



---

# Client/Server-Programmierung

WS 2019/2020

Roland Wismüller  
Betriebssysteme / verteilte Systeme  
roland.wismueller@uni-siegen.de  
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 17. Januar 2020



---

# Client/Server-Programmierung

WS 2019/2020

## 4 RMI / IIOP



### Inhalt

➔ RMI über IIOP

➔ Farley, Crawford, Flanagan: S. 88ff, Kap. 7

➔ <https://docs.oracle.com/javase/8/docs/technotes/guides/rmi-iiop>



### RMI über IIOP

➔ Ziel:

➔ Verbindung von Java-RMI Objekten mit nicht-Java (CORBA) Objekten

➔ Vorgehensweise:

➔ RMI nutzt CORBA IIOP-Protokoll zur Kommunikation zwischen Objekten

➔ Damit:

➔ Java-Client kann CORBA Objekt über RMI nutzen

➔ CORBA-Client kann RMI-Objekt über CORBA nutzen



### Vorgehen zur Nutzung von RMI/IIOP

- ➔ **Basisklasse:** *Remote*-Objekte müssen von `javax.rmi.PortableRemoteObject` erben statt von `java.rmi.server.UnicastRemoteObject`
- ➔ **RMI Compiler:** *Stubs* und *Skeletons* müssen über `rmic -iiop` erzeugt werden
  - ➔ Option `-idl` erzeugt zusätzlich OMG IDL Datei
- ➔ **Namensdienst:** statt RMI *Registry* muß JNDI benutzt werden, um auf CORBA-Namensdienst zuzugreifen
- ➔ **Downcast:** vom Namensdienst erhaltene Objektreferenzen müssen über `PortableRemoteObject.narrow()` in gültige Referenzen umgewandelt werden



### Beispiel: „Gemischtes“ *Hello World*

- ➔ Server mit RMI realisiert
  - ➔ Schnittstelle:

```
import java.rmi.Remote;  
import java.rmi.RemoteException;  
  
public interface Hello extends Remote  
{  
    String sayHello(String to) throws RemoteException;  
}
```
- ➔ Client wahlweise mit RMI oder CORBA



### Server-Code

```
import javax.naming.*;
import java.rmi.*;
import java.util.*;
import javax.rmi.PortableRemoteObject;

public class HelloServer extends PortableRemoteObject
    implements Hello {

    public HelloServer() throws RemoteException {
        super();
    }

    public String sayHello(String to) {
        System.out.println("Hello-Server called");
        return "Hello World to " + to + "!";
    }
}
```



### Server-Code ...

```
public static void main(String args[]) {
    try {
        // Server-Objekt erzeugen
        HelloServer obj = new HelloServer();

        // Referenz über JNDI beim Namensdienst registrieren
        Properties props = new Properties();
        props.put("java.naming.factory.initial",
            "com.sun.jndi.cosnaming.CNCtxFactory");
        props.put("java.naming.provider.url",
            "iiop://bspc02:5555");
        Context ctx = new InitialContext(props);
        ctx.rebind("HelloWorld", obj);
    }
    catch (Exception e) { ... }
} }
```



### RMI Client

```
public static void main(String args[]) {
    try {
        // Obektreferenz über JNDI vom Namensdienst besorgen
        Properties props = new Properties();
        props.put(Context.INITIAL_CONTEXT_FACTORY,
            "com.sun.jndi.cosnaming.CNCtxFactory");
        props.put(Context.PROVIDER_URL, "iiop://bspc02:5555");
        Context ctx = new InitialContext(props);
        Hello obj = (Hello)
            PortableRemoteObject.narrow(ctx.lookup("HelloWorld"),
                Hello.class);

        // Remote Methode aufrufen
        System.out.println(obj.sayHello("Roland"));
    }
    catch (Exception e) { ... }
}
```



### CORBA Client

```
public static void main(String args[]) {
    try{
        // ORB erzeugen und initialisieren
        ORB orb = ORB.init(args, null);

        // Root naming context holen
        org.omg.CORBA.Object ns =
            orb.resolve_initial_references("NameService");
        NamingContextExt ncRef = NamingContextExtHelper.narrow(ns);
        // Objektreferenz vom Namensdienst besorgen
        NameComponent path[] = ncRef.to_name("HelloWorld");
        Hello helloRef = HelloHelper.narrow(ncRef.resolve(path));

        System.out.println(helloRef.sayHello("Peter"));
    }
    catch(Exception e) { ... }
}
```

# Client/Server-Programmierung

WS 2019/2020

15.11.2019

Roland Wismüller  
 Betriebssysteme / verteilte Systeme  
 roland.wismueller@uni-siegen.de  
 Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 17. Januar 2020

## 4 RMI / IIOP ...



### Code-Erzeugung mit RMI-Server / CORBA-Client

