WebSphere® and CICS Transaction Server

The future of SOA on z/OS built on a smarter foundation of CICS TS V4, WAS V7 and Rational Developer for System z

Steve Kinder,
Ian Mitchell,
Colette Manoni,
Cindy Krauss

April 8th, 2009

Welcome...

- Your speakers today:
  - Steve Kinder – SOA Foundation Architect
  - Ian Mitchell – Product Architect for CICS Transaction Server
  - Colette Manoni – Product Architect, WebSphere Application Server for z/OS
  - Cindy Krauss – Product Architect, Rational Developer for System z

- What you will hear in this session:

*Building on the lessons learned in **Future-Proof your Applications with SCA: Programming model optimized for SOA** we will show you how SCA can be applied to your business applications running on z/OS.*

Using a simple application example we will show components wired across CICS and WebSphere Application Server (WAS) that will be used as an illustration of how CICS, Rational Developer for System z (RDz), and WebSphere Application Server for z/OS are delivering new capabilities in support of this important multi-lingual, multi-environment programming model.

Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- Summary and Questions

Agenda

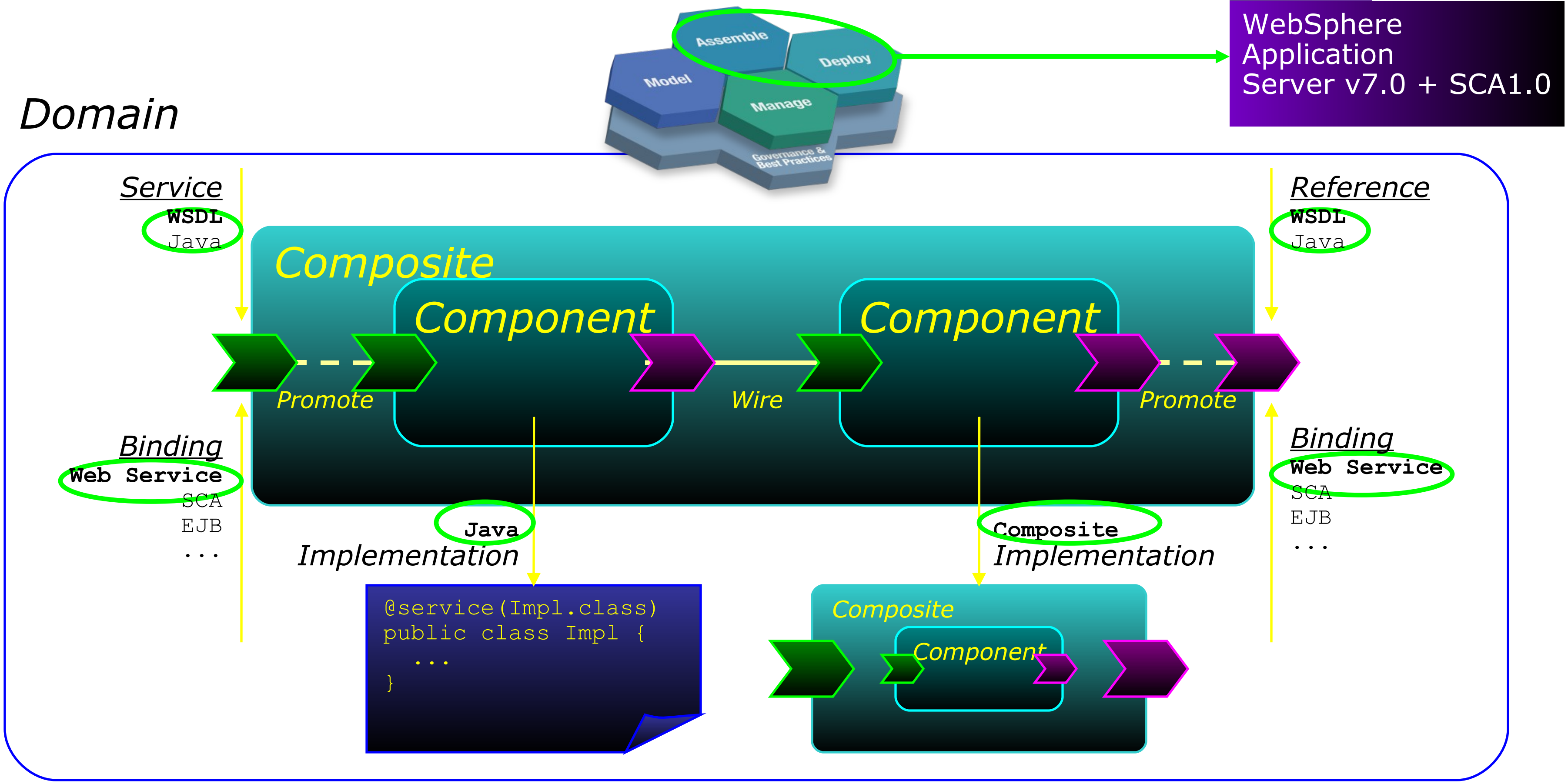
- **Quick SCA in WAS recap**
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- Summary and Questions

SCA: What it is

- Service Component Architecture.
- A concrete manifestation of an SOA way of thinking.
- Designed for building agile service oriented applications.
- A framework for implementing, assembling, composing and deploying services.
- Supports loose or tight coupling of coarse or fine grained services.
- Extends, exploits and complements existing technologies and standards.
- Language, Application Environment, Framework and Vendor neutral.
- Supports Java and Web Services, and more
- An extensible set of:
  - Protocol bindings (eg. SCA, WS, RMI, …)
  - Implementation languages (eg. Composite, Java, …)
  - Interface definitions (eg. WSDL, Java, …)
  - Pluggable Data bindings (eg. PoJo, JAXB, …)
  - Policies and Intents (eg. Integrity, Confidentiality).
- “Classic SCA” refers to Service Component Architecture as it is defined and built by IBM supported in a variety of WebSphere Family products starting with V6.
- “Open SCA” refers to Service Component Architecture as defined by the industry at both the OSOA collaboration

SCA Key Concepts

Design → Implement → Compose → Run → Test



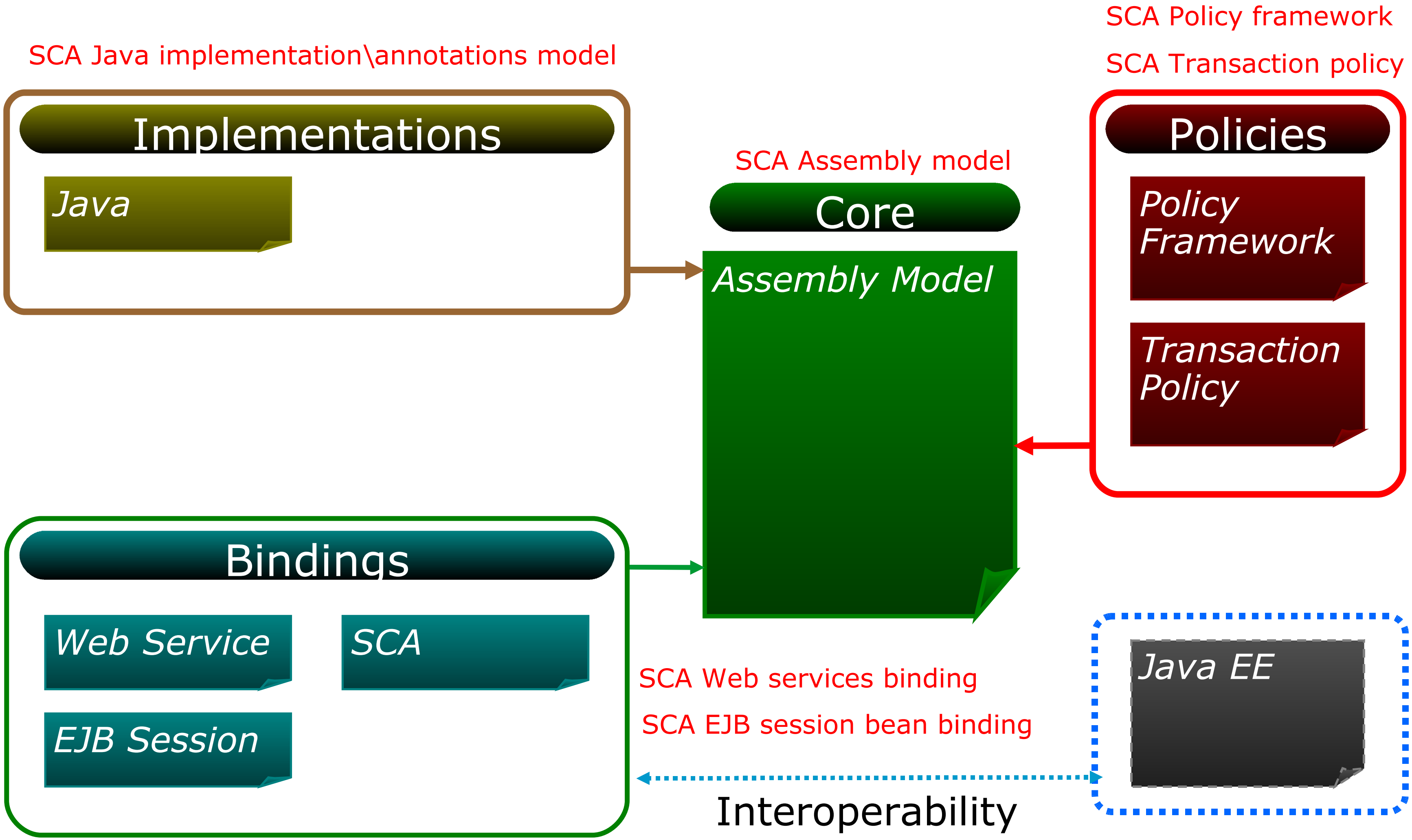
WebSphere software

WebSphere Application Server v7.0 + SCA1.0

```
@service(Impl.class)
public class Impl {
  ...
}
```

Reusability, Connectivity, Flexibility, Extensibility

SCA v1.0 Specifications – Flexible & Extensible



SCA Java implementation\annotations model

SCA Policy framework

SCA Transaction policy

SCA Assembly model

SCA Web services binding

SCA EJB session bean binding

Agenda

- Quick SCA in WAS recap
- **Introduction to SCA support in CICS TS v4.1**
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- Summary and Questions

CICS Transaction Server v4.1 allows you to:

Compete *for new opportunity by gaining insight into business processes and responding by modifying key business applications quickly and with confidence*

– Business Flexibility and Innovation

Comply *with corporate, industry and government policies to manage business risk of critical business applications*

– Governance and compliance

Control costs *by simplifying IT infrastructure and improving development and operations productivity through easier-to-use interfaces and functions*

– IT Simplification

CICS TS v4.1 will be available 2nd or early 3rd quarter 2009.

Join the Open Beta *– see http://www.ibm.com/software/htp/cics/tserver/v41/openbeta/*

CICS TS v4.1: Competing with Flexibility and Innovation

- **Application Components**
  - **Bundles**
- Web 2.0 and RESTful features
  - **Atom Feeds**
- Web Services, SOAP and XML Enhancements
  - **WS-Addressing**
  - **Improved XML data mapping**
- **Java 6**
- **Event Processing**

CICS TS v4.1: Complying

- Event Processing
- Meta-data for Resource Definitions
- Identity Propagation
- Security Enhancements for DB2 applications
- WebSphere Service Registry and Repository support

CICS TS v4.1: Controlling Costs with Simplification

- **CICS Explorer**
- **CICS Management Interface**
- **CSD Definition Repository API**
- **Discovery Library Adapter for CICS TS**
- **Dynamic Workload Management enhancements**
- **Extensions to CICS intercommunications over TCP/IP**
- **IPv6 support**
- **WebSphere MQ Group Attach**

CICS TS v4.1 Component Architecture

- Provide capability to easily develop flexible and reusable CICS application components
  - Rapid assembly and deployment of new Services
  - Express existing applications as re-usable components

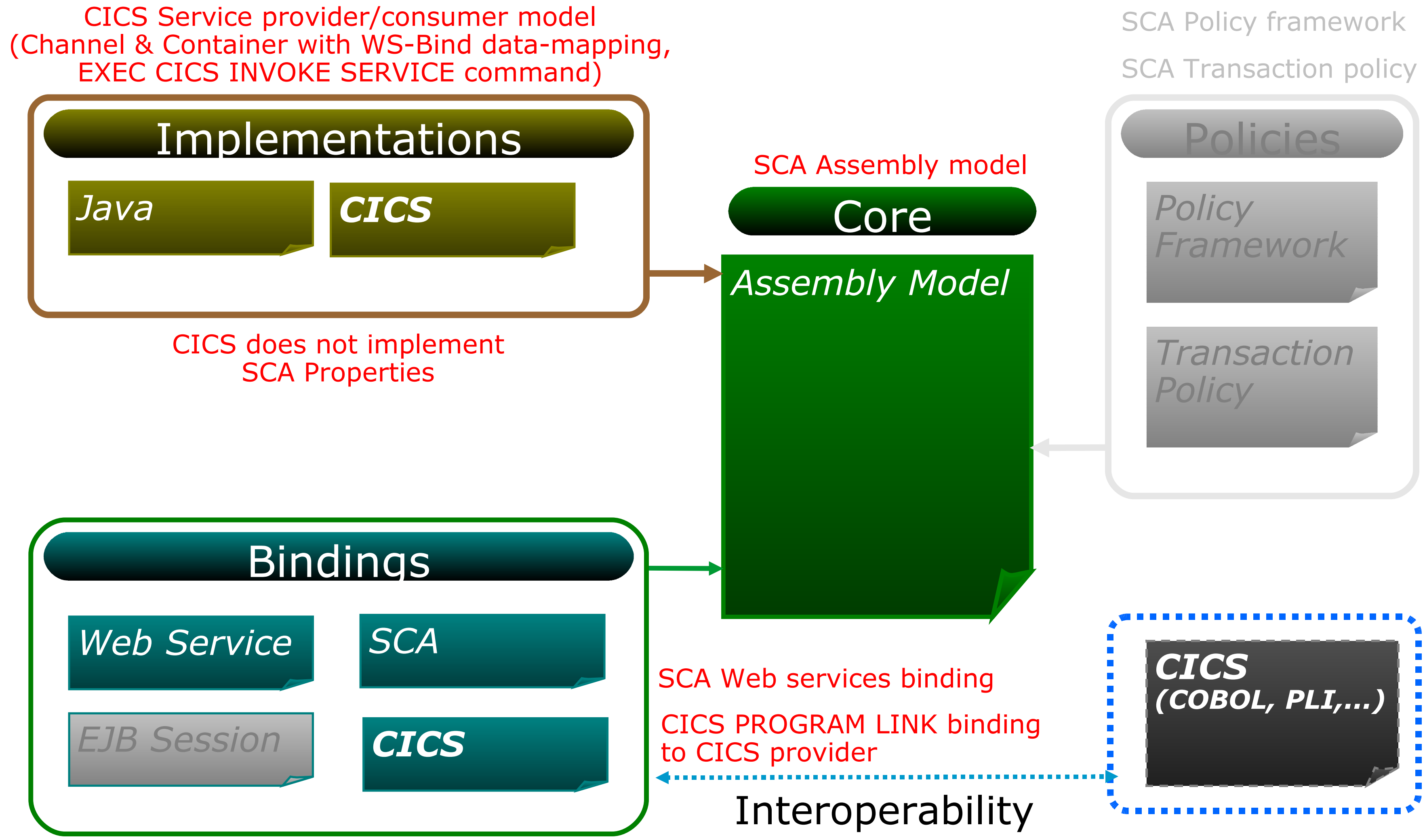
- Separation of bindings from application code allows flexible infrastructure changes

- Reduce skills and effort required to view and manage business applications

Component Architecture in CICS TS v4.1

- Ability to install and manage business applications as single CICS components
  - Abstract away from programs, transactions, resources
- Ability to describe CICS application as SCA components (using SCDL)
- Application bindings provided by CICS and configured using SCDL
  - Services and References
  - Invocation locally and via web services
  - EXEC CICS INVOKE SERVICE
- RDz providing CICS component tooling to enable component definition, assembly and deployment

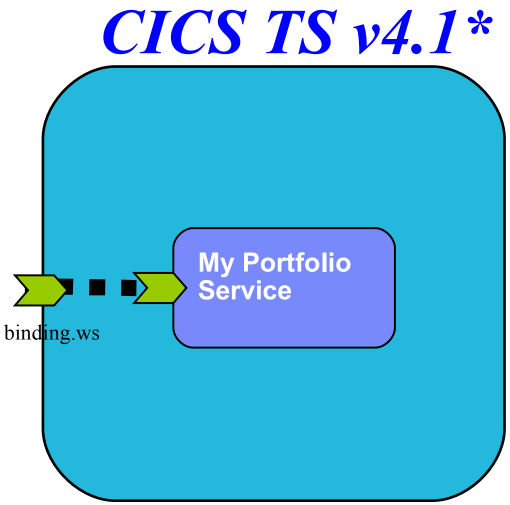
SCA v1.0 Specifications – CICS TS v4.1 capabilities



Agenda

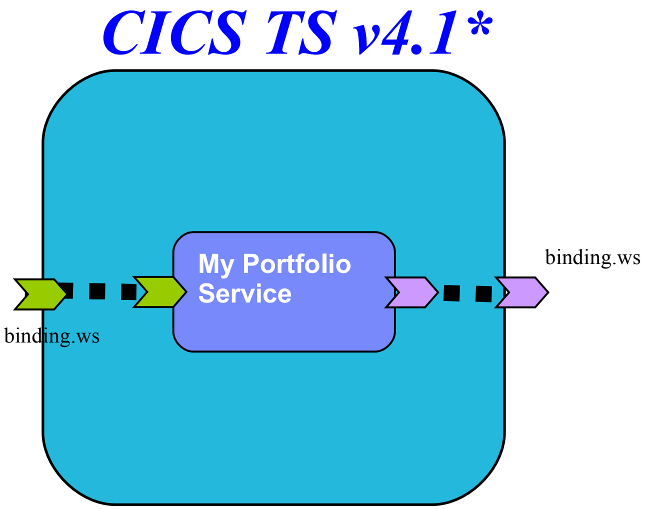
- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- **The Portfolio Scenario**
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- Summary and Questions

Service Component Architecture - Scenario



```
<component name = “MyPortfolioComponent”>
    <implementation.CICS program=”PORTFOLI”/>
    <service name=”MyPortfolioService”
        <binding.ws ... >
    </service>







</component>
```

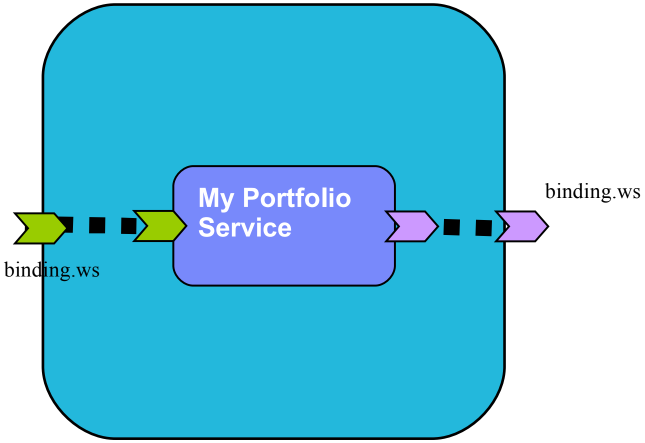
Service Component Architecture – Scenario



```
<component name = "MyPortfolioComponent">
    <implementation.CICS program="PORTFOLI"/>
    <service name="MyPortfolioService"
        <binding.ws ... >
    </service>
    <reference name="MyStockQuoteService">
        <binding.ws ... >
    </reference>
</component>
```

Service Component Architecture – Programming Model

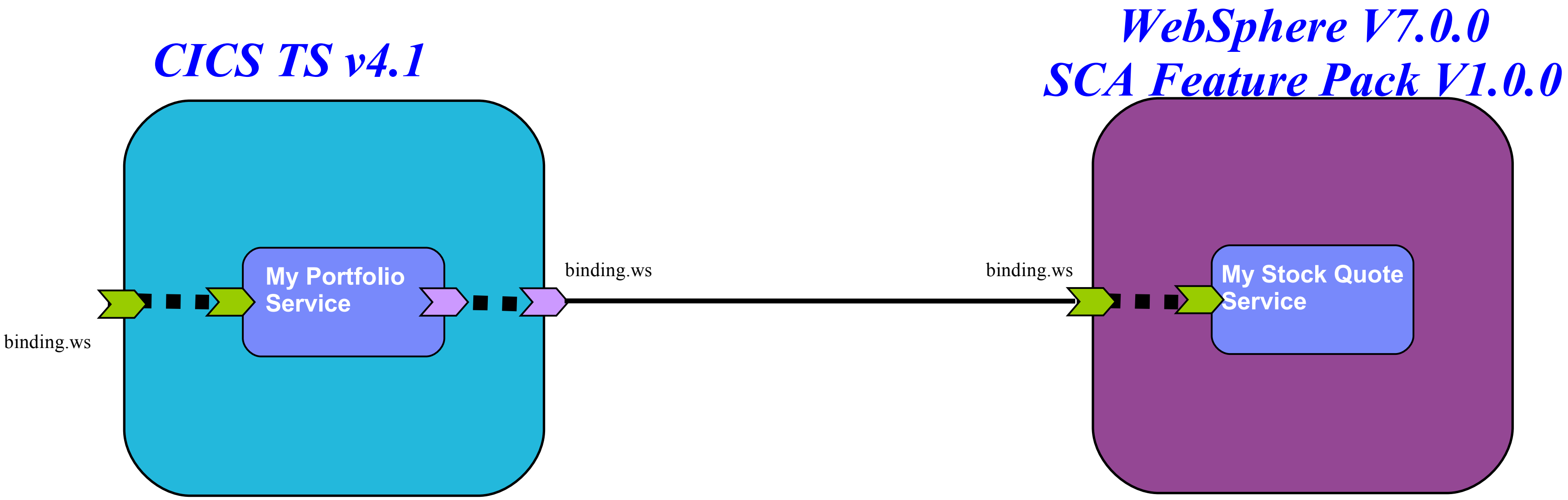
*CICS TS v4.1**



CICS COBOL Program

```
EXEC CICS PUT CONTAINER(“SYMBOL”)
              CHANNEL(“QUOTESERVICE”)
              FROM(“IBM  “)
.

EXEC CICS INVOKE SERVICE(“MyStockQuoteService”)
                 CHANNEL(“QUOTESERVICE”)
```

Service Component Architecture – Scenario



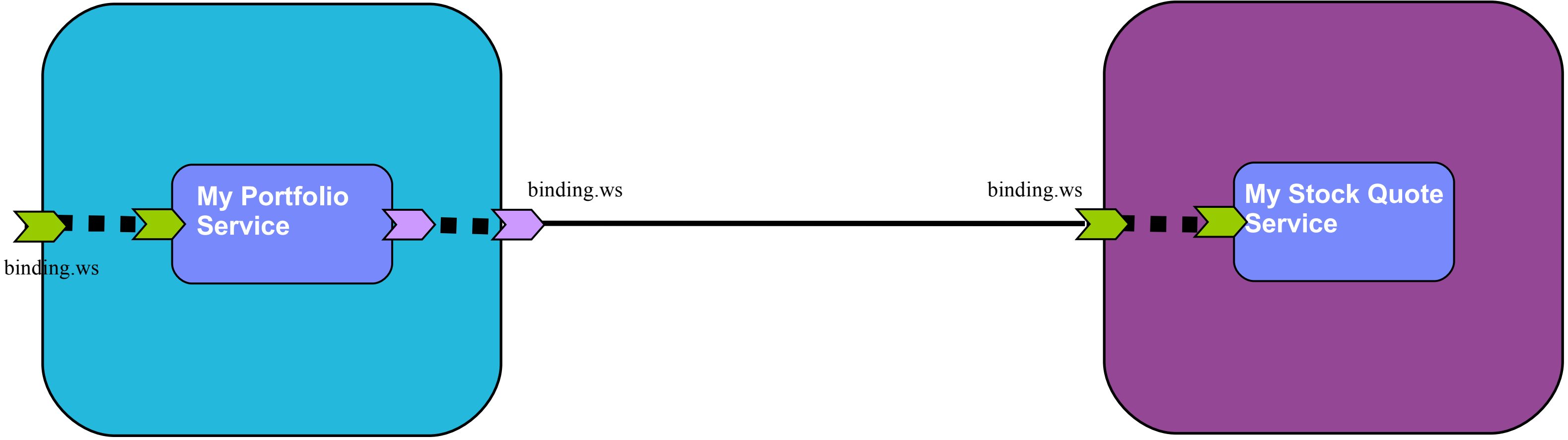
```
<component name = “MyPortfolioComponent”>
    <implementation.CICS program=”P
    <service name=”MyPortfolioServi
        <binding.ws ... >
    </service>
    <reference name=”MyStockQuoteSe
        <binding.ws ... >
    </reference>
</component>
```

```
<component name = “MyStockQuoteComponent”>
    <implementation.java
class=”mystockquoteImpl.class”/>
    <service name=”MyStockQuoteService”>
        <binding.ws ... >
    </service>
</component>
```

Service Component Architecture – Programming Model

*CICS TS v4.1**

WebSphere V7.0.0
SCA Feature Pack V1.0.0



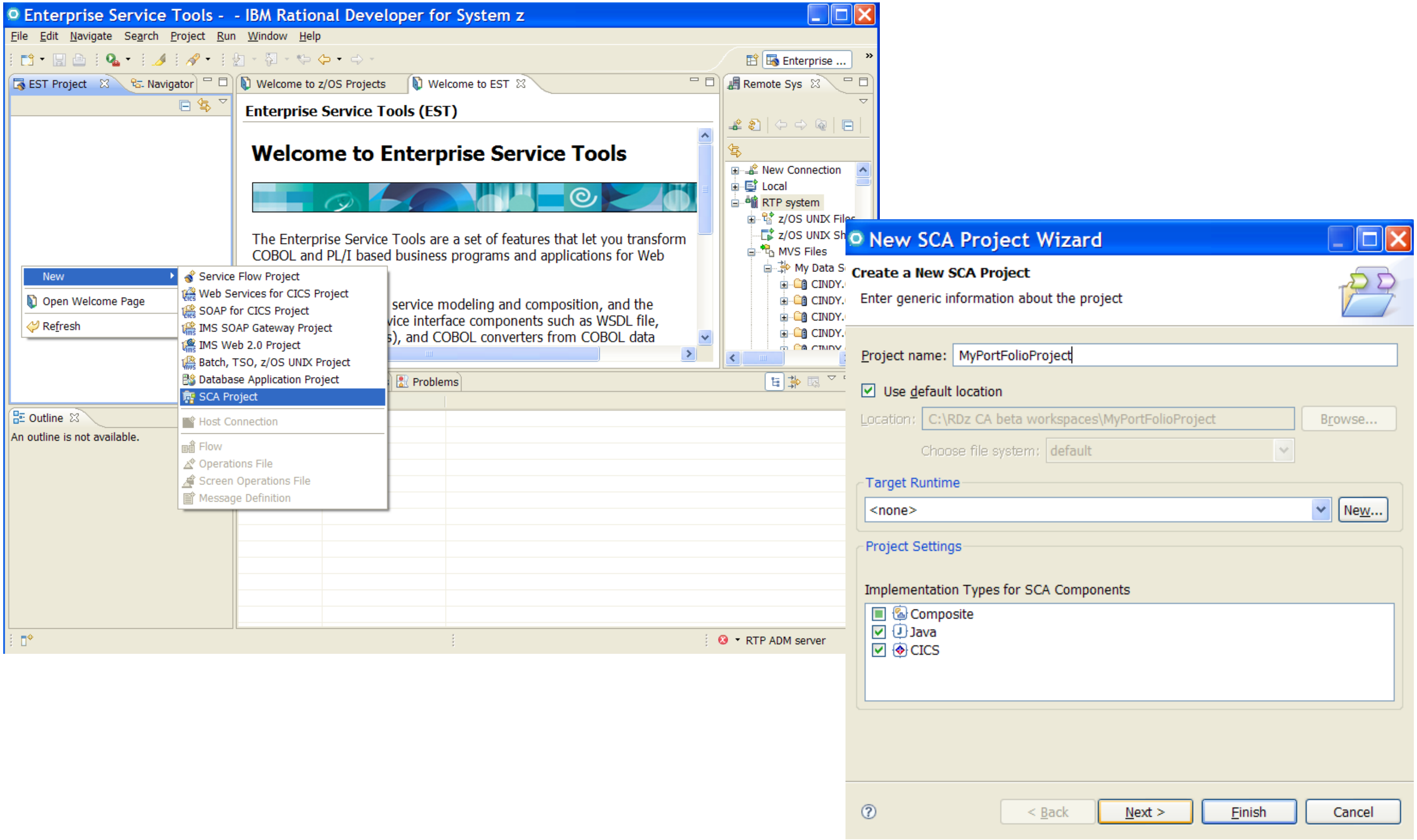
```
CICS COBOL Program

EXEC CICS PUT CONTAINER("SYMBO
                CHANNEL("QUOTESE
                FROM("IBM ")
.

EXEC CICS INVOKE SERVICE("MySt
                       CHANNEL("QUOT
```

```
WebSphere POJO

@Remotable
public interface stockQuote{
   public String myStockQuoteService();}

@Service(stockQuoteImpl.class)
 public class stockQuoteImpl implements stockQuote
```

Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- **Defining and implementing SCA components using Rational Developer for System z**
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- Summary and Questions

RDz SCA Tooling Support



Enterprise Service Tools - - IBM Rational Developer for System z

Enterprise Service Tools (EST)

Welcome to Enterprise Service Tools

The Enterprise Service Tools are a set of features that let you transform COBOL and PL/I based business programs and applications for Web

service modeling and composition, and the
vice interface components such as WSDL file,
s), and COBOL converters from COBOL data

An outline is not available.

New SCA Project Wizard

Create a New SCA Project

Enter generic information about the project

Project name: MyPortFolioProject

☑ Use default location

Location: C:\RDz CA beta workspaces\MyPortFolioProject

Choose file system: default

Target Runtime

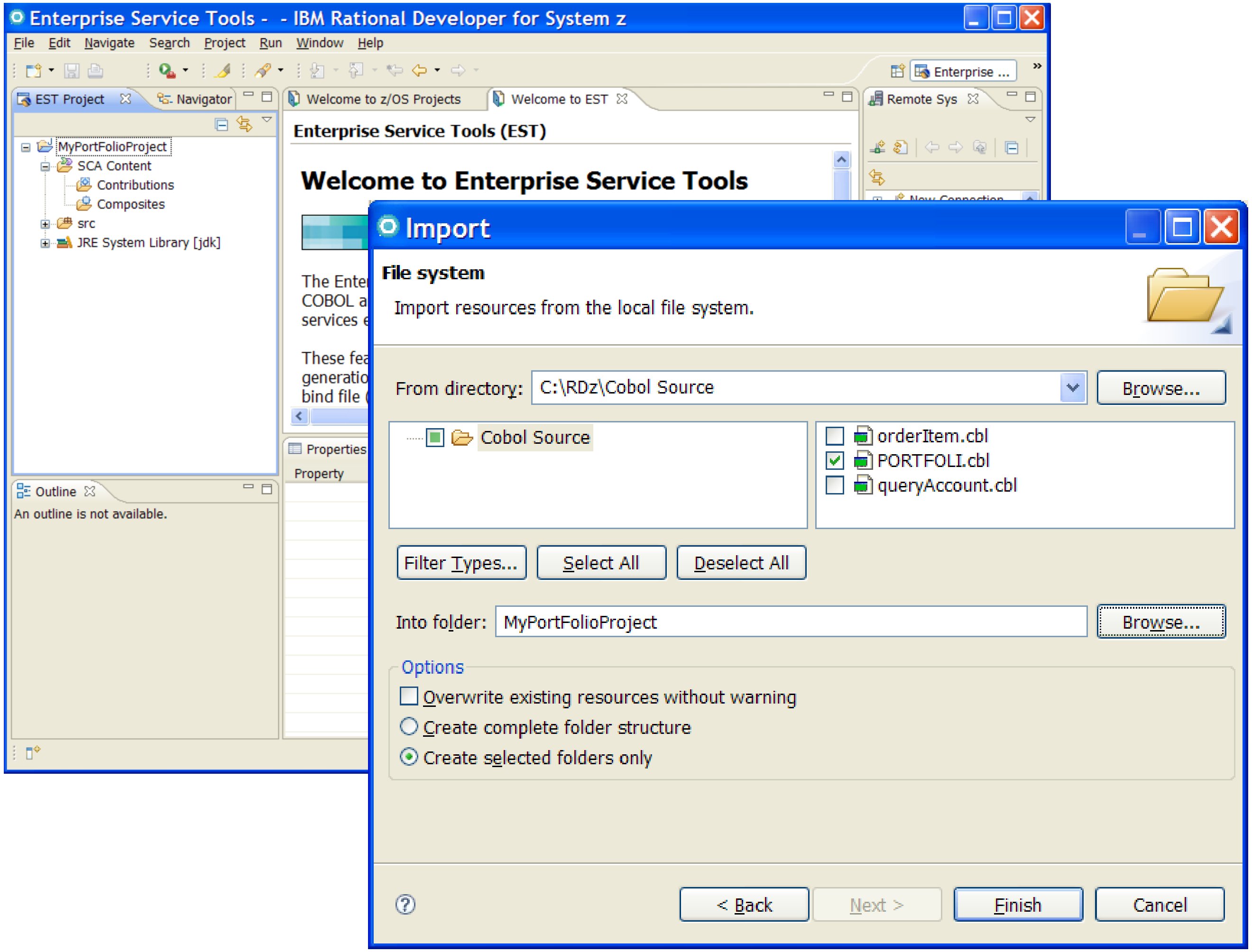
<none>

Project Settings

Implementation Types for SCA Components

☑ Composite
☑ Java
☑ CICS

RDz SCA Tooling Support



An outline is not available.

Enterprise Service Tools (EST)

Welcome to Enterprise Service Tools

The Ente
COBOL a
services

These fea
generatio
bind file

File system

Import resources from the local file system.

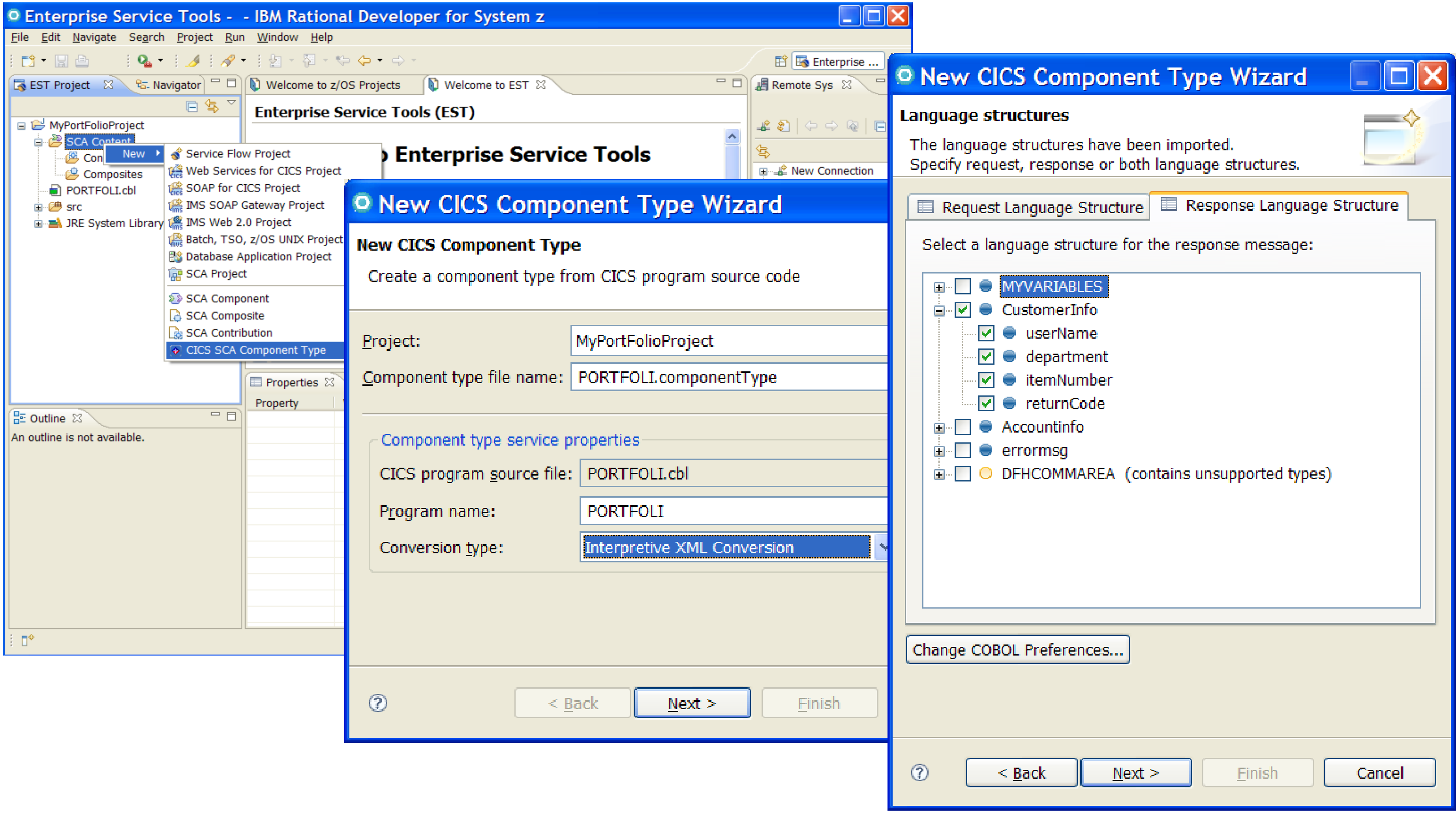
From directory: C:\RDz\Cobol Source

□ orderItem.cbl
☑ PORTFOLI.cbl
□ queryAccount.cbl

Into folder: MyPortFolioProject

□ Overwrite existing resources without warning
○ Create complete folder structure
● Create selected folders only

RDz SCA Tooling Support



New CICS Component Type Wizard

New CICS Component Type

Create a component type from CICS program source code

Project: MyPortFolioProject

Component type file name: PORTFOLI.componentType

Component type service properties

CICS program source file: PORTFOLI.cbl

Program name: PORTFOLI

Conversion type: Interpretive XML Conversion

New CICS Component Type Wizard

Language structures

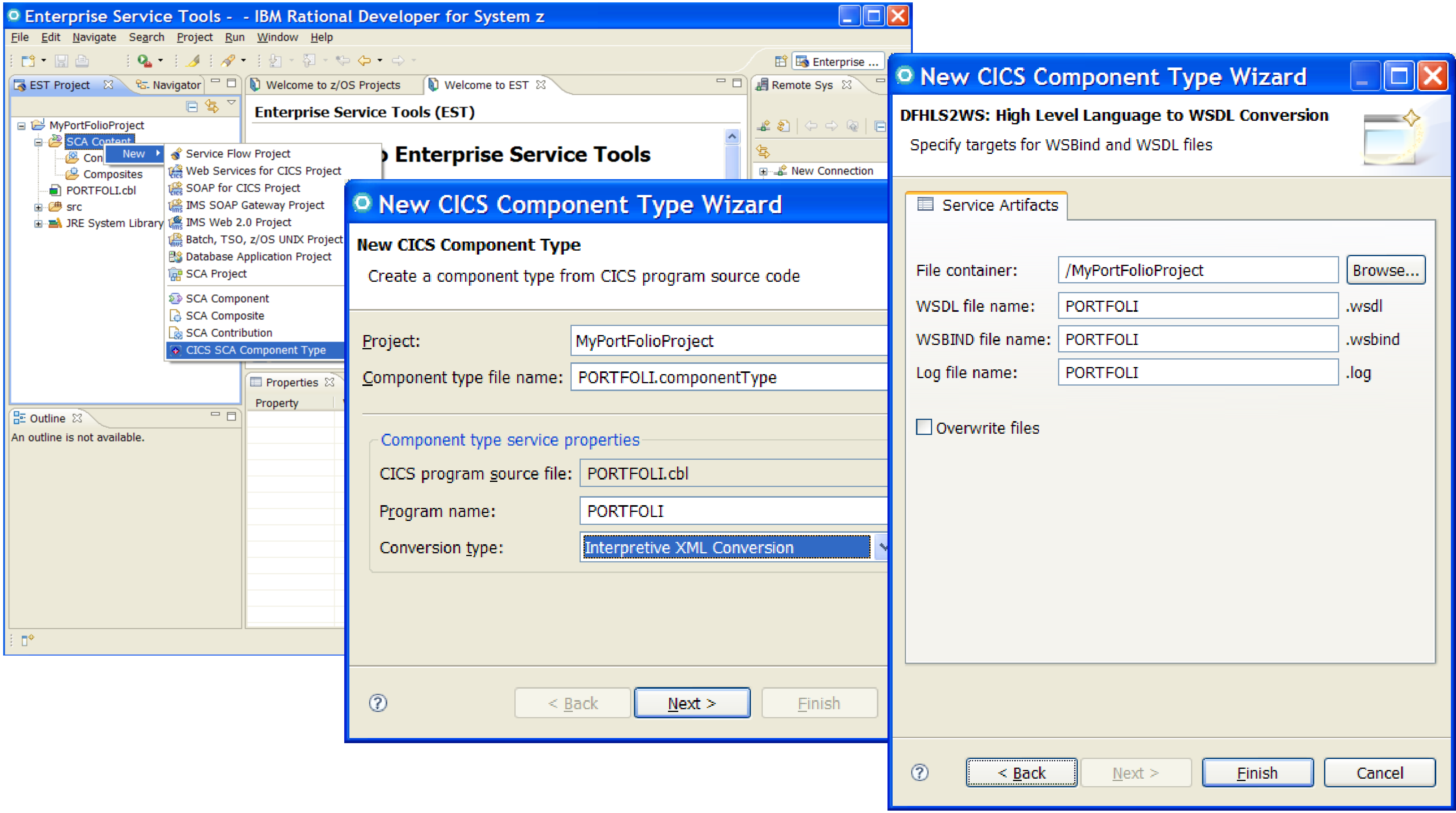
The language structures have been imported.
Specify request, response or both language structures.

Request Language Structure | Response Language Structure

Select a language structure for the response message:

- ☐ MYVARIABLES
- ☑ CustomerInfo
  - ☑ userName
  - ☑ department
  - ☑ itemNumber
  - ☑ returnCode
- ☐ Accountinfo
- ☐ errormsg
- ☐ DFHCOMMAREA (contains unsupported types)

RDz SCA Tooling Support



New CICS Component Type Wizard

New CICS Component Type

Create a component type from CICS program source code

Project: MyPortFolioProject

Component type file name: PORTFOLI.componentType

Component type service properties

CICS program source file: PORTFOLI.cbl

Program name: PORTFOLI

Conversion type: Interpretive XML Conversion

< Back | Next > | Finish

New CICS Component Type Wizard

DFHLS2WS: High Level Language to WSDL Conversion

Specify targets for WSBind and WSDL files

Service Artifacts

File container: /MyPortFolioProject Browse...

WSDL file name: PORTFOLI .wsdl

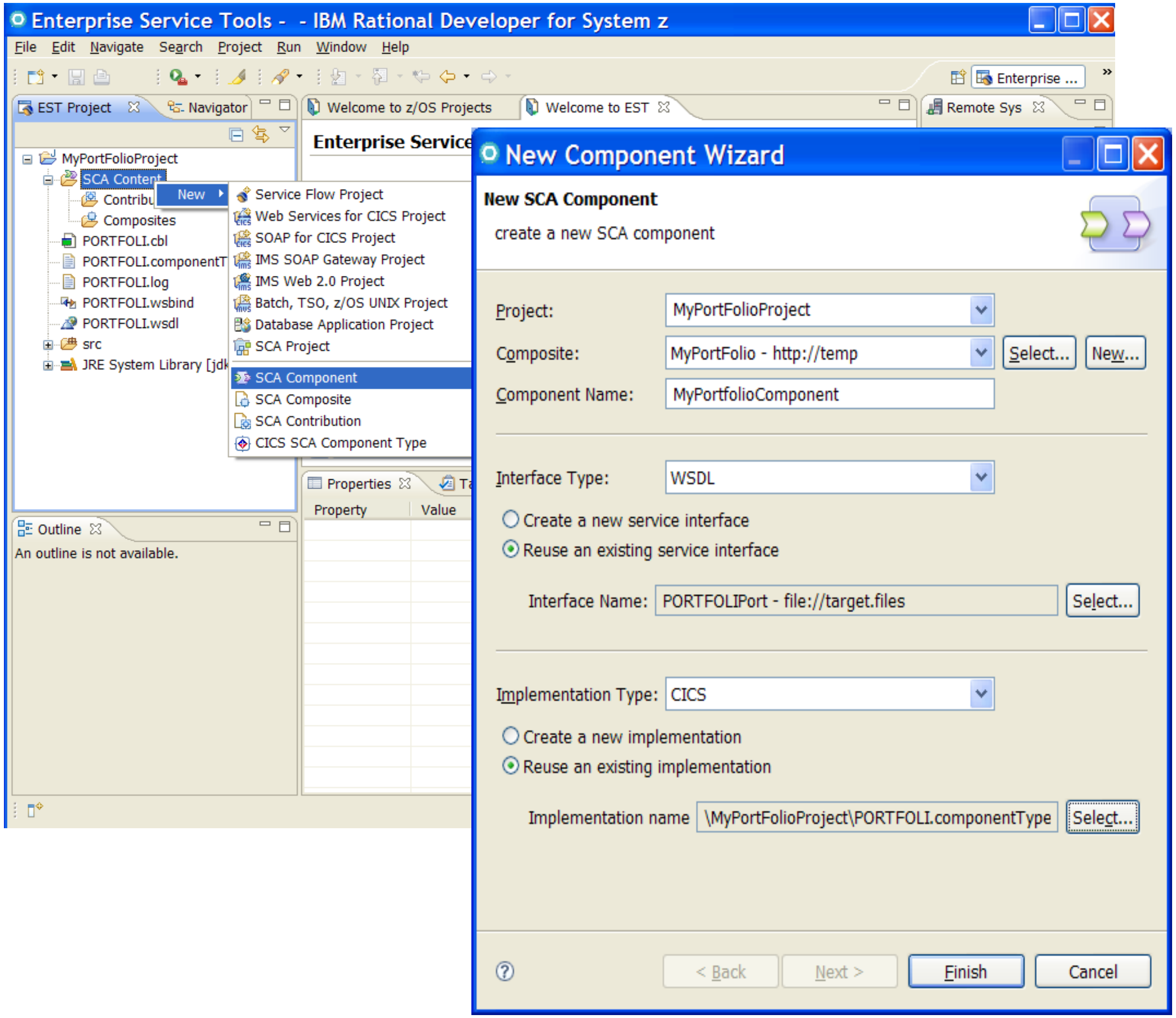
WSBIND file name: PORTFOLI .wsbind

Log file name: PORTFOLI .log

☐ Overwrite files

< Back | Next > | Finish | Cancel

RDz SCA Tooling Support



New SCA Component

create a new SCA component

Project: MyPortFolioProject

Composite: MyPortFolio - http://temp Select... New...

Component Name: MyPortfolioComponent

Interface Type: WSDL

○ Create a new service interface

● Reuse an existing service interface

Interface Name: PORTFOLIPort - file://target.files Select...

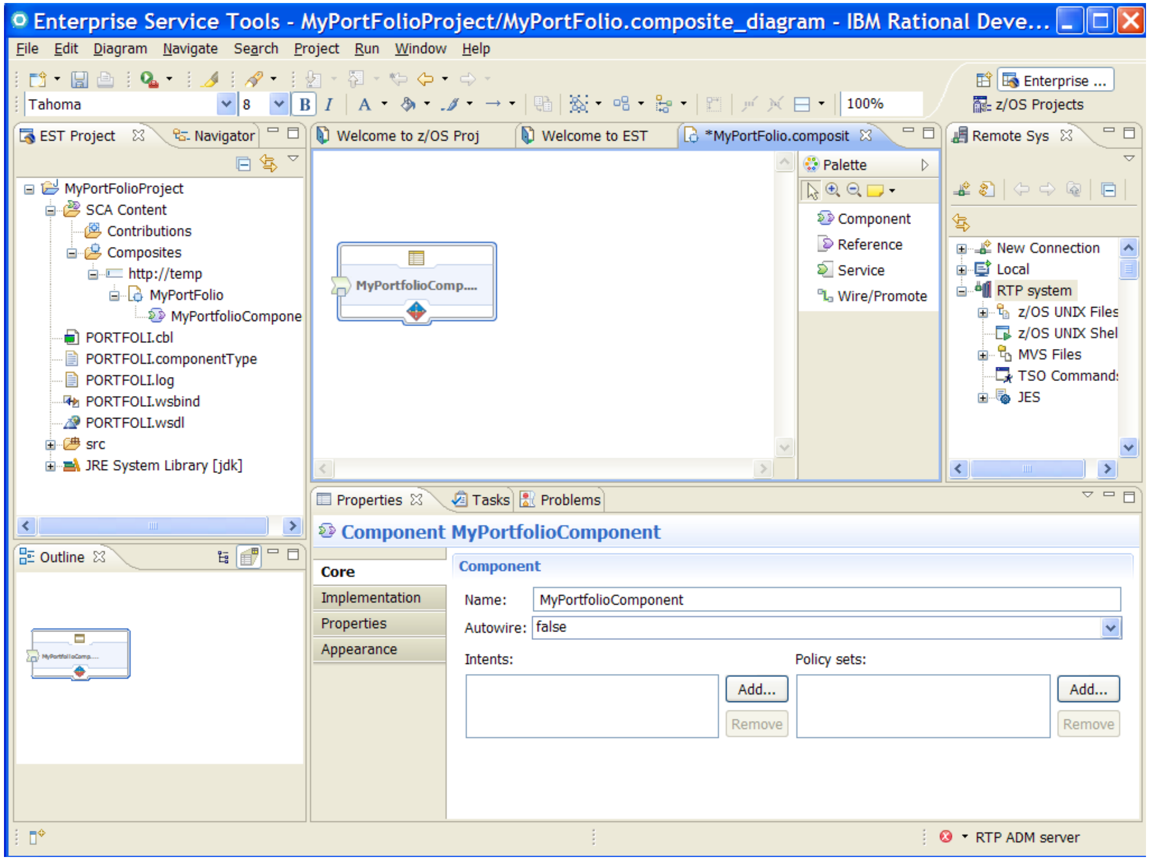
Implementation Type: CICS

○ Create a new implementation

● Reuse an existing implementation

Implementation name \MyPortFolioProject\PORTFOLI.componentType Select...

RDz SCA Tooling Support



Component MyPortfolioComponent

Component

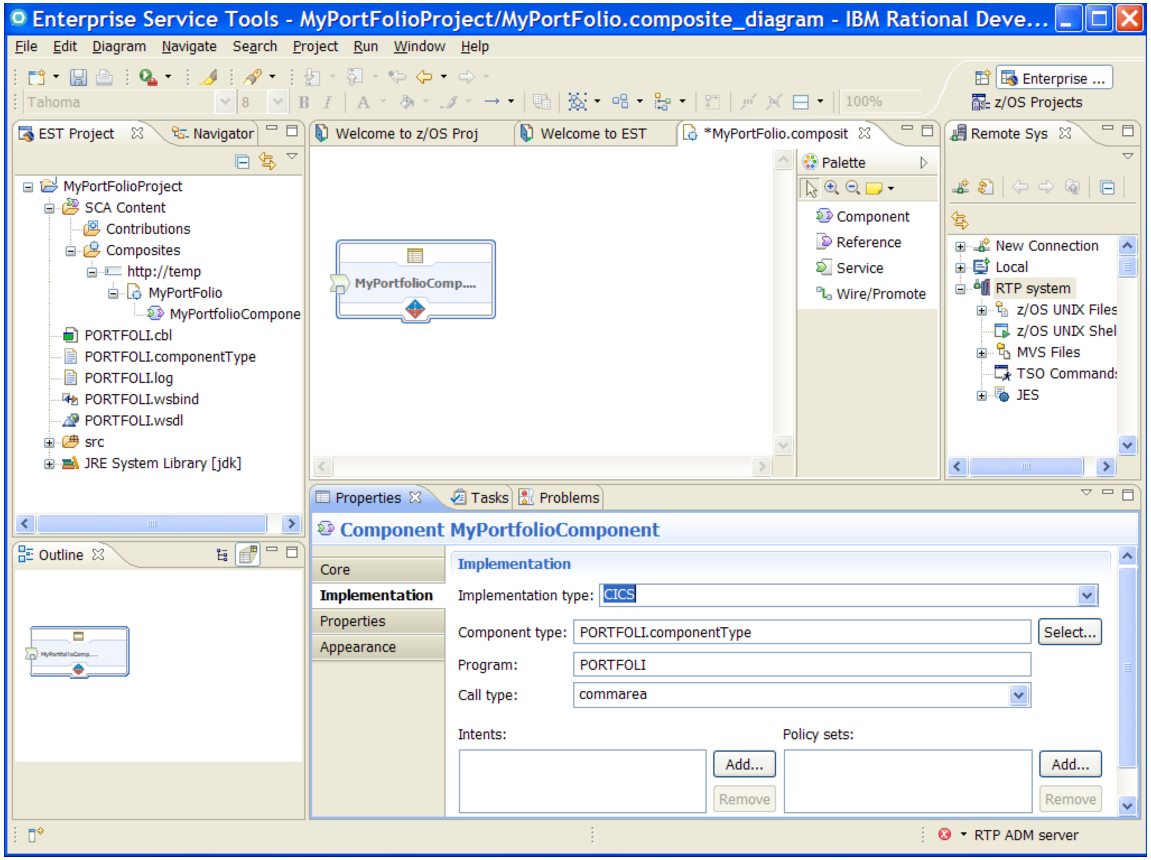
Name: MyPortfolioComponent

Autowire: false

Intents:

Policy sets:

RDz SCA Tooling Support



Component MyPortfolioComponent

Implementation

Implementation type: CICS

Component type: PORTFOLI.componentType

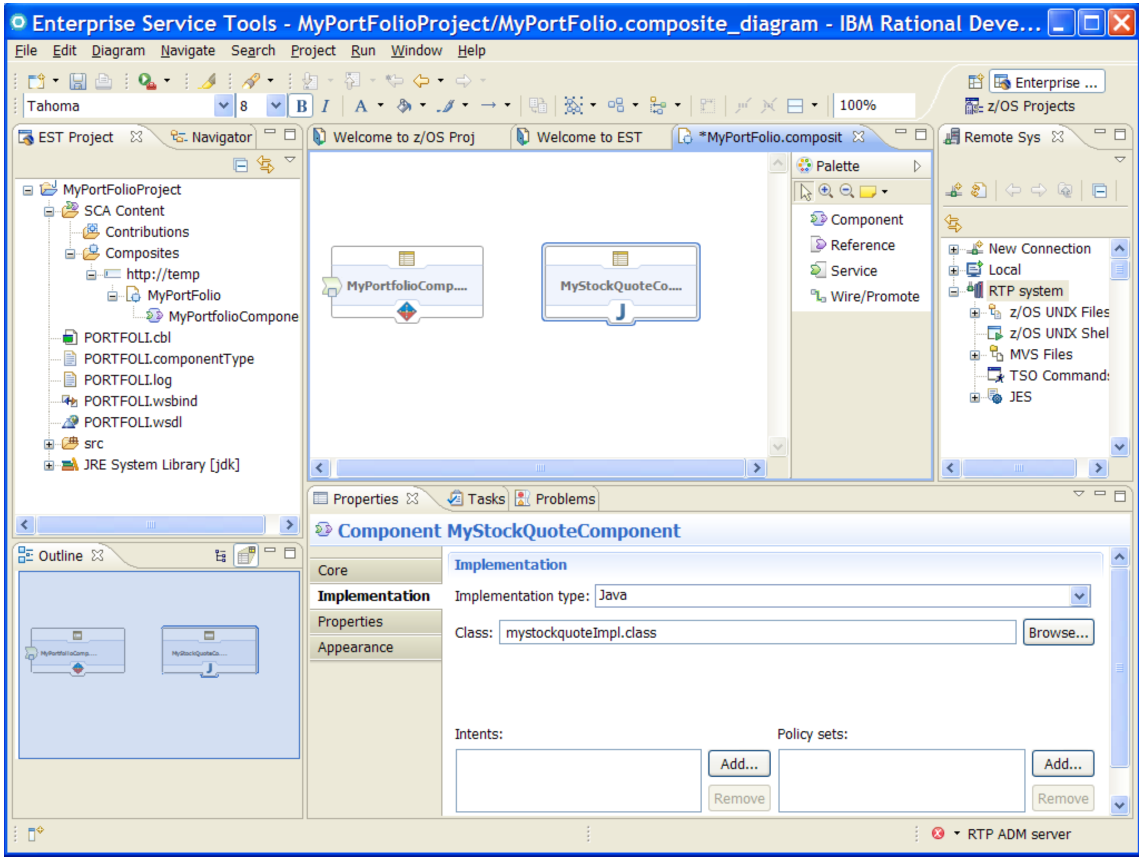
Program: PORTFOLI

Call type: commarea

Intents:

Policy sets:

RDz SCA Tooling Support



Enterprise Service Tools - MyPortFolioProject/MyPortFolio.composite_diagram - IBM Rational Deve...

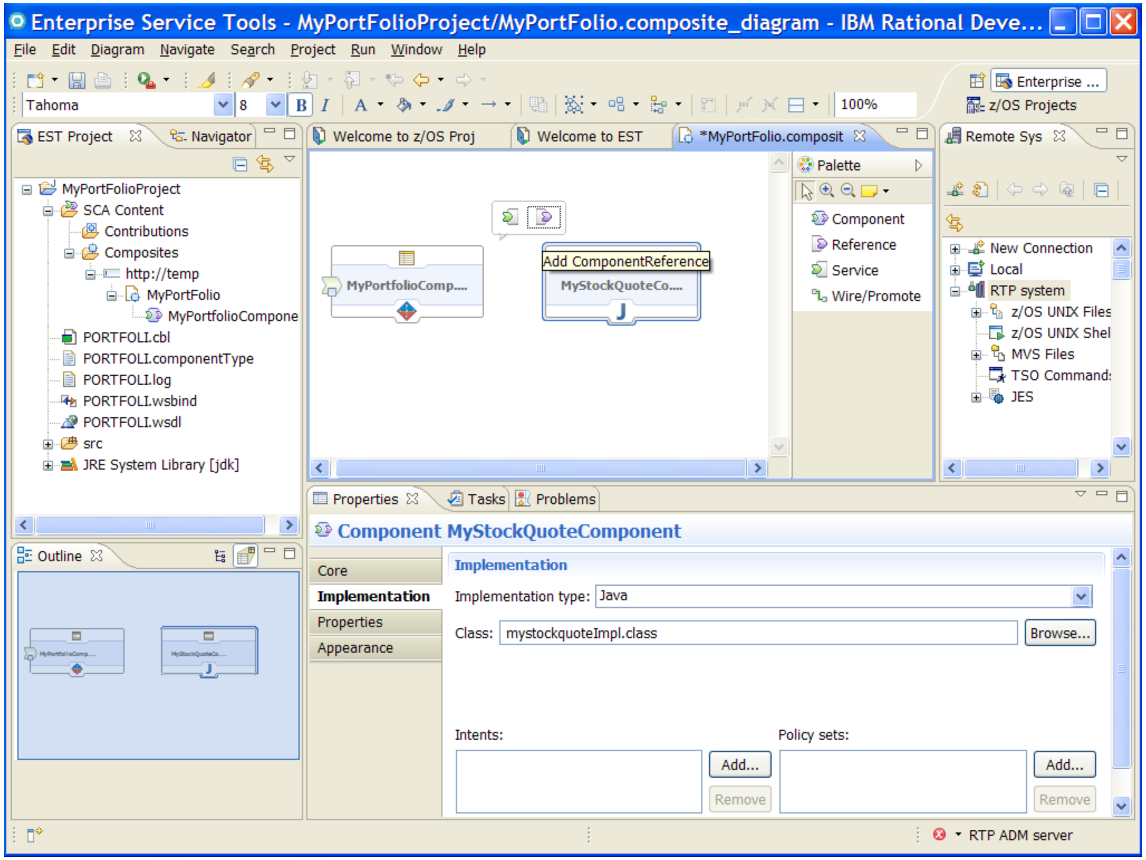
Component MyStockQuoteComponent

Implementation

Implementation type: Java

Class: mystockquoteImpl.class

RDz SCA Tooling Support



Component MyStockQuoteComponent

Implementation

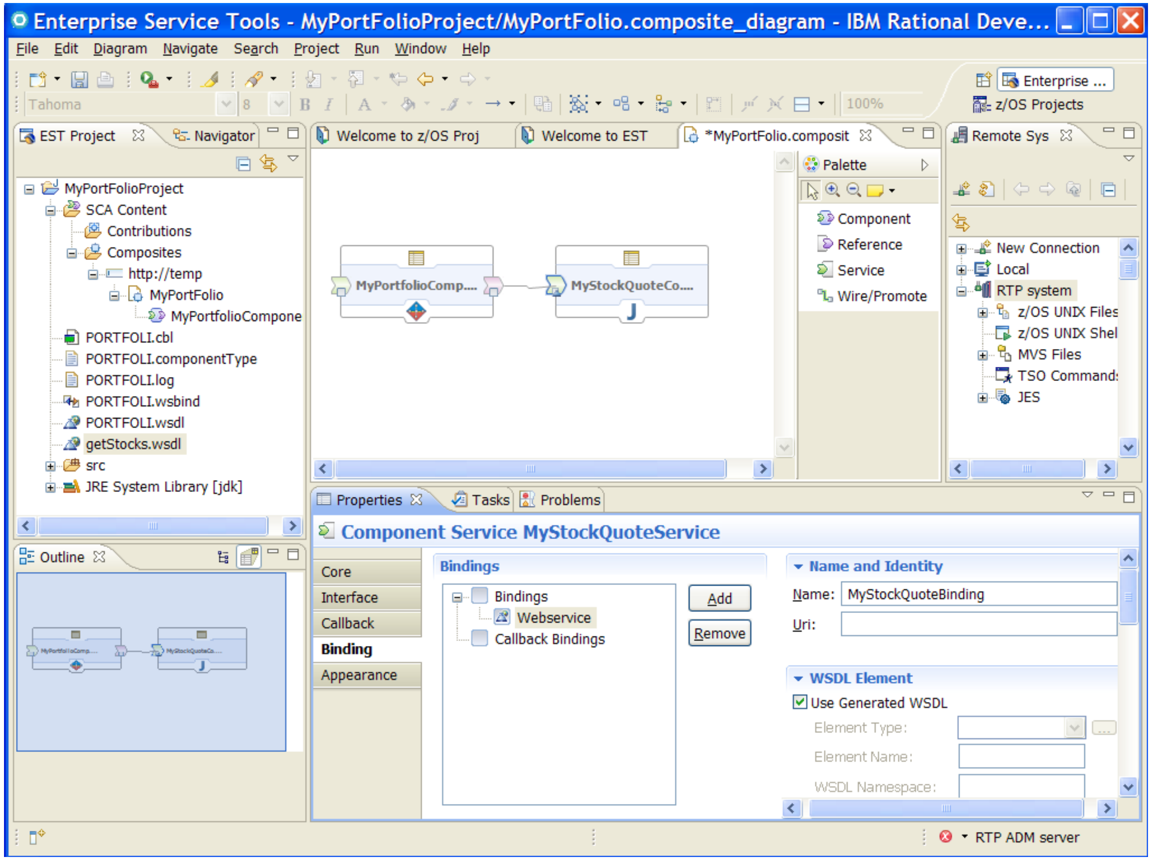
Implementation type: Java

Class: mystockquoteImpl.class

Intents:

Policy sets:

RDz SCA Tooling Support



Component Service MyStockQuoteService

Bindings

- Bindings
  - Webservice
- Callback Bindings

Name: MyStockQuoteBinding

Uri:

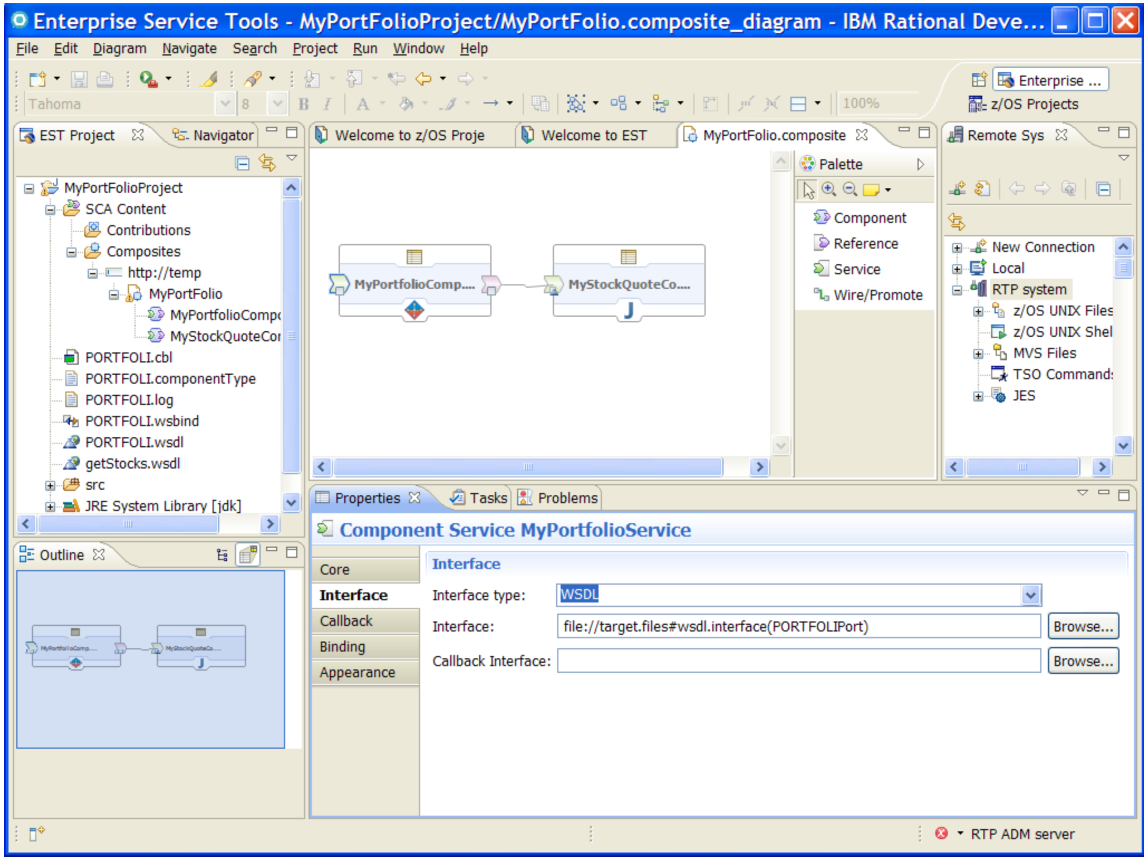
☑ Use Generated WSDL

Element Type:

Element Name:

WSDL Namespace:

RDz SCA Tooling Support



Enterprise Service Tools - MyPortFolioProject/MyPortFolio.composite_diagram - IBM Rational Deve...

Component Service MyPortfolioService

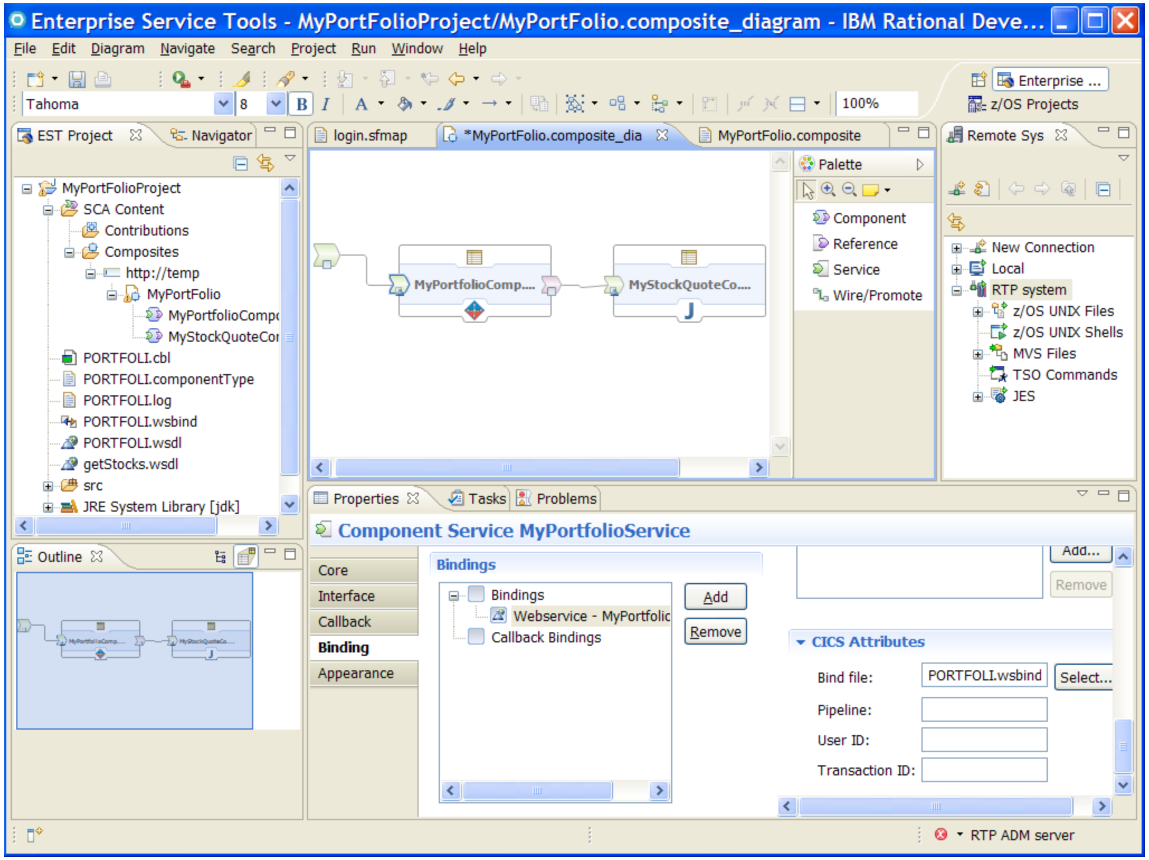
Interface

Interface type: WSDL

Interface: file://target.files#wsdl.interface(PORTFOLIPort)

Callback Interface:

RDz SCA Tooling Support


Enterprise Service Tools - MyPortFolioProject/MyPortFolio.composite_diagram - IBM Rational Deve...

Component Service MyPortfolioService

Bindings

- Bindings
  - Webservice - MyPortfolic
- Callback Bindings

CICS Attributes

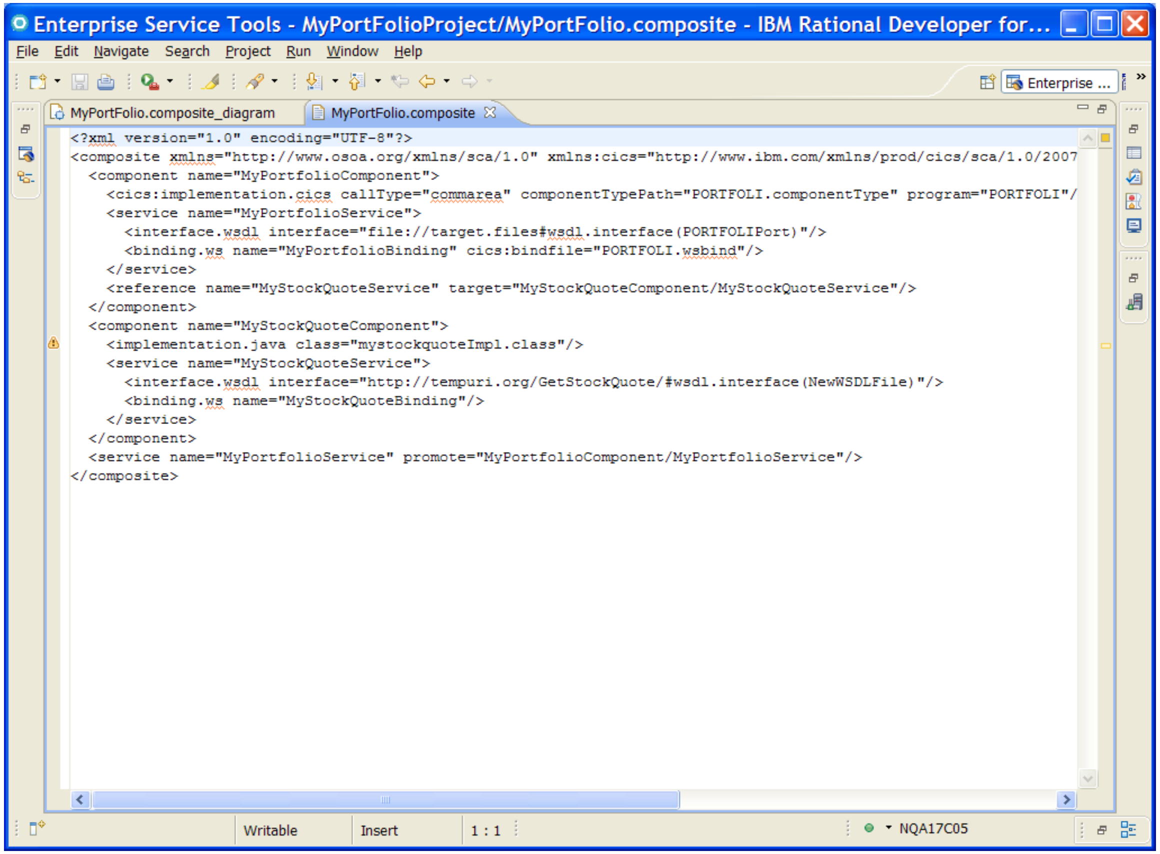
Bind file: PORTFOLI.wsbind

Pipeline:

User ID:

Transaction ID:

RDz SCA Tooling Support

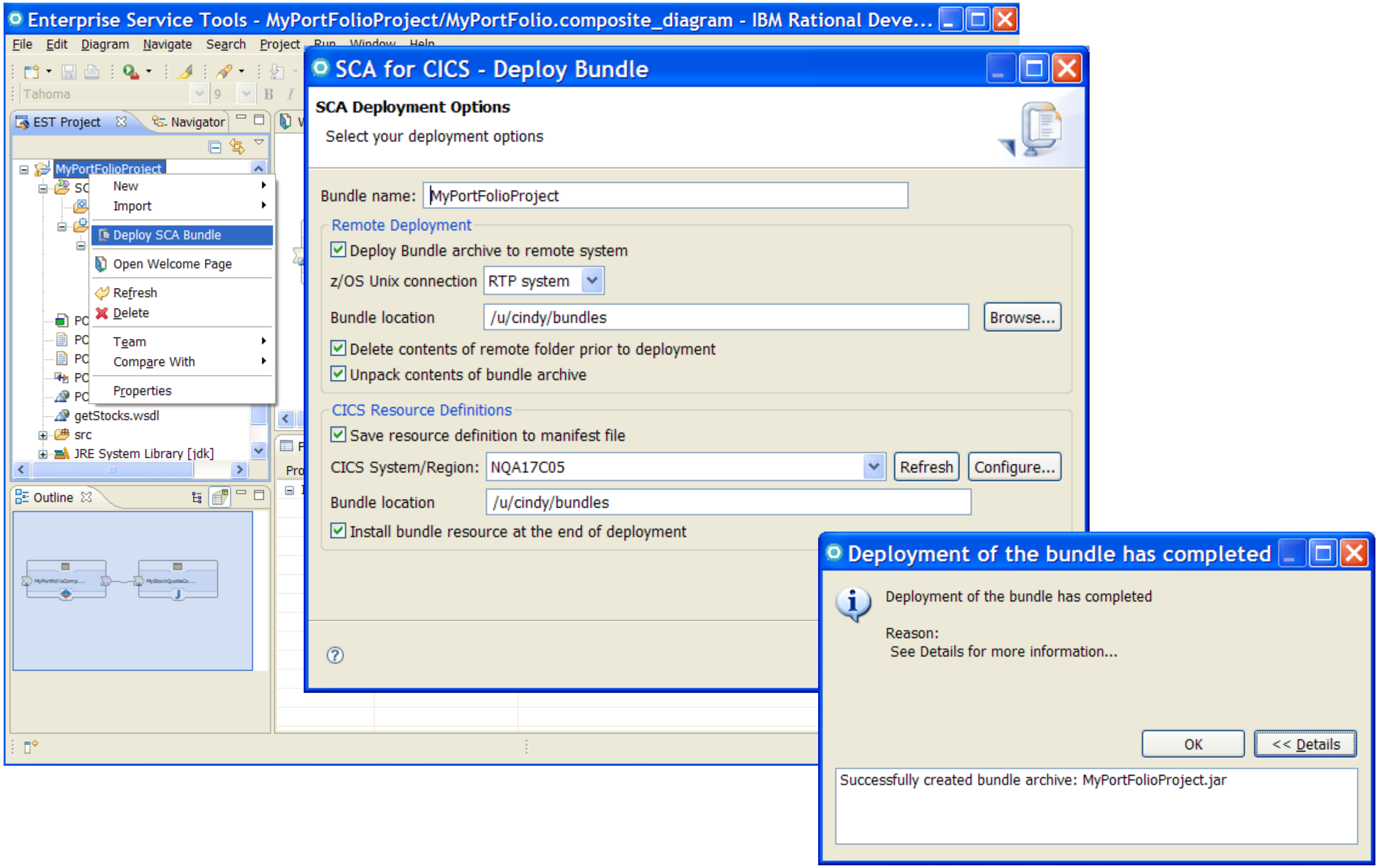


Enterprise Service Tools - MyPortFolioProject/MyPortFolio.composite - IBM Rational Developer for...
File Edit Navigate Search Project Run Window Help
Enterprise ...
MyPortFolio.composite_diagram
MyPortFolio.composite

```
<?xml version="1.0" encoding="UTF-8"?>
<composite xmlns="http://www.osoa.org/xmlns/sca/1.0" xmlns:cics="http://www.ibm.com/xmlns/prod/cics/sca/1.0/2007
  <component name="MyPortfolioComponent">
    <cics:implementation.cics callType="commarea" componentTypePath="PORTFOLI.componentType" program="PORTFOLI"/
    <service name="MyPortfolioService">
      <interface.wsdl interface="file://target.files#wsdl.interface(PORTFOLIPort)"/>
      <binding.ws name="MyPortfolioBinding" cics:bindfile="PORTFOLI.wsbind"/>
    </service>
    <reference name="MyStockQuoteService" target="MyStockQuoteComponent/MyStockQuoteService"/>
  </component>
  <component name="MyStockQuoteComponent">
    <implementation.java class="mystockquoteImpl.class"/>
    <service name="MyStockQuoteService">
      <interface.wsdl interface="http://tempuri.org/GetStockQuote/#wsdl.interface(NewWSDLFile)"/>
      <binding.ws name="MyStockQuoteBinding"/>
    </service>
  </component>
  <service name="MyPortfolioService" promote="MyPortfolioComponent/MyPortfolioService"/>
</composite>
```

Writable
Insert
1 : 1
NQA17C05

RDz SCA Tooling Support



SCA for CICS - Deploy Bundle

SCA Deployment Options

Select your deployment options

Bundle name: MyPortFolioProject

Remote Deployment

☑ Deploy Bundle archive to remote system

z/OS Unix connection: RTP system

Bundle location: /u/cindy/bundles

☑ Delete contents of remote folder prior to deployment

☑ Unpack contents of bundle archive

CICS Resource Definitions

☑ Save resource definition to manifest file

CICS System/Region: NQA17C05

Bundle location: /u/cindy/bundles

☑ Install bundle resource at the end of deployment

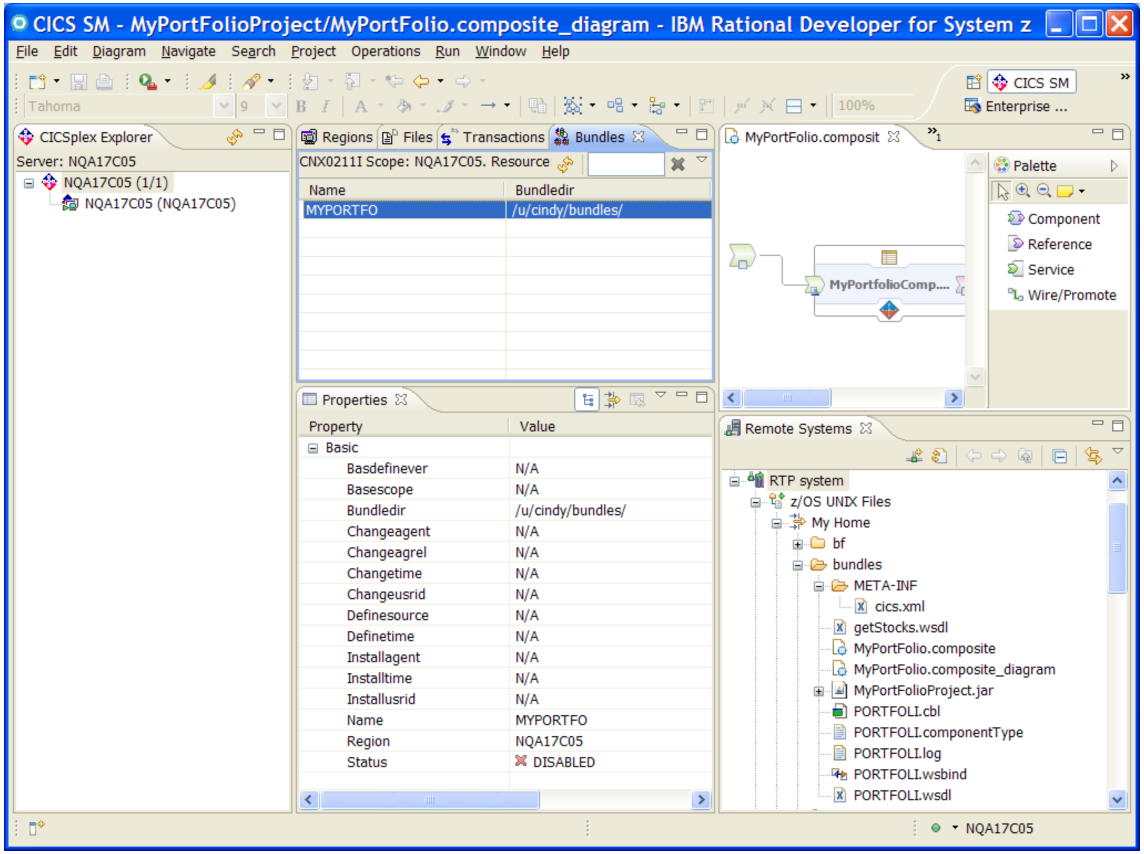
Deployment of the bundle has completed

Deployment of the bundle has completed

Reason:
See Details for more information...

```
Successfully created bundle archive: MyPortFolioProject.jar
```

RDz SCA Tooling Support



CICS SM - MyPortFolioProject/MyPortFolio.composite_diagram - IBM Rational Developer for System z

Server: NQA17C05

- NQA17C05 (1/1)
  - NQA17C05 (NQA17C05)

CNX0211I Scope: NQA17C05. Resource

Name	Bundledir
MYPORTFO	/u/cindy/bundles/

Property	Value
Basic	
Basdefinever	N/A
Basescope	N/A
Bundledir	/u/cindy/bundles/
Changeagent	N/A
Changeagrel	N/A
Changetime	N/A
Changeusrid	N/A
Definesource	N/A
Definetime	N/A
Installagent	N/A
Installtime	N/A
Installusrid	N/A
Name	MYPORTFO
Region	NQA17C05
Status	☒ DISABLED

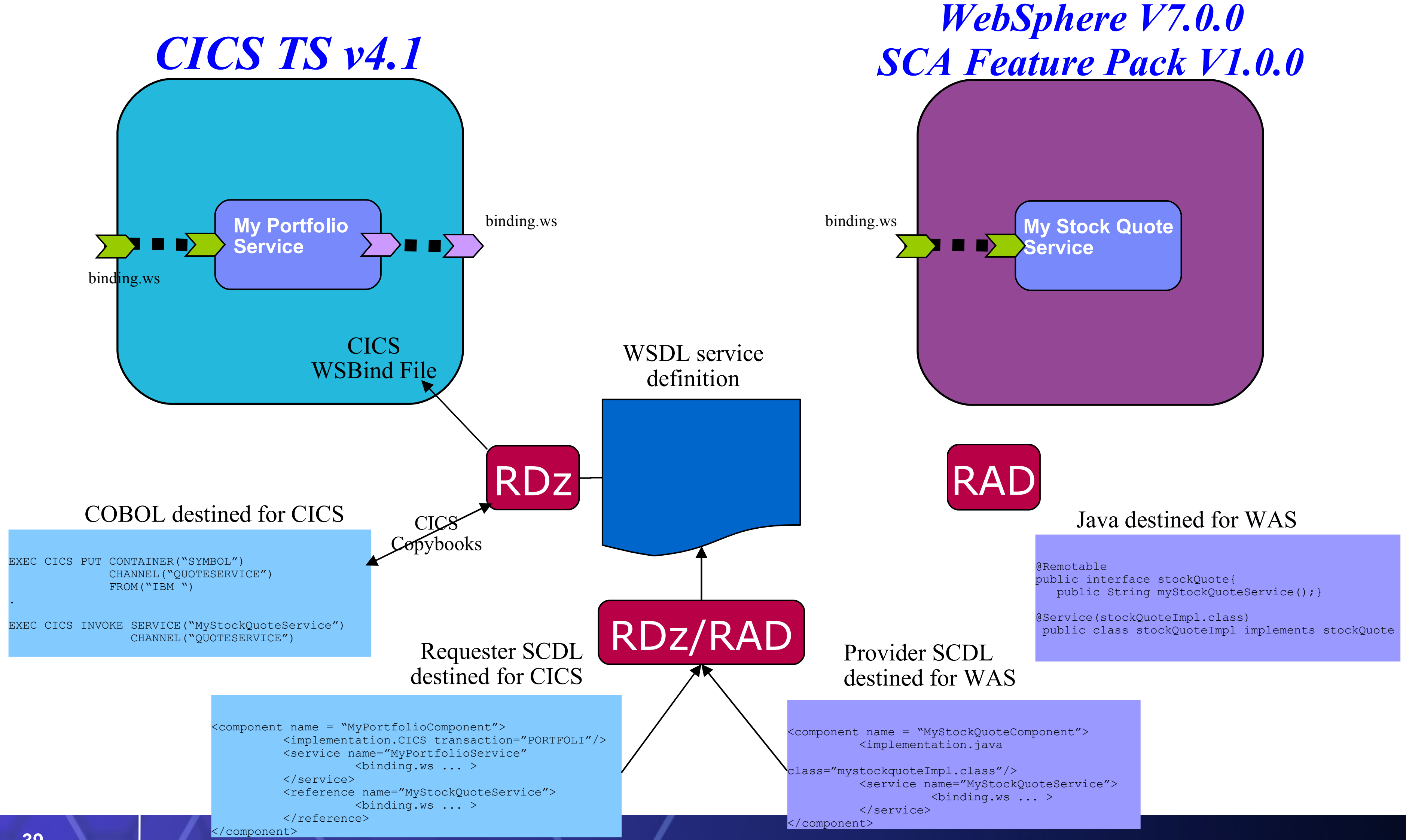
- RTP system
  - z/OS UNIX Files
    - My Home
      - bf
      - bundles
        - META-INF
          - cics.xml
        - getStocks.wsdl
        - MyPortFolio.composite
        - MyPortFolio.composite_diagram
        - MyPortFolioProject.jar
        - PORTFOLI.cbl
        - PORTFOLI.componentType
        - PORTFOLI.log
        - PORTFOLI.wsbind
        - PORTFOLI.wsdl

● NQA17C05

Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- **Deploying and running SCA components in CICS TS**
- Gaining value from SCA-based approach to architecture
- Summary and Questions

Service Component Architecture – Development Artifacts



COBOL destined for CICS

```
EXEC CICS PUT CONTAINER(“SYMBOL”)
              CHANNEL(“QUOTESERVICE”)
              FROM(“IBM “)
.

EXEC CICS INVOKE SERVICE(“MyStockQuoteService”)
                 CHANNEL(“QUOTESERVICE”)
```

Java destined for WAS

```
@Remotable
public interface stockQuote{
   public String myStockQuoteService();}

@Service(stockQuoteImpl.class)
 public class stockQuoteImpl implements stockQuote
```

Requester SCDL destined for CICS

```
<component name = “MyPortfolioComponent”>
          <implementation.CICS transaction=”PORTFOLI”/>
          <service name=”MyPortfolioService”
                    <binding.ws ... >
          </service>
          <reference name=”MyStockQuoteService”>
                    <binding.ws ... >
          </reference>
</component>
```

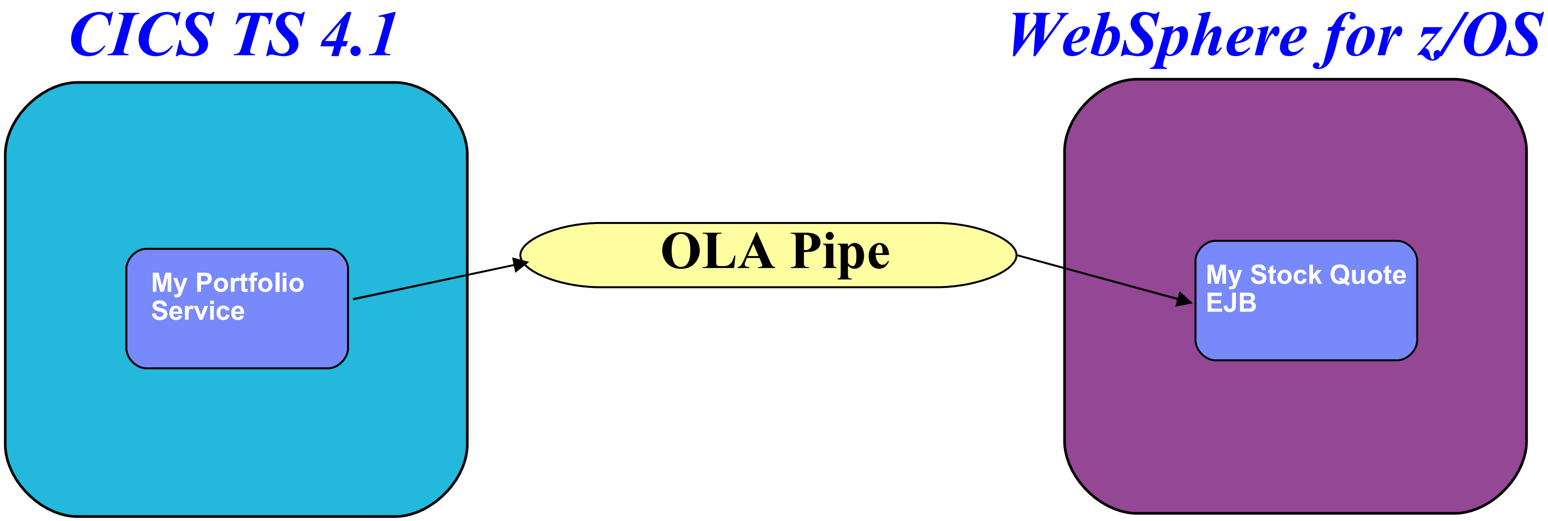
Provider SCDL destined for WAS

```
<component name = “MyStockQuoteComponent”>
          <implementation.java

class=”mystockquoteImpl.class”/>
          <service name=”MyStockQuoteService”>
                    <binding.ws ... >
          </service>
</component>
```

Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- The Portfolio Scenario
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- **Gaining value from SCA-based approach to architecture**
- Summary and Questions

Optimized Local Adapter – Low Level APIs



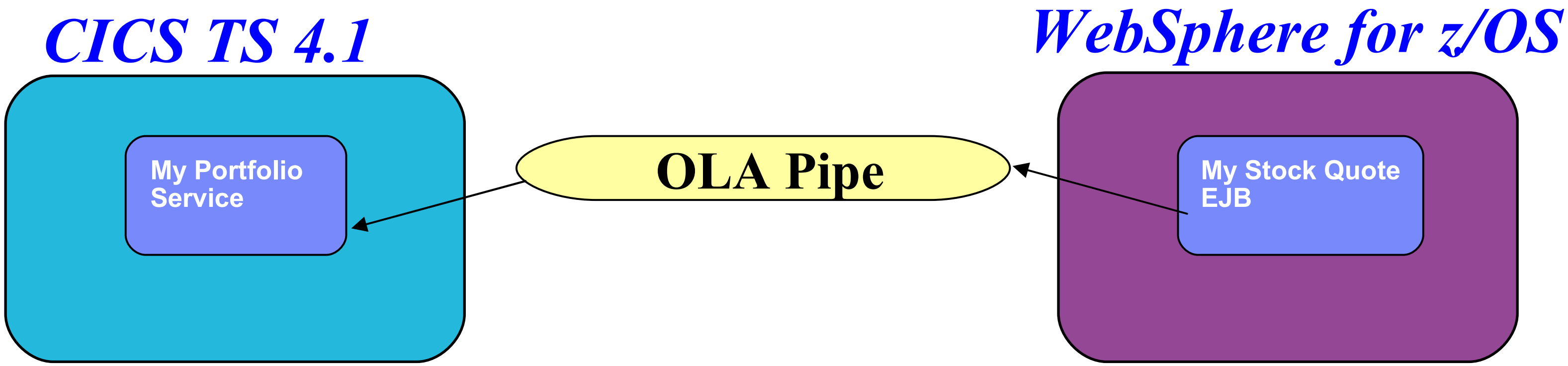
CICS COBOL Program

```
.
.
CALL to register to send requests into WAS
.
.
CALL to invoke specific ejb/com/myFirm/MyStockQuote
.
.
```

```
public class MyStockQuote implements
javax.ejb.SessionBean {
public byte[] execute(byte[] arg0) {

   EJB wrapper that will extract binary
information passed on the call from COBOL

 }
}
```

IBM currently intends to deliver an Optimized Local Adapters feature for WebSphere Application Server for z/OS V7 as a future enhancement. This feature would provide non-WAS applications (written in Cobol, C/C++, HLASM) running on z/OS a native connection optimized for exploitation of WAS for z/OS. The connection would be specifically optimized for application aware collocation within a single z/OS image where performance requirements justify closely bound integration. Target availability is second quarter 2009. All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.

Optimized Local Adapter – Low Level APIs



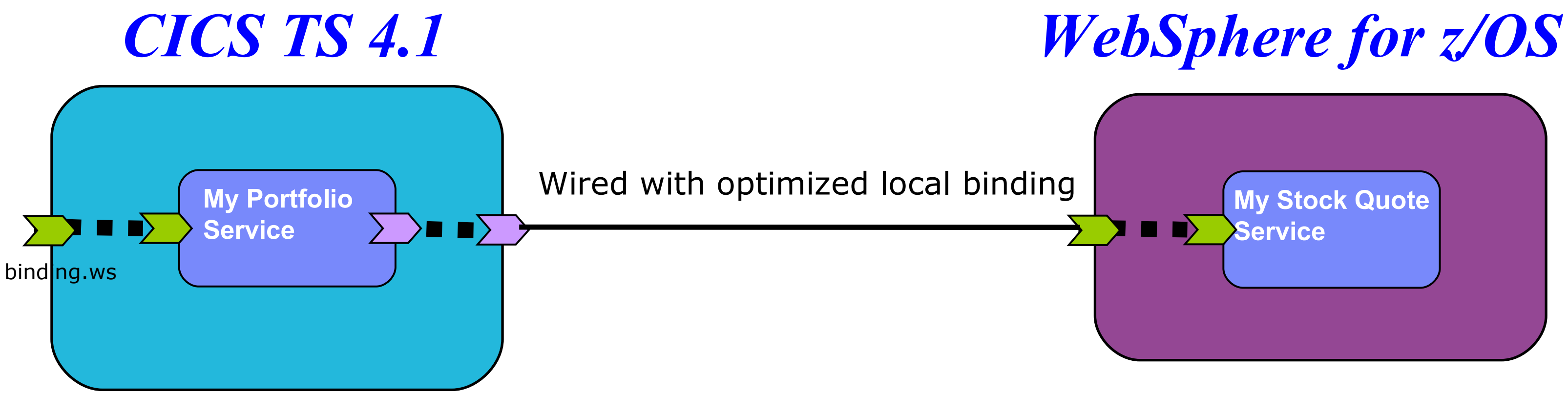
```
CICS COBOL Program
.
  Typical program that gets and puts data into a
container

*  Read content length of container InputData

   EXEC CICS GET CONTAINER(INPUTCONTAINER)
                 CHANNEL(CHANNELNAME)
                             ...
*  Place output string in container OutputData

   EXEC CICS PUT CONTAINER(OUTPUTCONTAINER)
                FROM(OUTPUTSTRING)
                            ...
```

```
public interface MyStockQuote extends
javax.ejb.EJBObject {

  Uses JCA specification to make call into
CICS


}
```

IBM currently intends to deliver an Optimized Local Adapters feature for WebSphere Application Server for z/OS V7 as a future enhancement.

This feature would provide non-WAS applications (written in Cobol, C/C++, HLASM) running on z/OS a native connection optimized for exploitation of WAS for z/OS. The connection would be specifically optimized for application aware collocation within a single z/OS image where performance requirements justify closely bound integration. Target availability is second quarter 2009. All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.

Service Component Architecture



```
<component name = "MyPortfolioComponent">
 <implementation.CICS program="PORTFOLI"/>
 <service name="MyPortfolioService"
      <binding.ws ... >
 </service>
 <reference name="MyStockQuoteService">
 <binding based on optimized local adapter>
 </reference>
</component>
```

```
<component name = "MyStockQuoteComponent">
     <implementation.java
     class="mystockquoteImpl.class"/>
     <service name="MyStockQuoteService">
         <binding based on optimized local
adapter >
      </service>
</component>
```

IBM currently has no plans to deliver an SCA binding based on the optimized local adapter. This page is a depiction of how the technology it does plan to deliver could be combined and/or used within the Service Component Architecture as currently defined. All statements regarding IBM's plans, directions, and intent are subject to change or withdrawal without notice.

Agenda

- Quick SCA in WAS recap
- Introduction to SCA support in CICS TS v4.1
- Defining and implementing SCA components using Rational Developer for System z
- Deploying and running SCA components in CICS TS
- Gaining value from SCA-based approach to architecture
- **Summary and Questions**

References

- CICS Transaction Server v4.1
  - Open Beta: http://www.ibm.com/software/htp/cics/tserver/v41/openbeta/
- Rational Developer for System z
  - http://www.ibm.com/software/awdtools/rdz/
- Open Service Oriented Architecture Web site for SCA v1.0 Specifications
  - http://www.osoa.org/
- OASIS Open CSA Web site for SCA v1.x
  - http://www.oasis-opencsa.org/sca
- Apache Tuscany Web site
  - http://incubator.apache.org/tuscany/
- SCA feature pack support website
  - http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&dc=DB600&uid=swg21329175
- DeveloperWorks
  - http://www.ibm.com/developerworks/websphere

WebSphere software