Objectives

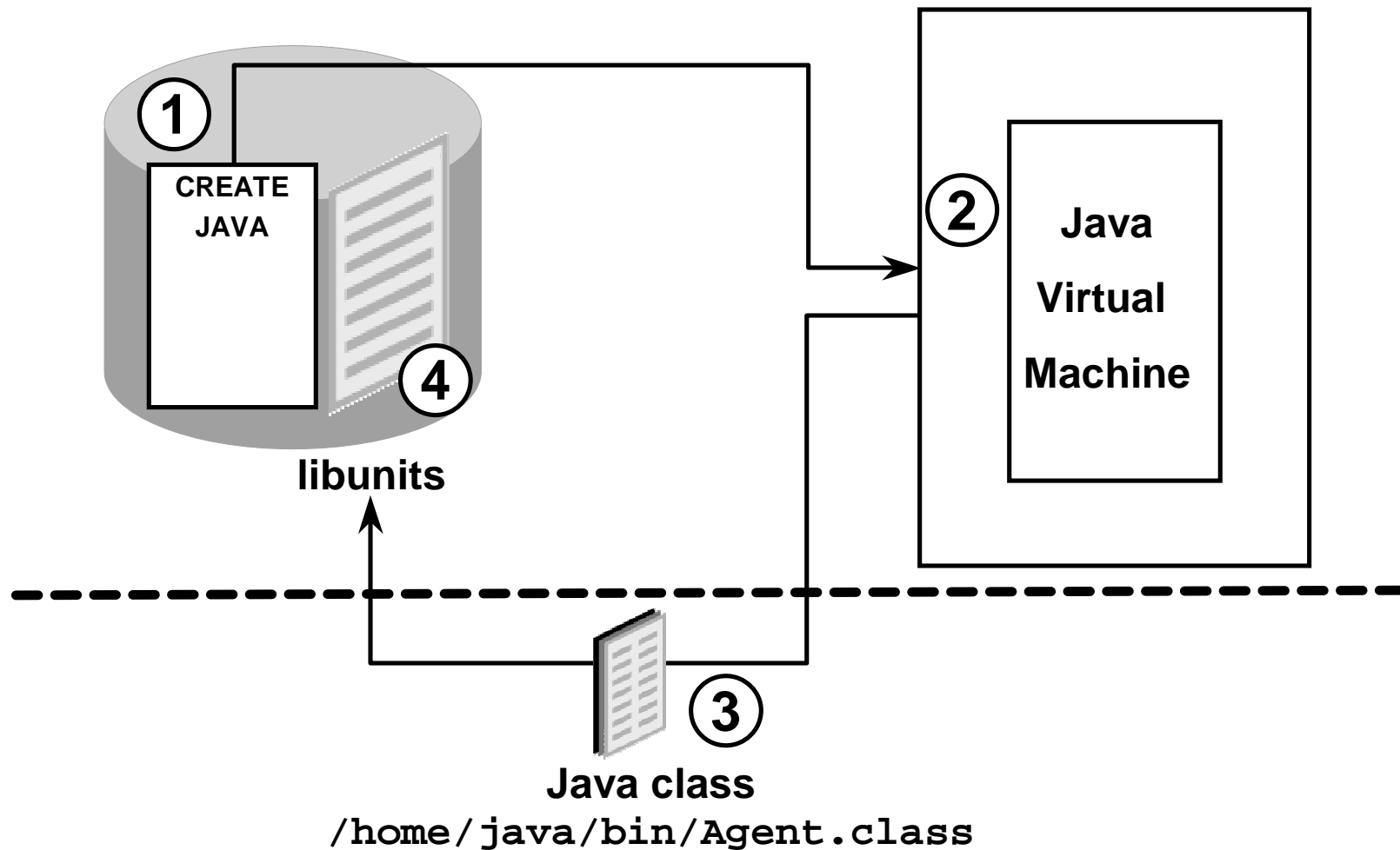
After completing this lesson, you should be able to do the following:

- **Write PL/SQL programs that call Java class methods**

Overview of Java

- **The Oracle database can store Java classes and Java source**
- **Stored in the database as procedures, functions, or triggers**
- **Run inside the database**
- **Manipulate data**

How PL/SQL Calls a Java Class Method



ORACLE

Copyright © Oracle Corporation, 2004. All rights reserved.

Development Steps for Java Class Methods

- 1. Upload the Java file.**
- 2. Publish the Java class method by creating the PL/SQL subprogram unit specification that references the Java class methods.**
- 3. Execute the PL/SQL subprogram that invokes the Java class method.**

Loading Java Class Methods

1. Upload the Java file.

- At the operating system use the `loadjava` command line utility to load either the Java class file or the Java source file.

- To load the Java class file, use:

```
>loadjava -user oe/oe Factorial.class
```

- To load the Java source file, use:


```
>loadjava -user oe/oe Factorial.java
```

- If you load the Java source file, you do not need to load the Java class file.

Publishing a Java Class Method

1. Publish the Java class method by creating the PL/SQL subprogram unit specification that references the Java class methods.
 - Identify the external body within a PL/SQL program to publish the Java class method.
 - The external body contains the name of the Java class method.

```
CREATE OR REPLACE
{  PROCEDURE procedure_name [(parameter_list)]
  | FUNCTION function_name [(parameter_list]...)]
  RETURN datatype}
  regularbody | externalbody
END;
```



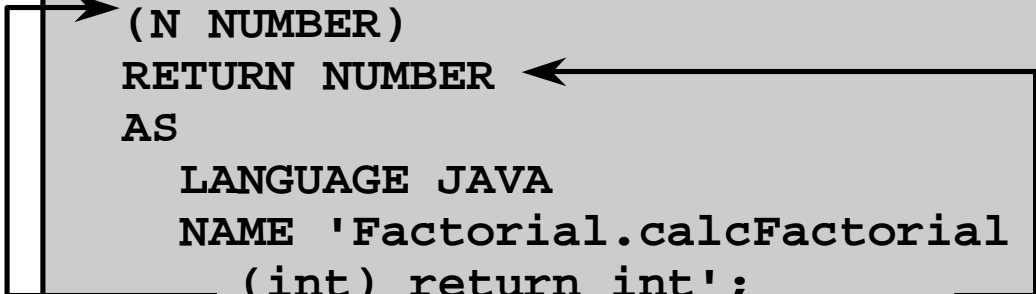
```
{IS | AS} LANGUAGE JAVA
  NAME 'method_fullname (java_type_fullname
    [, java_type_fullname]...)
    [return java_type_fullname]';
```

ORACLE

Publishing a Java Class Method

- **Example**

```
CREATE OR REPLACE FUNCTION plstojavafac_fun  
  (N NUMBER)  
  RETURN NUMBER  
AS  
  LANGUAGE JAVA  
  NAME 'Factorial.calcFactorial  
        (int) return int';
```



- **Java method definition**

```
public class Factorial {  
  public static int calcFactorial (int n) {  
    if (n == 1) return 1;  
    else return n * calcFactorial (n - 1) ;  
  }  
}
```

Executing the Java Routine

1. Upload the Java file.
2. Publish the Java class method by creating the PL/SQL subprogram unit specification that references the Java class methods.
3. Execute the PL/SQL subprogram that invokes the Java class method.

Creating Packages for Java Class Methods

```
CREATE OR REPLACE PACKAGE Demo_pack  
AUTHID DEFINER  
AS
```

```
    PROCEDURE plsToJ_InSpec_proc  
→   (x BINARY_INTEGER, y VARCHAR2, z DATE) ←  
END;
```

```
CREATE OR REPLACE PACKAGE BODY Demo_pack  
AS
```

```
    PROCEDURE plsToJ_InSpec_proc  
      (x BINARY_INTEGER, y VARCHAR2, z DATE)  
    IS LANGUAGE JAVA  
    NAME 'pkg1.class4.J_InSpec_meth  
→      (int, java.lang.String, java.sql.Date');
```

ORACLE

Copyright © Oracle Corporation, 2004. All rights reserved.

Summary

In this lesson, you should have learned how to:

- **Use Java methods and call them from your PL/SQL programs.**

Practice Overview

This practice covers the following topics:

- **Write programs to interact with Java code**