



TPF Users Group Fall 2007

Creating Web Services For Your z/TPF System

Edwin W. van de Grift
Venue: Ongoing TPF Education

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1:1.0

© IBM Corporation 2007

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

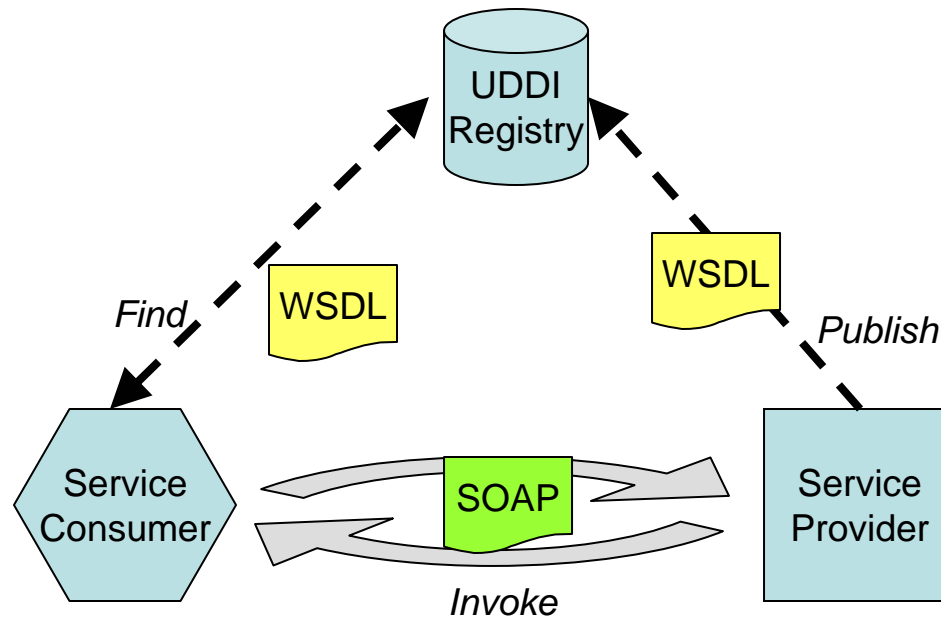
Contents

- Web Services
- Where we were
- Where we are now
- Creating a Web Service for an existing application



Web Services

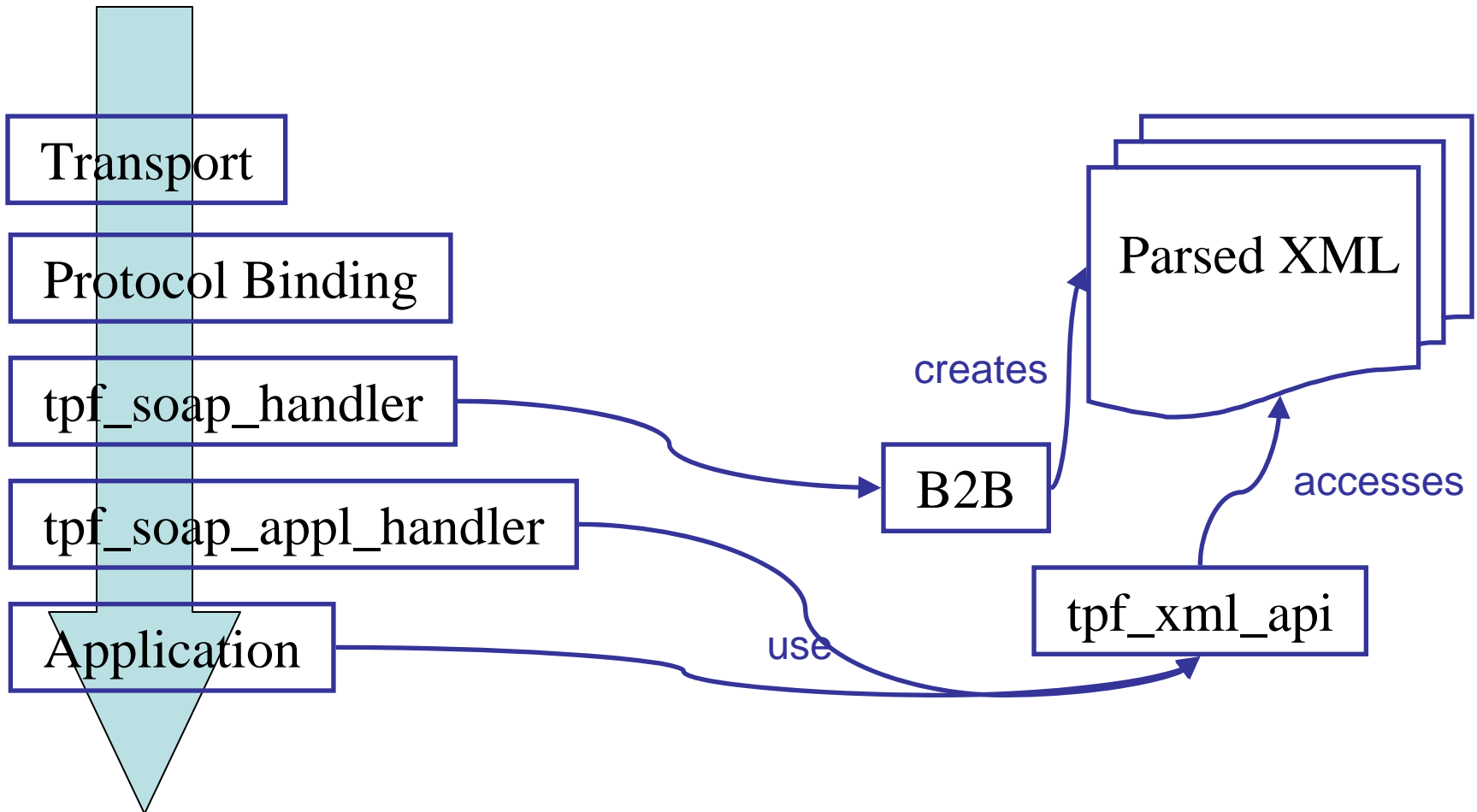
SOA



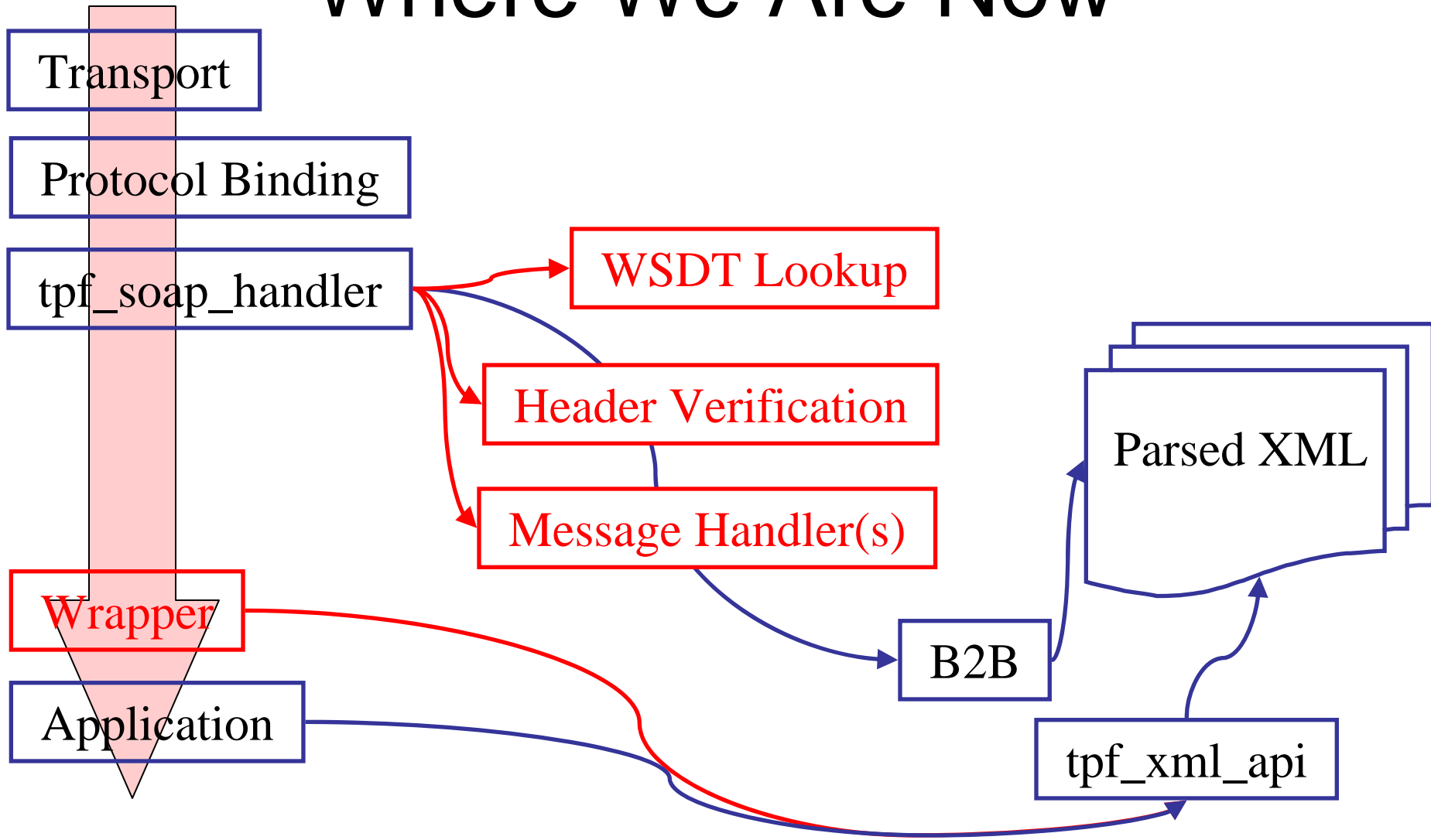
Acronyms

- WSDL
 - Web Services Description Language
- WSIL
 - Web Services Inspecting Language
 - Alternative to using UDDI registry
- WSDT
 - Web Services Deployment Table
 - z/TPF

Where We Were



Where We Are Now



What's New

- New Development Components
 - Create WSDLs
 - Generate Web Service Wrapper Templates
 - Create Deployment Descriptors
 - Test Web Services
 - Generate WSILs
- New Deployment Components
 - Deploy Web Service Wrappers and Deployment Descriptors
 - Publish WSDLs

Creating A Web Service For An Existing TPF Application



Creating A Web Service

- Create the WSDL
- Generate the Wrapper Template
- Create the Deployment Descriptor
- Deploy the Web Service
- Test the Web Service
- Generate the WSIL

The Application

`zdtim`

```
CSMP0097I 12.45.28 CPU-B SS-BSS SSU-HPN IS-01
```

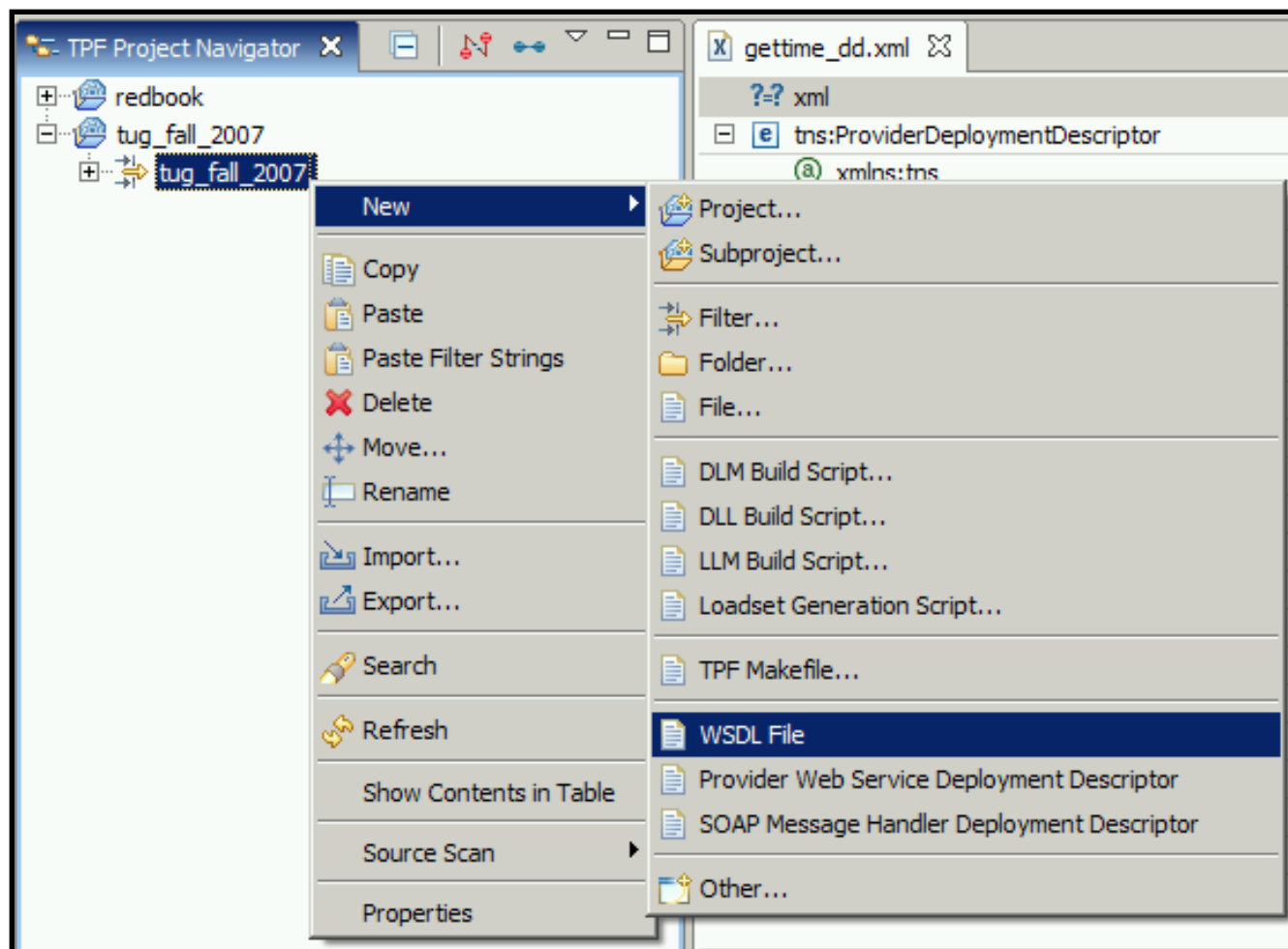
```
DTIM0001I 12.45.28 SUBSYSTEM BSS LOCAL STANDARD TIME+
```

- Same sample application that was used in **Red**book “z/TPF and WebSphere Application Server in a Service Oriented Architecture”

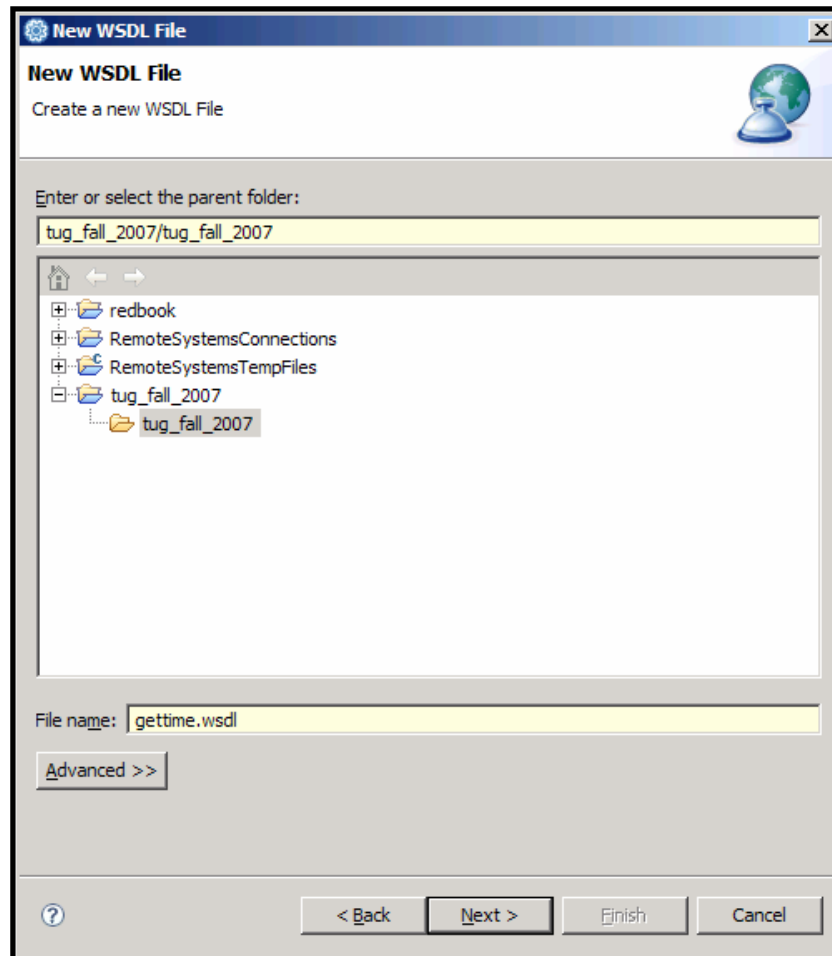
Creating The WSDL



Create WSDL



Name WSDL File



Generate WSDL Template

New WSDL File

Options
Specify the attributes for the new WSDL file.

Target namespace:

Prefix:

Create WSDL Skeleton

Protocol:

SOAP Binding Options

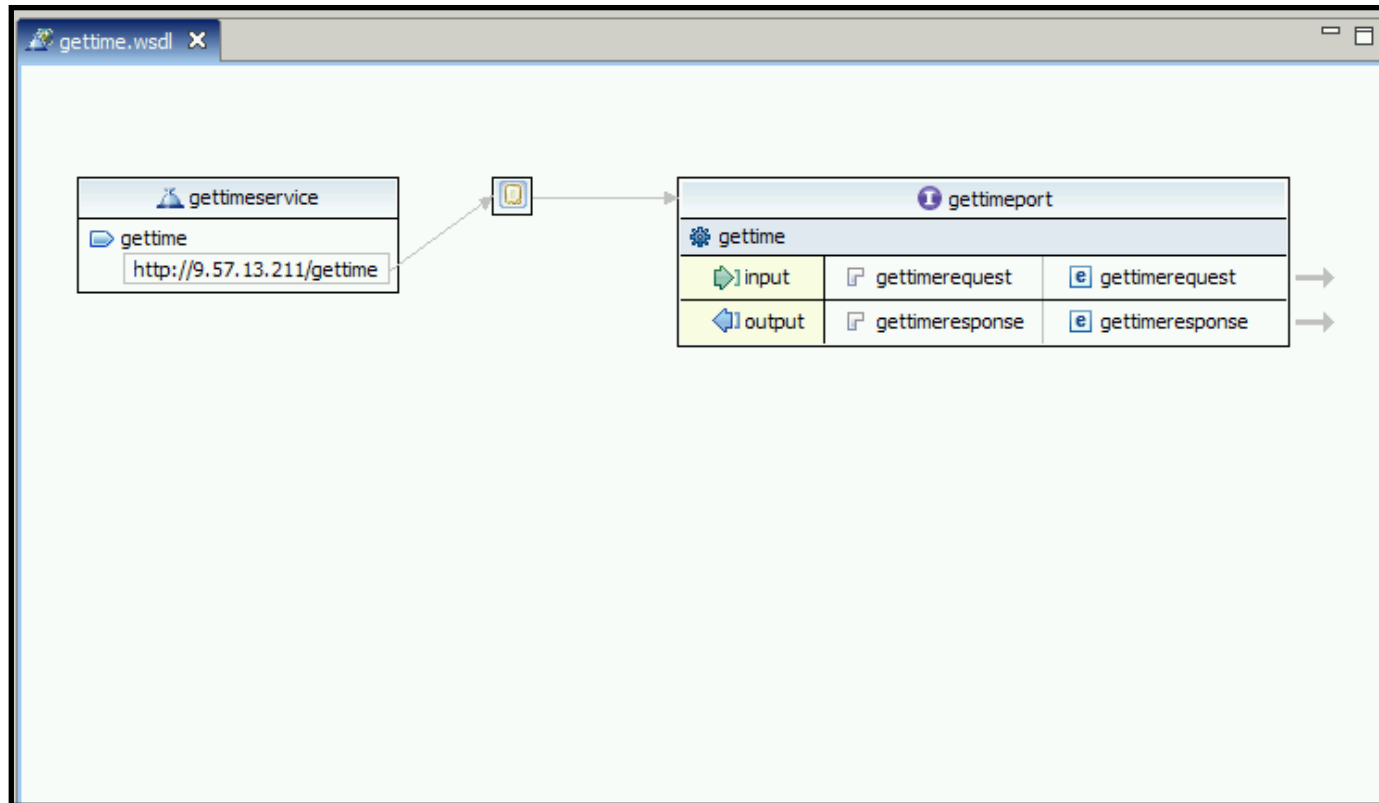
document literal

rpc literal

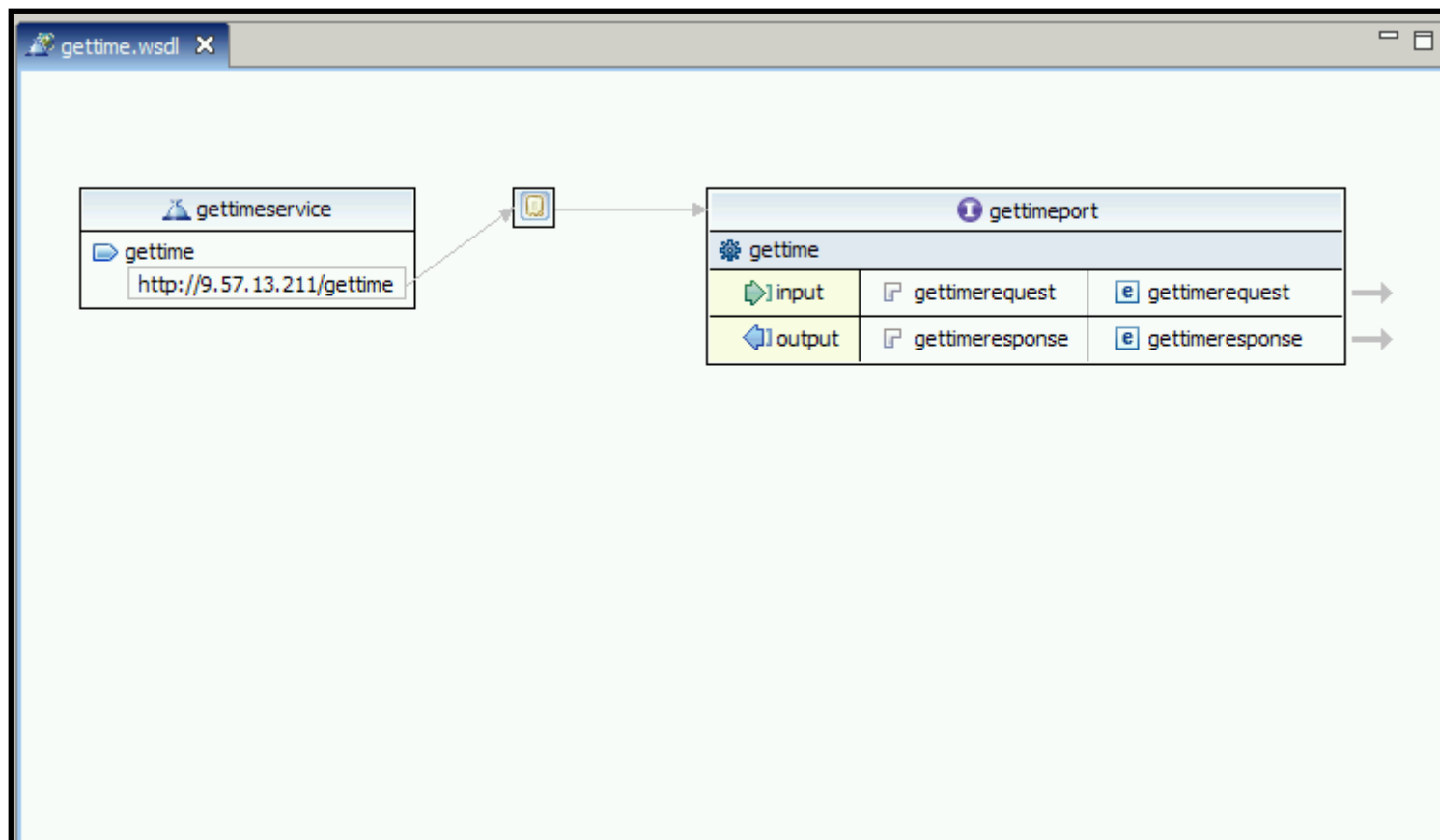
rpc encoded

[Modify project compliance setting](#)

Graphical WSDL Editor



Graphical WSDL Editor



Source WSDL Editor

```
gettime.wsdl x
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soa
  <wsdl:types>
    <xsd:schema targetNamespace="http://www.example.org/gettime/" xmlns:x
      <xsd:element name="NewOperation">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="in" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="NewOperationResponse">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="out" type="xsd:string"/>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:schema>
  </wsdl:types>
  <wsdl:message name="NewOperationRequest">
    <wsdl:part element="tns:NewOperation" name="parameters"/>
  </wsdl:message>
```

The WSDL Header

```
<?xml version="1.0" encoding="UTF-8"?>  
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"  
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"  
  xmlns:tns="http://9.57.13.211/gettime"  
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
  name="tpfservice"  
  targetNamespace="http://9.57.13.211/gettime">
```

foo

XXX



foo

XXX



foo

XXX



foo

XXX



foo

XXX



foo

XXX



foo

XXX



The WSDL Types

```
<wsdl:types>
  <xsd:schema targetNamespace="http://9.57.13.211/gettime"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    xmlns:tns="http://9.57.13.211/gettime"
    xmlns:m="http://www.ibm.com/tpf">
    <xsd:element name="gettimerequest">
      <xsd:complexType/>
    </xsd:element>
    <xsd:element name="gettimeresponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="hours" type="xsd:string"/>
          <xsd:element name="minutes" type="xsd:string"/>
          <xsd:element name="seconds" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wsdl:types>
```

The WSDL Messages And Port

```
<wsdl:message name="gettimerequest">  
  <wsdl:part name="gettimerequest" element="tns:gettimerequest"/>  
</wsdl:message>  
  
<wsdl:message name="gettimeresponse">  
  <wsdl:part name="gettimeresponse" element="tns:gettimeresponse"/>  
</wsdl:message>  
  
<wsdl:portType name="gettimeport">  
  <wsdl:operation name="gettime">  
    <wsdl:input message="tns:gettimerequest"/>  
    <wsdl:output message="tns:gettimeresponse"/>  
  </wsdl:operation>  
</wsdl:portType>
```

The WSDL Binding and Service

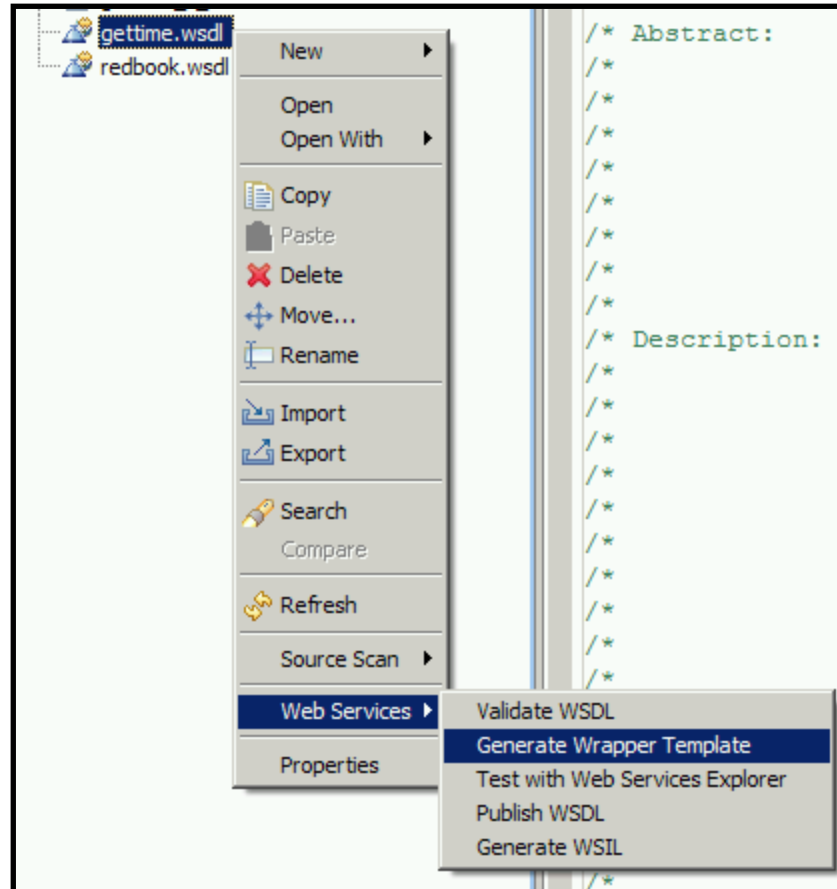
```
<wSDL:binding name="gettimesoap" type="tns:gettimeport">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
    style="document"/>
  <wSDL:operation name="gettime">
    <soap:operation soapAction="gettime"/>
    <wSDL:input>
      <soap:body use="literal"/>
    </wSDL:input>
    <wSDL:output>
      <soap:body use="literal"/>
    </wSDL:output>
  </wSDL:operation>
</wSDL:binding>

<wSDL:service name="gettimeservice">
  <wSDL:port binding="tns:gettimesoap" name="gettime">
    <soap:address location="http://9.57.13.211/gettime"/>
  </wSDL:port>
</wSDL:service>

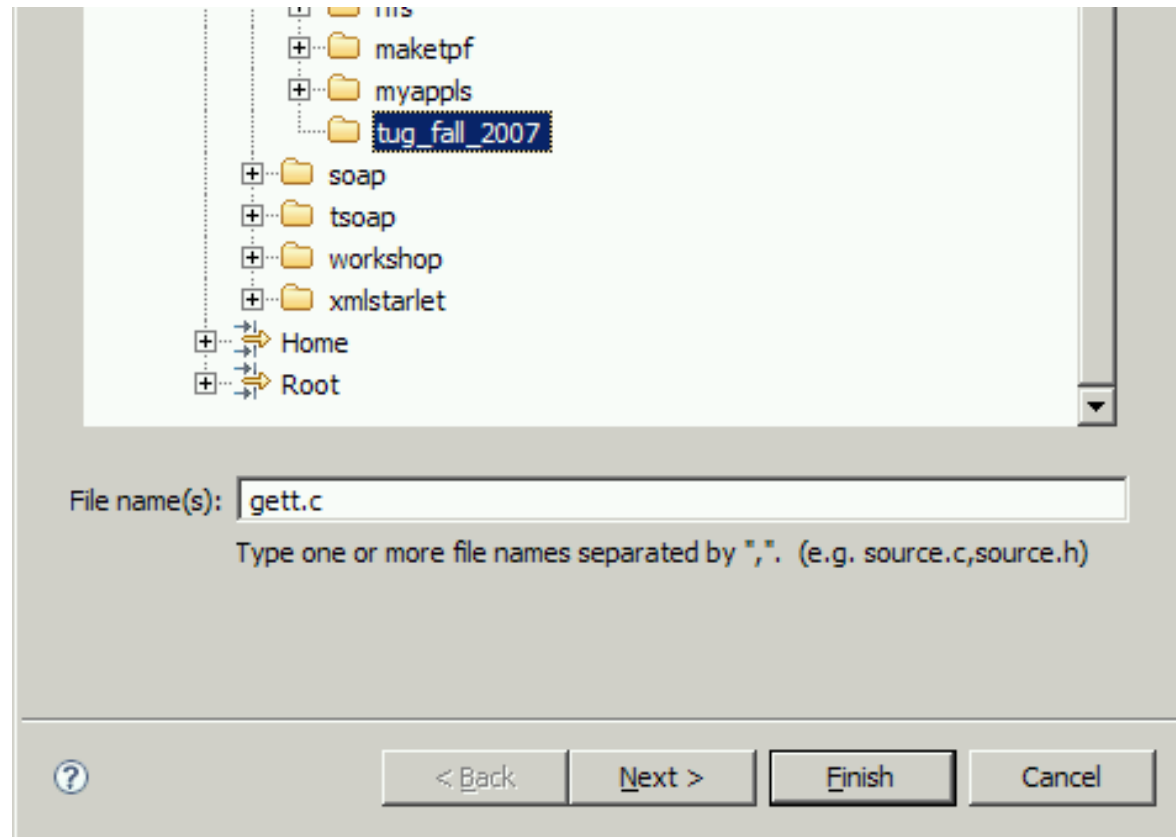
</wSDL:definitions>
```

Generating The Web Service Wrapper Template

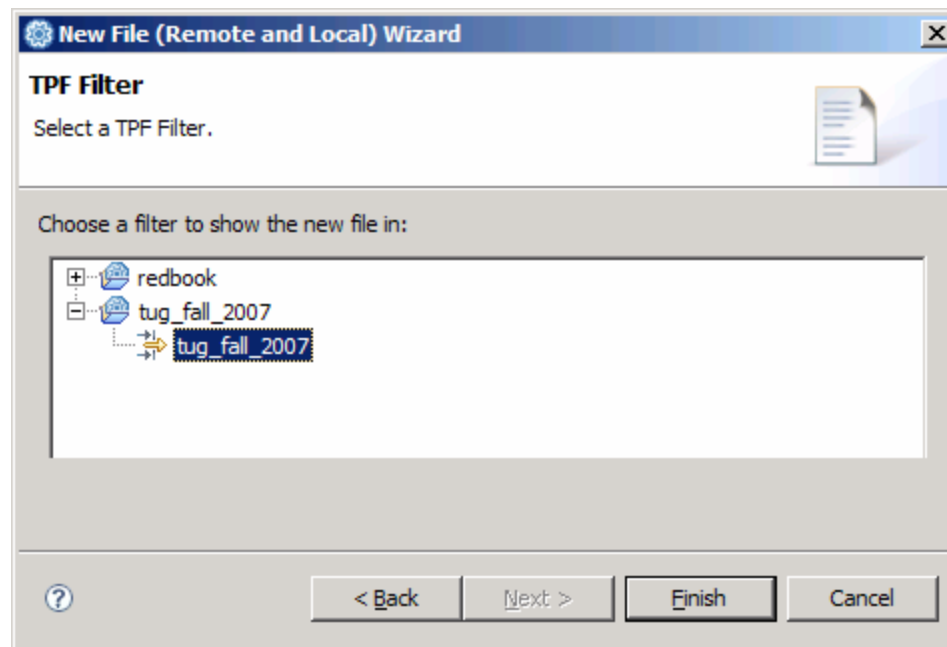
Generate The Wrapper Template



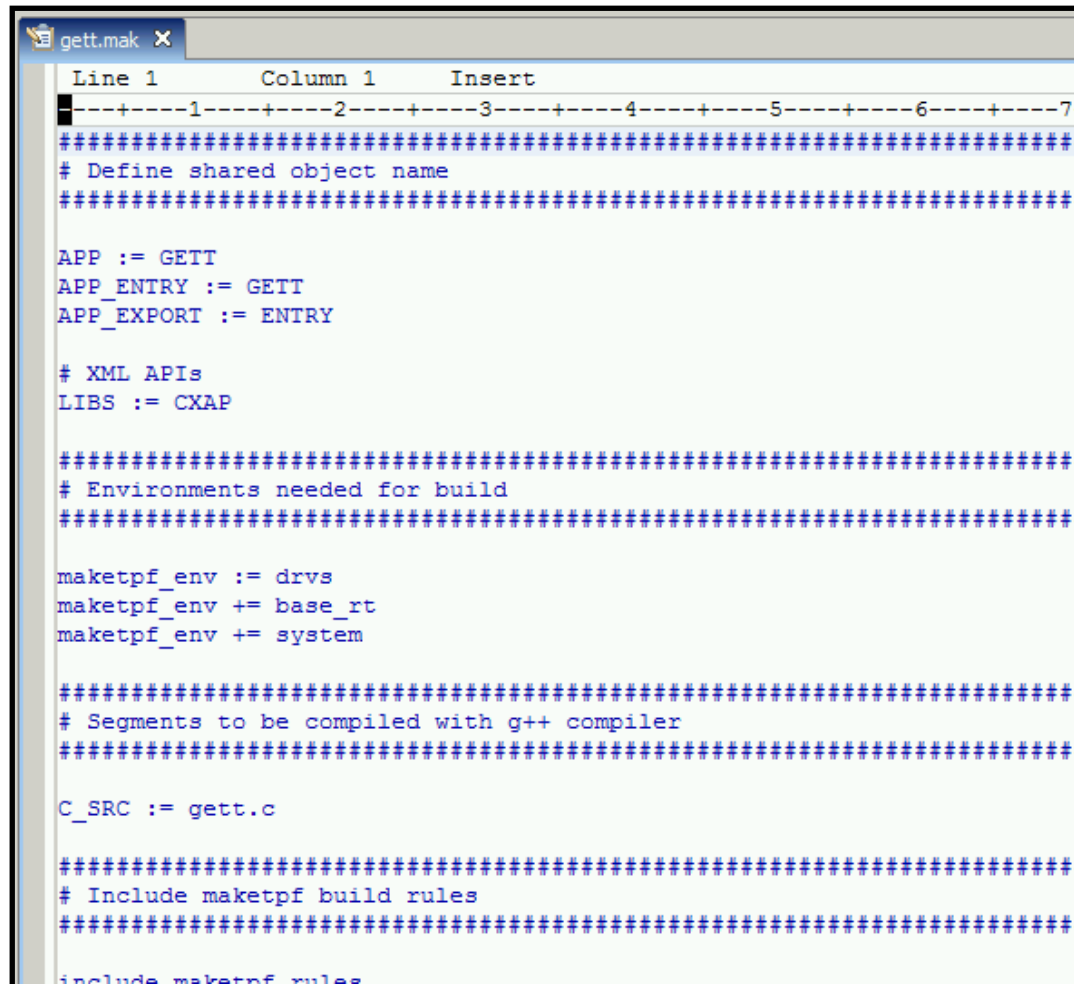
Name The Wrapper Template



Pick A Filter For The Wrapper Template



The Wrapper Makefile



```
gett.mak x
Line 1      Column 1      Insert
-----1-----2-----3-----4-----5-----6-----7-----
#####
# Define shared object name
#####

APP := GETT
APP_ENTRY := GETT
APP_EXPORT := ENTRY

# XML APIs
LIBS := CXAP

#####
# Environments needed for build
#####

maketpf_env := drvs
maketpf_env += base_rt
maketpf_env += system

#####
# Segments to be compiled with g++ compiler
#####

C_SRC := gett.c

#####
# Include maketpf build rules
#####
include maketpf.rules
```

In The Wrapper Template

```
switch (operation) {
  case OPERATION1:
    /* *****
    /* Use the TPF XML APIs to get the parameter data      */
    /* for operation1                                     */
    /* *****

    /* *****
    /* Invoke the TPF application that implements          */
    /* operation1.                                         */
    /* *****

    /* *****
    /* Based on the response from the TPF application,    */
    /* build the Body portion of the SOAP response message */
    /* and store a pointer to the TPF XML API handle in   */
    /* tpfSoapMsgCtx->outputMsgHdl                        */
    /* *****

  }

  // Return to SOAP Handler
  return SendReply;
}
```

The Modified Wrapper Makefile

```
#####  
# Segments to be compiled with g++ compiler  
#####  
  
C_SRC := gett.c  
CXX_SRC := gettime.cpp # Edwin
```

The Modified Wrapper Template

```
switch (operation) {
case OPERATION1:
/*****/
/* Use the TPF XML APIs to get the parameter data      */
/* for operation1                                     */
/*****/

/*****/
/* Invoke the TPF application that implements          */
/* operation1.                                       */
/*****/

/*****/
/* Based on the response from the TPF application,    */
/* build the Body portion of the SOAP response message */
/* and store a pointer to the TPF XML API handle in   */
/* tpfSoapMsgCtx->outputMsgHdl                       */
/*****/

getTimeAndBuildResponse(msgCtx->response_handle.xml_handle); // Edwin

msgCtx->response_handle.type = TPF_HDL_RESP;
return SendReply;
}

// Return to SOAP Handler
return SendReply;
}
```

Program Flow

CSO GETT
gettime.cpp

Register with PCRS

- Program = CHA1
- Command = ZDTIM

Route command via PCRS

Create event and wait

Unregister from PCRS

Build SOAP output message

Release system heap

CSO CHA1
outputcatcher.cpp

Get system heap

Copy ZDTIM output

Post event

gettime.h

```
#ifndef gettime_h
#define gettime_h

#include <tpf/c_node.h>

#ifdef __cplusplus
extern "C"
#endif
void getTimeAndBuildResponse(XMLHandle* h);

#endif
```

gettime.cpp

```
#include <gettime.h>

char* getTime();
char* getHours(const char* const s);
char* getMinutes(const char* const s);
char* getSeconds(const char* const s);

void getTimeAndBuildResponse(XMLHandle* h) {
    tpf_xml_initialize_handle(h, NO_PARSER, 0);
    tpf_xml_createXMLstructure(*h, "1.0", 819, "ISO-8859-1", STANDALONE_YES, 2,
        "Web Service Wrapper Sample", "gettimeresponse");
    tpf_xml_appendElement(*h, 0, "Envelope", 0, "SOAP-ENV",
        "http://schemas.xmlsoap.org/soap/envelope/", 0);
    tpf_xml_appendElement(*h, "Envelope", "Body", 0, "SOAP-ENV",
        "http://schemas.xmlsoap.org/soap/envelope/", 0);
    tpf_xml_appendElement(*h, "Body", "gettimeresponse", 0,
        "ns1", "http://9.57.13.211/gettime", 0);
}
```

gettime.cpp

```
char* t = getTime();
tpf_xml_appendElement(*h, "gettimeresponse", "hours", getHours(t),
                      "ns1", "http://9.57.13.211/gettime", 0);
tpf_xml_appendElement(*h, "gettimeresponse", "minutes", getMinutes(t),
                      "ns1", "http://9.57.13.211/gettime", 0);
tpf_xml_appendElement(*h, "gettimeresponse", "seconds", getSeconds(t),
                      "ns1", "http://9.57.13.211/gettime", 0);
}
```


gettime.cpp

```
char* getTime() {
    int lniata = 0xD00102;
    pcrs_input pi;
    pi.operation = PCRS_REGISTER;
    pi.lniata = lniata;
    strncpy(pi.pgm_name, "CHA1", sizeof "CHA1");
    PCRS_RC rc;
    if((rc = tpf_pcrs(&pi)) != PCRS_OK) {
        if(rc != PCRS_LNIATA_ALREADY_REG) {
            return 0;
        }
    }
    pi.operation = PCRS_ROUTECMD;
    pi.command = "ZDTIM";
    pi.cmd_len = strlen("ZDTIM");
    if(tpf_pcrs(&pi) != PCRS_OK) {
        return 0;
    }
}
```

gettime.cpp

```
ev0bk event;
memcpy(event.evnbnkn, "CHAL", sizeof "CHAL");
memcpy(event.evnbnkn+sizeof "CHAL", &lniata, sizeof lniata);
if(evntc(&event, EVENT_CB_DA, 'Y', 5, EVNTC_NORM) != 0) {
    return 0;
}
if(evnwc(&event, EVENT_CB_DA) != 0) {
    pi.operation = PCRS_UNREGISTER;
    tpf_pcrs(&pi);
    return 0;
}
pi.operation = PCRS_UNREGISTER;
tpf_pcrs(&pi);
if(!levtest(DA)) {
    return 0;
}
```

gettime.cpp

```
char* levelA = reinterpret_cast<char*>(ecbptr()->celcra);
char* systemHeapArea;
memcpy(&systemHeapArea,levelA, sizeof systemHeapArea);
int numberOfPages;
memcpy(&numberOfPages, levelA + sizeof systemHeapArea, sizeof numberOfPages);
    char heapToken[8];
memcpy(heapToken, levelA + sizeof systemHeapArea + sizeof numberOfPages,
    sizeof heapToken);
int lengthOfMessage;
memcpy(&lengthOfMessage, levelA + sizeof systemHeapArea +
    sizeof numberOfPages + sizeof heapToken, sizeof numberOfPages);
relcc(DA);
return systemHeapArea;
}
```

gettime.cpp

```
char* getHours(const char* const s) {
    char* p = strstr(s, "DTIM0001I") + 10; // Find hours
    *(p+2) = 0;                               // Zero terminate
    return p;
}

char* getMinutes(const char* const s) {
    char* p = strstr(s, "DTIM0001I") + 13; // Find minutes
    *(p+2) = 0;                               // Zero terminate
    return p;
}

char* getSeconds(const char* const s) {
    char* p = strstr(s, "DTIM0001I") + 16; // Find seconds
    *(p+2) = 0;                               // Zero terminate
    return p;
}
```

outputcatcher.cpp

```
#include <tpf/tpfapi.h>
#include <tpf/tpfio.h>
#include <tpf/c_syseq.h>
#include <tpf/sysapi.h>
#include <tpf/c_eb0eb.h>
#include <tpf/c_cinfc.h>
#include <stdlib.h>
#include <string.h>
#include <sstream>

extern "C" void CHA1() {
    if(!levtest(D0)) {
        exit(0);
    }
    am0sg* omsg = reinterpret_cast<am0sg*>(ecbptr()->celcr0);
    int lniata = omsg->am0lit;

    int magic = 5;    // sizeof am0sg::am0lit + sizeof am0sg::am0np1 +
                    // sizeof am0sg::am0np2
}
```

outputcatcher.cpp

```
std::ostringstream os;
os.write(&ormsg->am0txt , ormsg->am0cct-magic);

if(reinterpret_cast<unsigned int*>(ormsg->am0fch) != 0) {

    unsigned int* fileAddress = reinterpret_cast<unsigned int*>(ormsg->am0fch);
    char fileId[2];
    memcpy(fileId,ormsg->am0rid, sizeof fileId);
    relcc(D0);
}
```

outputcatcher.cpp

```
while(fileAddress != 0) {
omsg = reinterpret_cast<am0sg*>(find_record(D0,
      const_cast<unsigned int*>(fileAddress), fileId, '\\0', NOHOLD));

if(omsg == 0) {
    fileAddress = 0;
} else {
    os.write(&omsg->am0txt , omsg->am0cct-magic);
}

if(fileAddress != 0) {
    fileAddress = reinterpret_cast<unsigned int*>(omsg->am0fch);
    memcpy(fileId, omsg->am0rid, sizeof fileId);
}
crusa(1,D0);
}
}
```

outputcatcher.cpp

```
int numberOfPages = (os.rdbuf()->str().size() + _4K_SIZE) / _4K_SIZE;
char heapToken[8] = { 'E', 'D', 'W', 'I', 'N', '0', '0', '0'};
char* systemHeapArea = reinterpret_cast<char*>(gsysc(numberOfPages,heapToken));
if(systemHeapArea != 0) {
    cinfo(CINFO_WRITE,CINFO_CMMHEAP);
    memcpy(systemHeapArea, os.rdbuf()->str().c_str(),
           os.rdbuf()->str().size());
    keyrc();

    char* levelA = reinterpret_cast<char*>(getcc(DA,GETCC_TYPE,L4));

    memcpy(levelA, &systemHeapArea, sizeof systemHeapArea);
    memcpy(levelA + sizeof systemHeapArea, &numberOfPages,
           sizeof numberOfPages);
    memcpy(levelA + sizeof systemHeapArea + sizeof numberOfPages,
           heapToken, sizeof heapToken);
    int lengthOfMessage = os.rdbuf()->str().size();
    memcpy(levelA + sizeof systemHeapArea + sizeof numberOfPages +
           sizeof heapToken, &lengthOfMessage, sizeof lengthOfMessage);
```

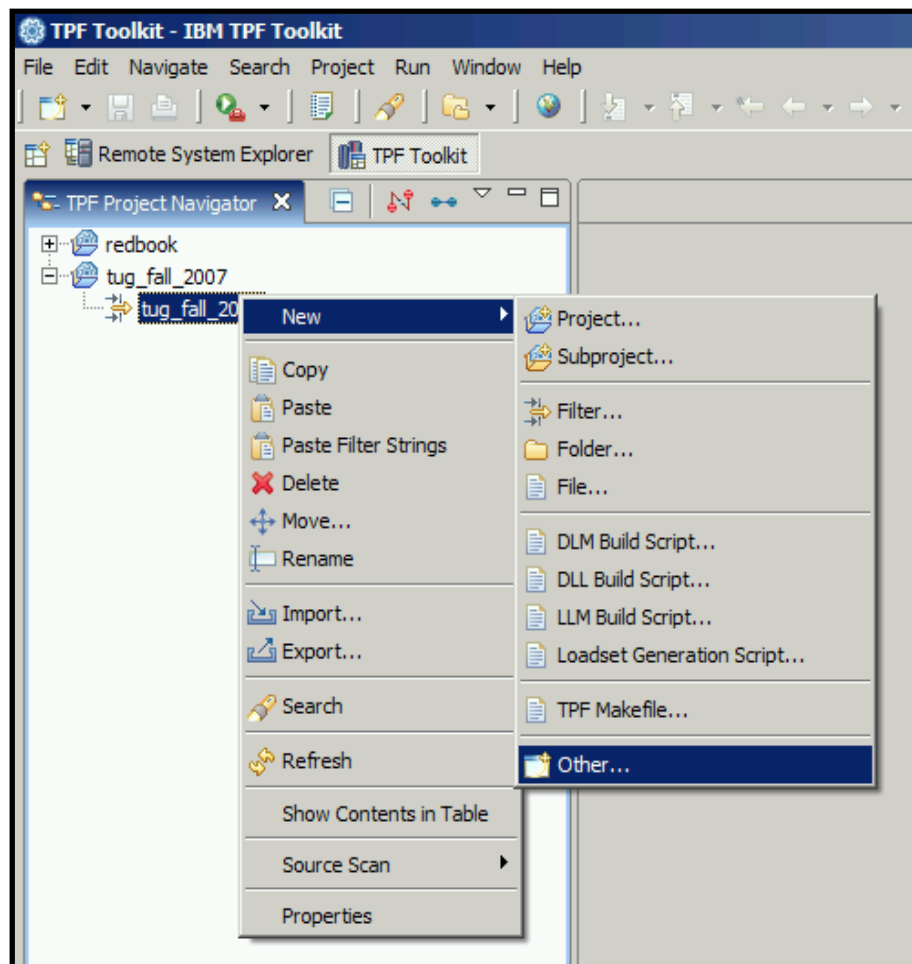

outputcatcher.cpp

```
    ev0bk event;  
    memcpy(event.evnbkn, "CHAL", sizeof "CHAL");  
    memcpy(event.evnbkn+sizeof "CHAL", &lniata, sizeof lniata);  
    postc(&event, EVENT_CB_DA, 0);  
}  
exit(0);  
}
```

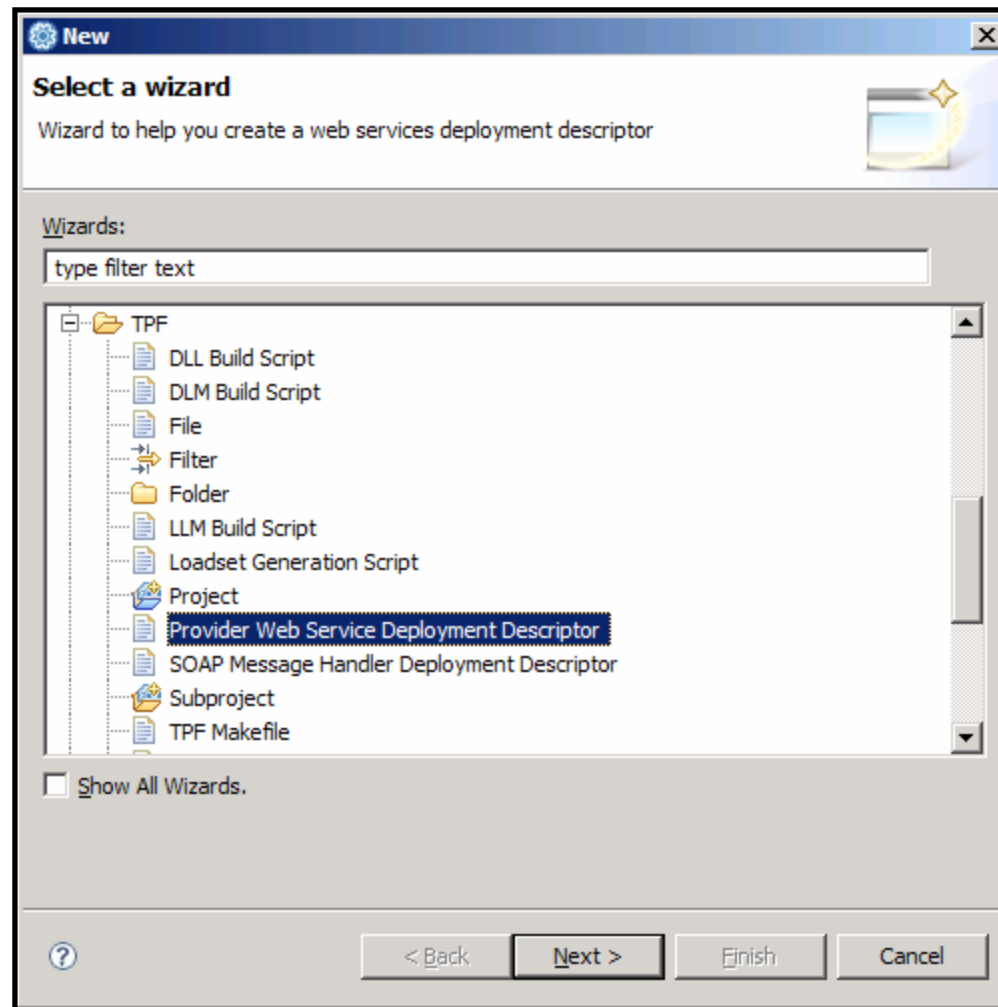
Creating The Provider Web Service Deployment Descriptor



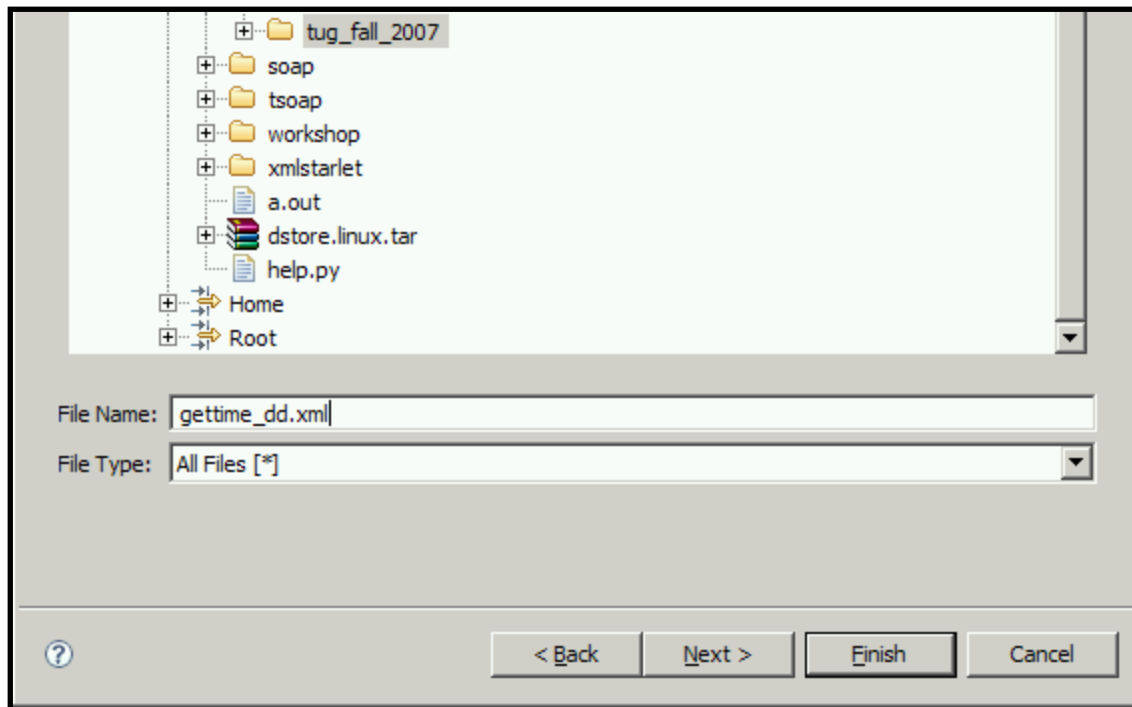
Create DD Step 1



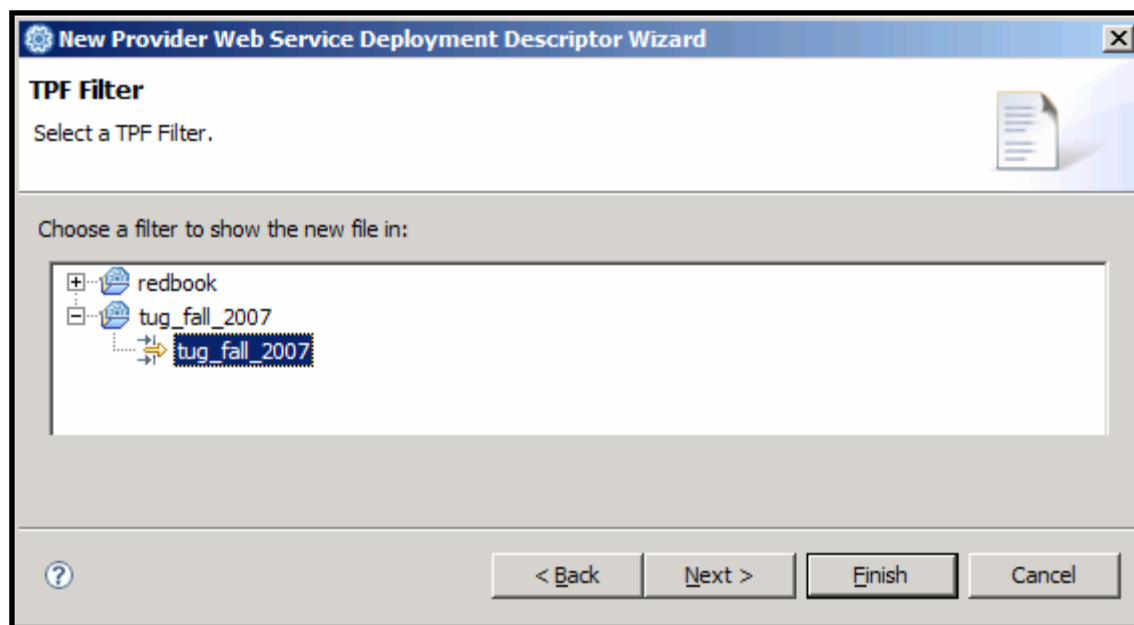
Create DD Step 2



Name The DD



Pick A Filter For The DD



Specify The DD Options

New Provider Web Service Deployment Descriptor Wizard

Deployment Descriptor
Specify web service details

Web service URI:

Web service wrapper program:

Web service description:

SOAP Processing
Specify the SOAP message handlers to invoke when a SOAP request is received:

Name	Required

Buttons: Add..., Edit..., Remove, Move Up, Move Down

Specify the applicable SOAP versions:

- SOAP 1.1
- SOAP 1.2
- BP 1.1_SSBP 1.0

Verify SOAP headers

Buttons: < Back, Next >, Finish, Cancel

The DD

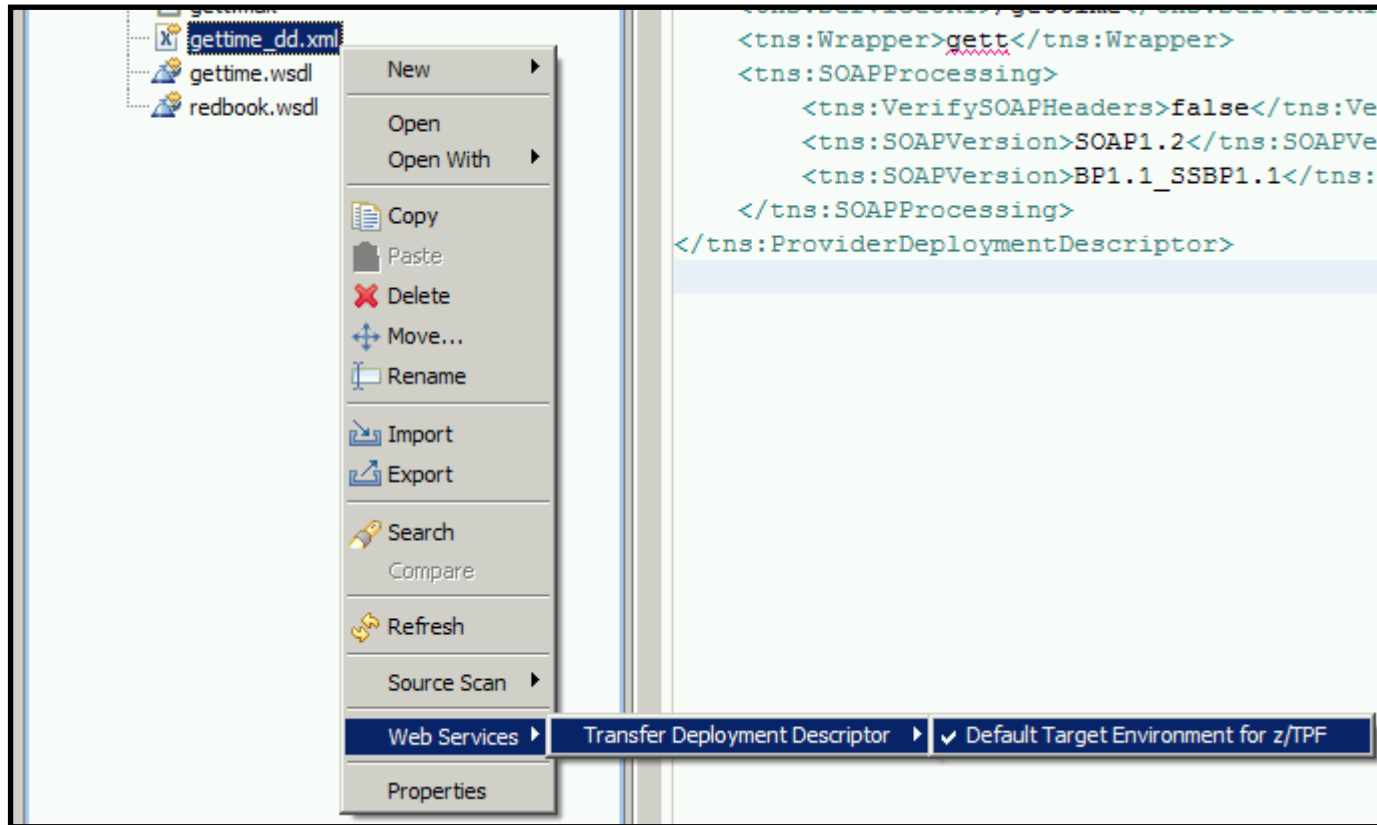
```
gettime_dd.xml x
<?xml version="1.0" encoding="UTF-8"?>
<tns:ProviderDeploymentDescriptor
  xmlns:tns="http://www.ibm.com/xmlns/prod/ztpf/soap"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/ztpf/soap file:/
  <tns:ServiceURI>/gettime</tns:ServiceURI>
  <tns:Wrapper>GETT</tns:Wrapper>
  <tns:SOAPProcessing>
    <tns:VerifySOAPHeaders>false</tns:VerifySOAPHeaders>
    <tns:SOAPVersion>SOAP1.2</tns:SOAPVersion>
    <tns:SOAPVersion>BP1.1_SSBP1.0</tns:SOAPVersion>
  </tns:SOAPProcessing>
</tns:ProviderDeploymentDescriptor>
```


The DD In XML Design View

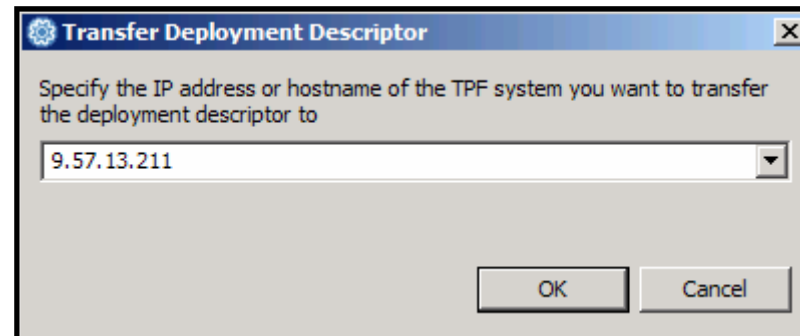
The screenshot shows the XML Design View of a Provider Deployment Descriptor (DD) file named 'gettime_dd.xml'. The view is organized into a tree structure with the following elements and their values:

Element	Value
xmlns:tns	http://www.ibm.com/xmlns/prod/ztpf/soap
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation	http://www.ibm.com/xmlns/prod/ztpf/soap file:/etc/tpf-ws/schema/tpf_ws_dd_v1.xsd
tns:ServiceURI	/gettime
tns:Wrapper	GETT
tns:SOAPProcessing	
tns:VerifySOAPHeaders	false
tns:SOAPVersion	SOAP 1.2
tns:SOAPVersion	BP1.1_SSBP1.0

Transfer The DD



Specify The System's Address



Provide FTP Login

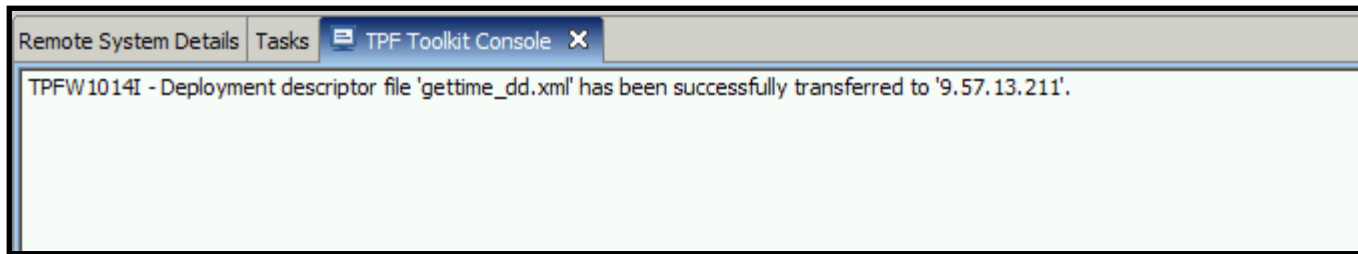
Enter Password

System type: TPF
Host name: 9.57.13.211
User ID: anonymous
Password: *****

Save user ID
 Save password

OK Cancel

Completion



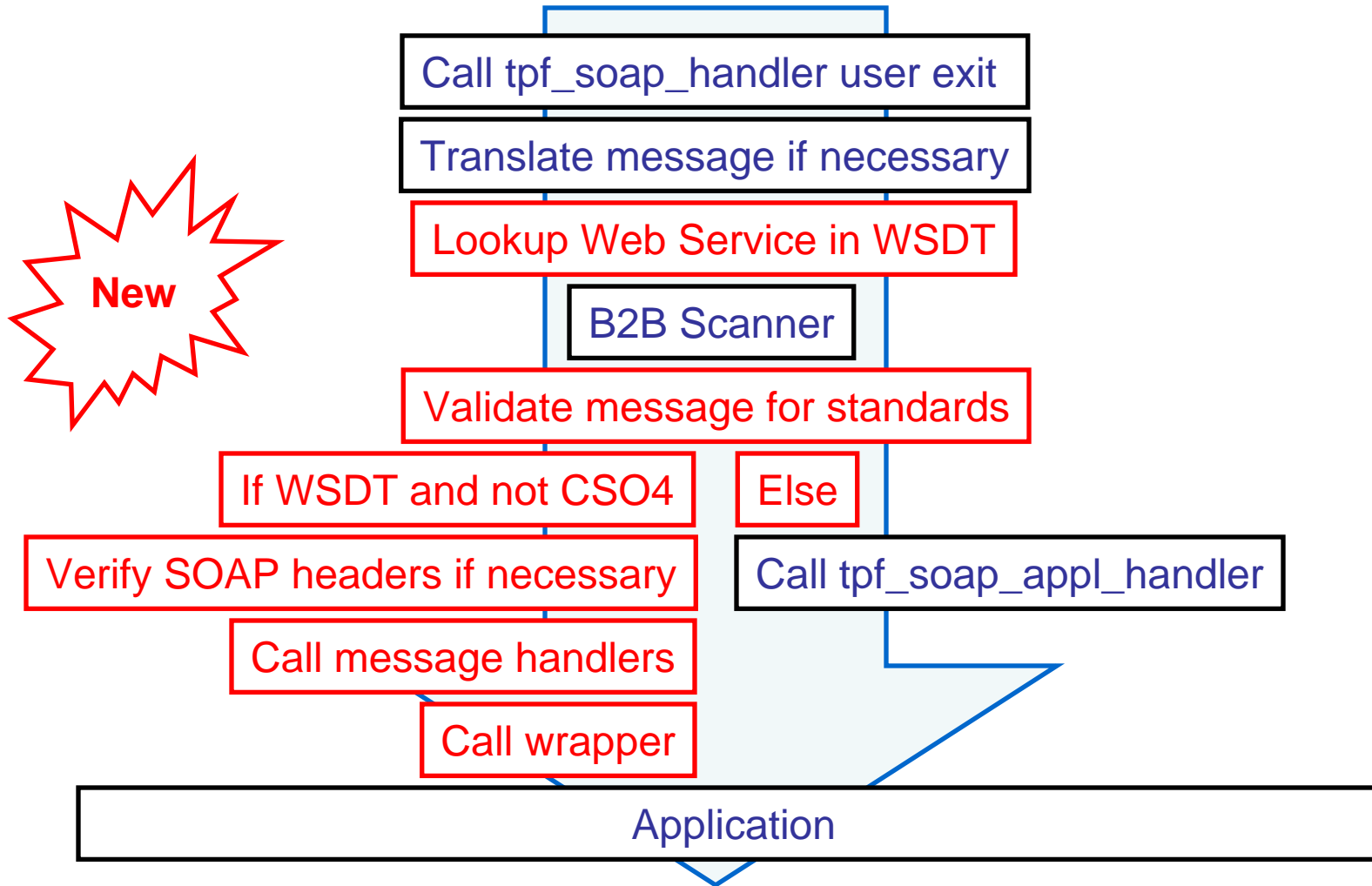
Deploying The Deployment Descriptors



The Web Services Deployment Table (WSDT)

- Registry of deployed Web services
 - Local to a TPF processor
 - Used by the SOAP runtime environment
- Managed with the ZWSAT command

New z/TPF SOAP Handler Flow



The /etc/tpf-ws Directory

- Deployment descriptors
- SOAP message handlers
- Schema definitions in /schema

The /etc/tpf-ws Directory

```
zfile ls -al
```

```
CSMP0097I 10.36.36 CPU-B SS-BSS SSU-HPN IS-01 _  
FILE0001I 10.36.36 START OF DISPLAY FROM ls -al  
total 15  
drwxrwxrwx 1 root bin 15 Sep 6 07:32 .  
drwxrwxrwx 1 root bin 18 Sep 6 07:31 ..  
drwxr-xr-x 1 root bin 7 Sep 7 10:18 .system  
-rw-r--r-- 1 ftp bin 592 Sep 6 07:32 logging.xml _  
drwxrwxrwx 1 root bin 3 Sep 6 07:31 schema  
-rw-r--r-- 1 ftp bin 820 Sep 6 07:32 zTPFCommandService.xml  
-rw-r--r-- 1 ftp bin 864 Sep 6 07:32 CalculatorService.xml  
-rw-r--r-- 1 ftp bin 1062 Sep 6 07:32 CalculatorService_logging.xml  
-rw-r--r-- 1 ftp bin 880 Sep 6 07:32 ContinuousCalculator.xml _  
-rw-r--r-- 1 ftp bin 874 Sep 6 07:32 ContinuousCalculatorDupl.xml  
-rw-r--r-- 1 ftp bin 1237 Sep 6 07:32 ContinuousHeaderCalculator.xml  
-rw-r--r-- 1 ftp bin 1061 Sep 6 07:32 ContinuousLoggingCalculator.xml  
-rw-r--r-- 1 ftp bin 1067 Sep 6 07:32 ContinuousLoggingCalculatorDupl.xml _  
-rw-r--r-- 1 ftp bin 1236 Sep 6 07:32 ContinuousVerifyCalculator.xml  
-rw-r--r-- 1 ftp bin 608 Sep 6 07:32 LoggingDupl.xml  
END OF DISPLAY+
```

Using The ZWSAT Command



Add a DD to the WSDT

```
zwsat add dd-gettime_dd.xml
```

```
CSMP0097I 10.53.42 CPU-B SS-BSS SSU-HPN IS-01
```

```
WSAT0037I 10.53.42 ADD OF /gettime SCHEDULED FOR PROCESSOR B+
```

```
CSMP0097I 10.53.43 CPU-B SS-BSS SSU-HPN IS-01
```

```
WSAT0017I 10.53.43 WEB SERVICE /gettime ADDED SUCCESSFULLY ON PROCESSOR B AS REQUESTED BY B+
```

Display the WSDT

```
zwsat display all
CSMP0097I 10.54.58 CPU-B SS-BSS SSU-HPN IS-01
WSAT0016I 10.54.58 START OF DISPLAY FOR ALL RESOURCES

PROCESSOR: B

RESOURCE NAME TYPE STATUS
-----
/gettime PROVIDER UNDEPLOYED

/ContinuousCalculatorService PROVIDER DEPLOYED

END OF DISPLAY+
```

Display one WSDT Entry

```
zwsat display dd-gettime_dd.xml
CSMP0097I 12.20.09 CPU-B SS-BSS SSU-HPN IS-01
WSAT0014I 12.20.09 START OF DISPLAY FOR DEPLOYMENT DESCRIPTOR gettime_dd.xml

DEPLOYMENT DESCRIPTOR: /etc/tpf-ws/gettime_dd.xml
TYPE: PROVIDER
URI: /gettime
WRAPPER: GETT _
OPERATIONS:
SOAP VERSION: SOAP 1.2
BP 1.1/SSBP 1.0
VERIFY SOAP HEADERS: FALSE
STATUS: UNDEPLOYED
PROCESSOR: B
DESCRIPTION:

END OF DISPLAY.+
```

Deploy a Web Service

```
zwsat deploy dd-gettime_dd.xml
CSMP0097I 14.39.33 CPU-B SS-BSS SSU-HPN IS-01
WSAT0037I 14.39.33 DEPLOY OF /gettime SCHEDULED FOR PROCESSOR B+
CSMP0097I 14.39.33 CPU-B SS-BSS SSU-HPN IS-01
WSAT0001I 14.39.33 WEB SERVICE /gettime DEPLOYED SUCCESSFULLY ON PROCESSOR B AS
REQUESTED BY B+
```

- Note: Deploying fails if the wrapper program is not loaded and active yet

Display one WSDT Entry

```
zwsat display dd-gettime_dd.xml
```

```
CSMP0097I 12.20.09 CPU-B SS-BSS SSU-HPN IS-01
```

```
WSAT0014I 12.20.09 START OF DISPLAY FOR DEPLOYMENT DESCRIPTOR gettime_dd.xml
```

```
DEPLOYMENT DESCRIPTOR: /etc/tpf-ws/gettime_dd.xml
```

```
TYPE: PROVIDER
```

```
URI: /gettime
```

```
WRAPPER: GETT _
```

```
OPERATIONS:
```

```
SOAP VERSION: SOAP 1.2
```

```
BP 1.1/SSBP 1.0
```

```
VERIFY SOAP HEADERS: FALSE
```

```
STATUS: DEPLOYED
```

```
PROCESSOR: B
```

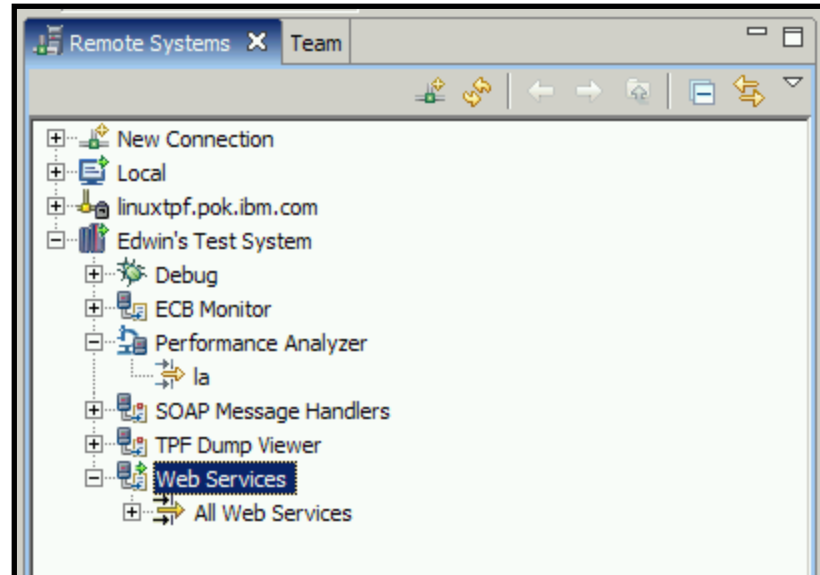
```
DESCRIPTION:
```

```
END OF DISPLAY.+
```

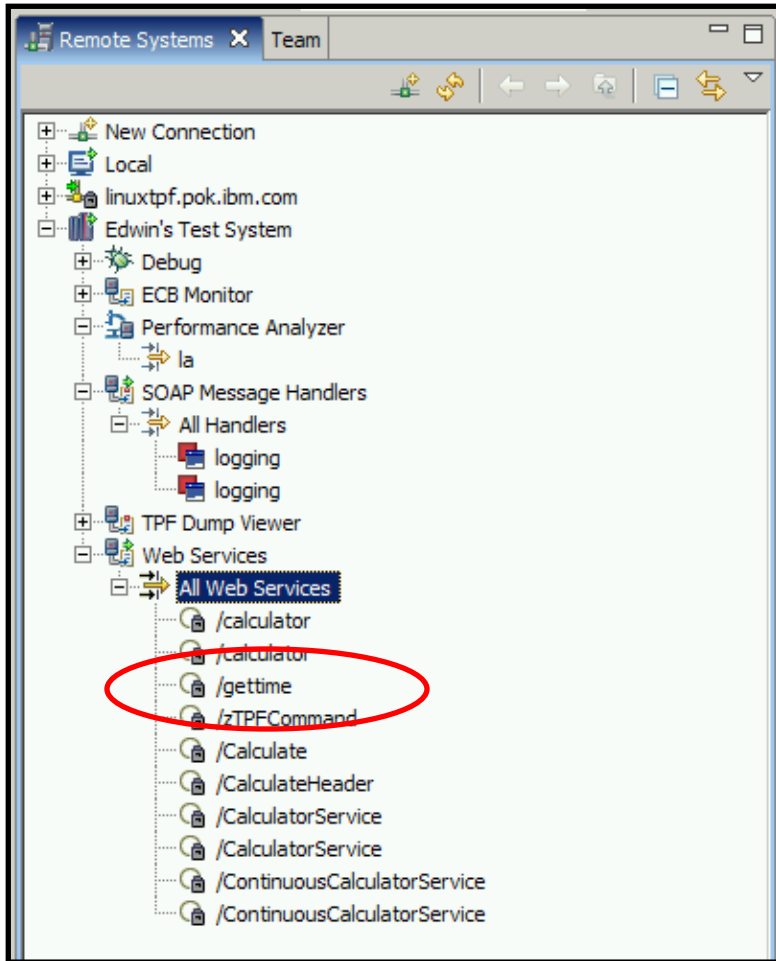

Using The Web Services Explorer Instead



Open the Web Services Subsystem



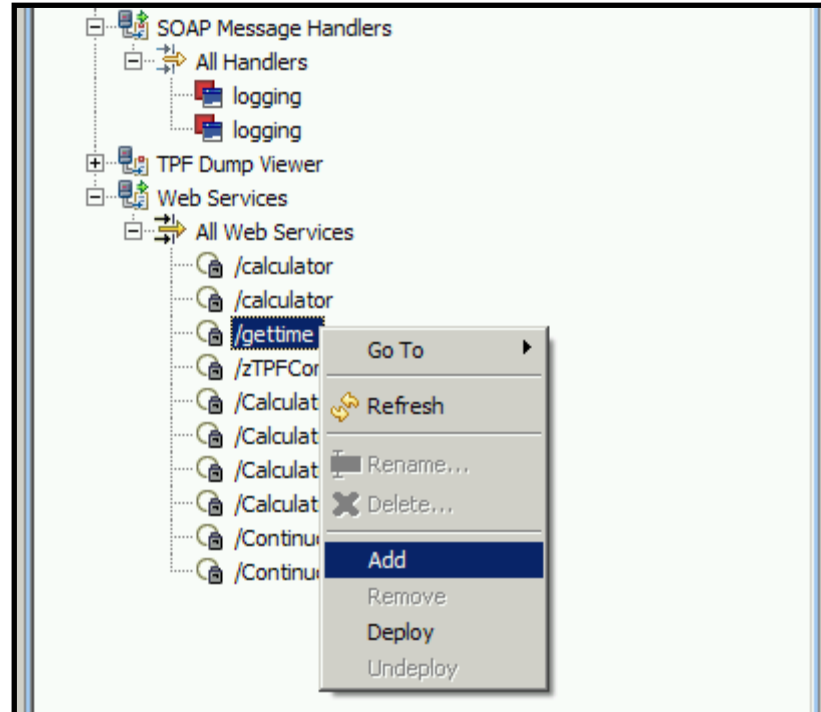
Open the All Web Services Filter



The screenshot shows the Properties window for the selected service. The window is titled 'Remote Scratchpad' and displays a table of properties and their values.

Property	Value
Misc	
DD Name	gettime_dd.xml
Operation	
Status	Deployed
Subsystem	BSS
Type	Provider
URI	/gettime
Wrapper Program	GETT
SOAP Processing	
SOAP message handler	
SOAP version	[SOAP 1.2, BP 1.1 SSBP 1.0]
Verify SOAP header	false

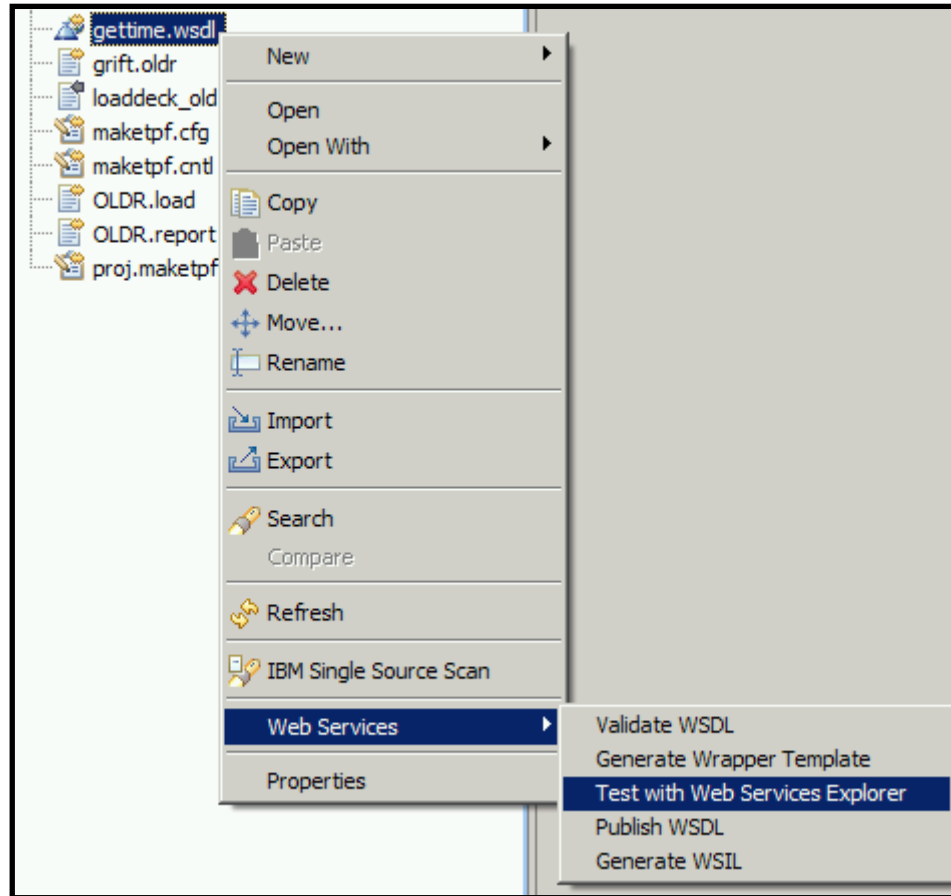
Add, Remove, Deploy, Undeploy



Testing A Web Service



Test With Web Services Explorer



Web Services Explorer

Web Services Explorer - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:59844/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.exp

Web Services Explorer

Navigator

- WSDL Main
 - file:/C:/Program Files/IBM/tpfservice
 - timeportbinding
 - gettime

Actions

WSDL Binding Details

Shown below are the details for this **SOAP** <binding> element. Click on an operation to fill in its parameters and invoke it or specify additional endpoints.

Operations

Name	Documentation
gettime	--

Endpoints [Add](#) [Remove](#)

Endpoints

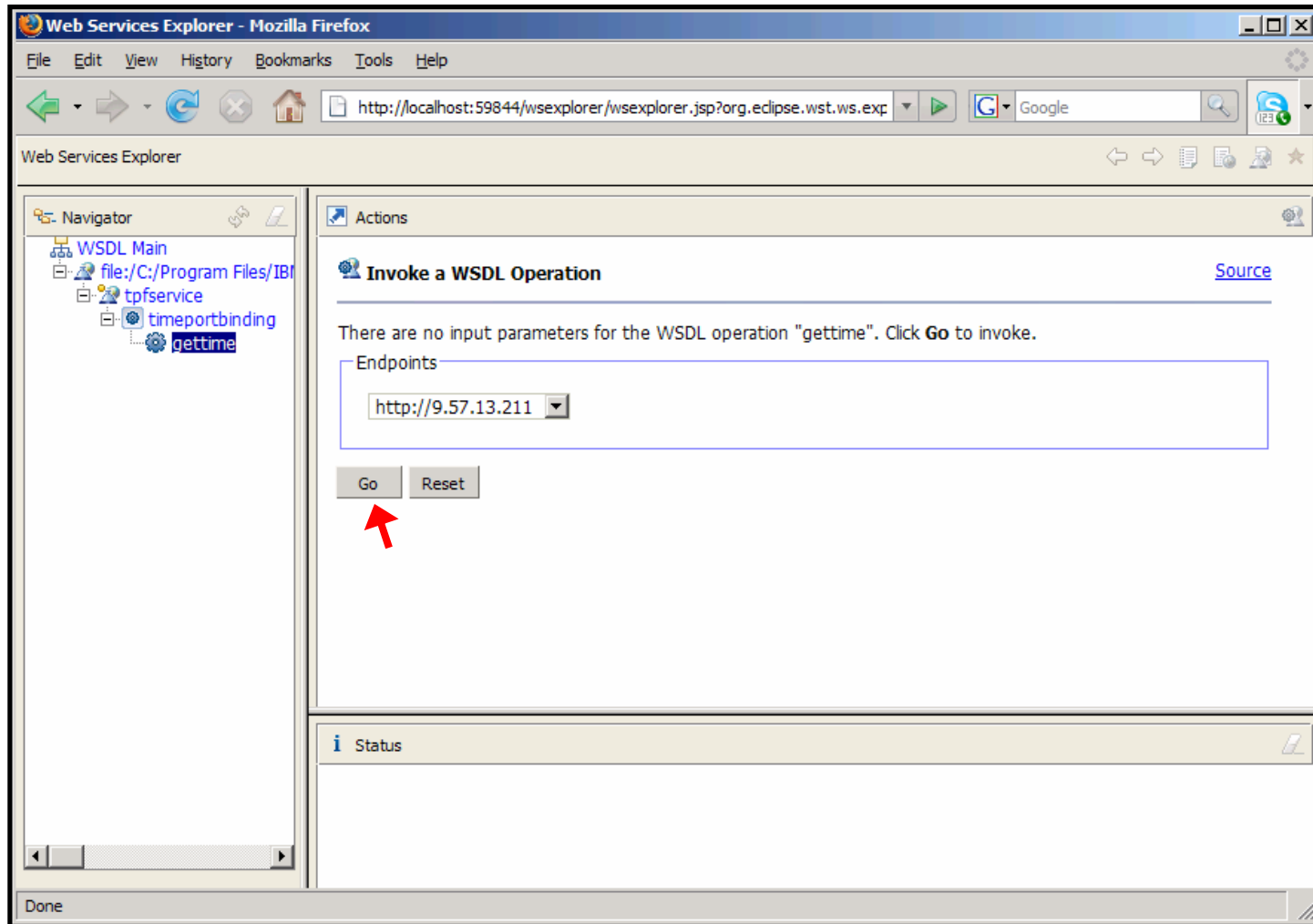
- http://9.57.13.211

Go Reset

Status

Done

Web Services Explorer



Web Services Explorer

Web Services Explorer - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost:61848/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.expl

Google

http://9.57.13.211/gettime.wsdl

Web Services Explorer

Navigator

- WSDL Main
 - file:/C:/Program Files/IBM/TPF Toolkit
 - gettimeservice
 - gettimesoap
 - gettime

Actions

Invoke a WSDL Operation [Source](#)

There are no input parameters for the WSDL operation "gettime". Click **Go** to invoke.

Endpoints

http://9.57.13.211/gettime

Go Reset

Status [Source](#)

- gettimerservice
 - hours (string): 15
 - minutes (string): 20
 - seconds (string): 22

Done

Source View

i Status

▼ SOAP Request Envelope:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <soapenv:Envelope>
  - <soapenv:Body>
    <q0:gettimerequest/>
  </soapenv:Body>
</soapenv:Envelope>
```

▼ SOAP Response Envelope:

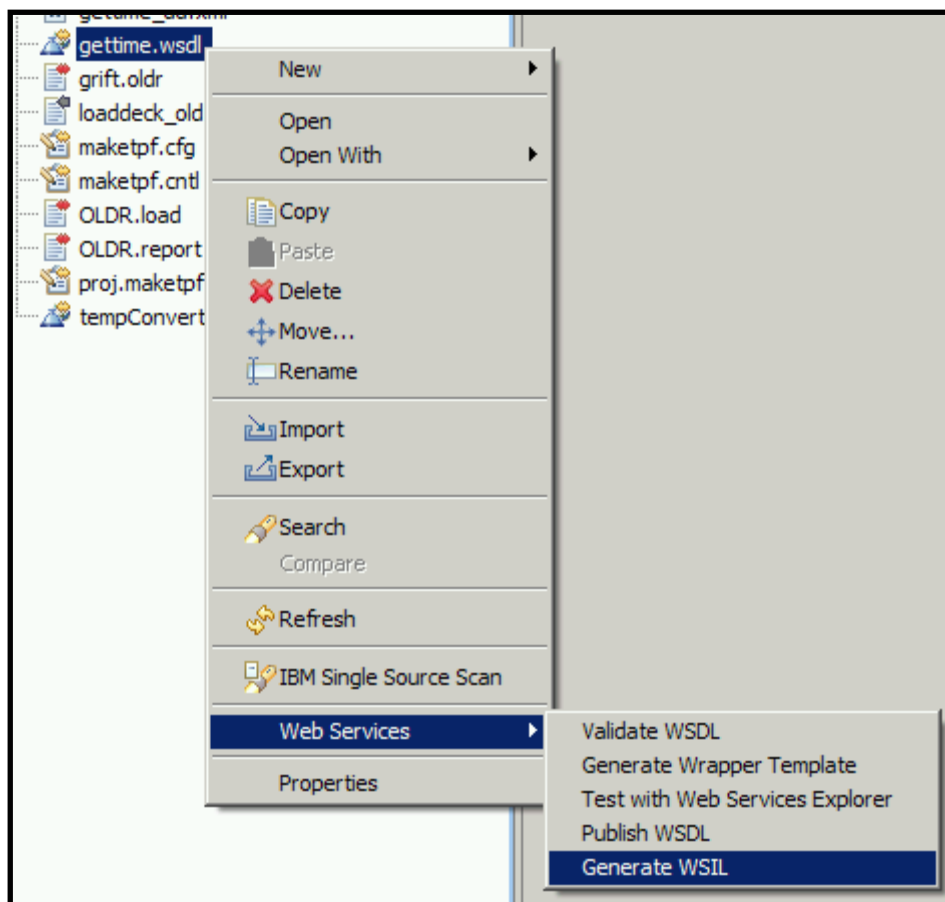
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <SOAP-ENV:Envelope>
  - <SOAP-ENV:Body>
    - <ns1:gettimereponse>
      <ns1:hours>15</ns1:hours>
      <ns1:minutes>20</ns1:minutes>
      <ns1:seconds>22</ns1:seconds>
    </ns1:gettimereponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

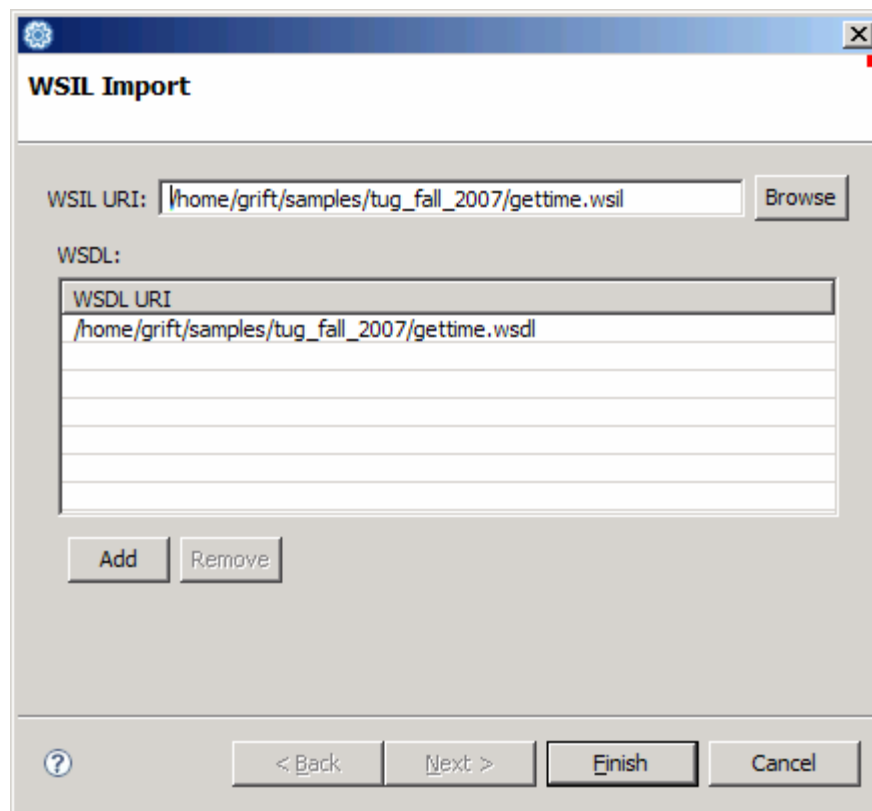
Generating The WSIL File



Generate The WSIL File



Specify The URIs



The WSIL File

```
<?xml version="1.0" encoding="UTF-8"?>
<inspection xmlns="http://schemas.xmlsoap.org/ws/2001/10/inspection/"
  xmlns:wsilwsdl="http://schemas.xmlsoap.org/ws/2001/10/inspection/wsdl/"
  xmlns:wsiluddi="http://schemas.xmlsoap.org/ws/2001/10/inspection/uddi/"
  xmlns:uddi="urn:uddi-org:api">
  <service>
    <abstract xml:lang="en-US">
    </abstract>
    <description referencedNamespace="http://schemas.xmlsoap.org/wsdl/"
      location="http://9.57.13.211/gettime.wsdl">
      <wsilwsdl:reference endpointPresent="true">
        <wsilwsdl:referencedService xmlns:impl="http://9.57.13.211/gettime">
          impl:tpfservice
        </wsilwsdl:referencedService>
      </wsilwsdl:reference>
    </description>
  </service>
</inspection>
```

Travel Home Safely!

Edwin W. van de Grift

edwinvandegrift@us.ibm.com

IBM Software Group

Application & Integration Middleware



Trademarks

- IBM and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Other company, product, or service names may be trademarks or service marks of others.

Notes

- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.