



## **An Introspective Look at Emerging Solutions for Web Services**

Author: Dan Finerty  
[dFinerty@InnerAccessTech.com](mailto:dFinerty@InnerAccessTech.com)  
September 2002

An Introspective Look at Emerging.....	1
Solutions for Web Services.....	1
Sooner or later someone does it right .....	3
z/Services Architecture .....	5
z/Services Server.....	5
Introspection Services.....	5
Execution Services.....	6
Terminal API .....	6
Development Suite.....	7
Studio .....	7
Map Browser.....	7
Wizard.....	8
Scribe .....	9
Web Terminal .....	10
XML Console.....	11
Web Services .....	11
z/Services Recommendations .....	14
Application Selection.....	14

## Sooner or later someone does it right

### An InnerAccess Technologies White Paper on z/Services

Why is everyone looking at Web Services? Evans Data<sup>1</sup> reports that 70% of IT organizations are either currently implementing or planning to implement web services within the next 12 months. The simple truth is that web services is one vehicle that can bring to bear the benefits promised during the technology heydays of the dot-com frenzy. However, AMR Research<sup>2</sup> shows that only 5% have actually deployed a Web Services application.

Early adopters of Web Services are examining different application areas where true competitive advantage can be gained by exposing enterprise systems throughout both their own organization and throughout the entire supply chain. Typical applications include Business to Employee (B2E), Customer Self Service (CRM), and Supply Chain Management (SCM). Transforming vision into reality is the challenge.

Aberdeen Group<sup>3</sup> indicates that Web services are on the critical path for IT departments. Included in Aberdeen's conclusions is the suggestion that legacy applications should be serviced first. The most important concerns of Enterprise IT Managers according to Gartner<sup>4</sup> include:

- The availability of skills (43%)
- Web development (31%)
- Integration with legacies (47%)

A further complication faced by IT organizations is the continuing pressure from the business to control costs, improve efficiencies, and improve delivery cycles with an ROI in less than 12 months. The z/Services vision delivers on the promise of rapidly developed and deployed Web Services.

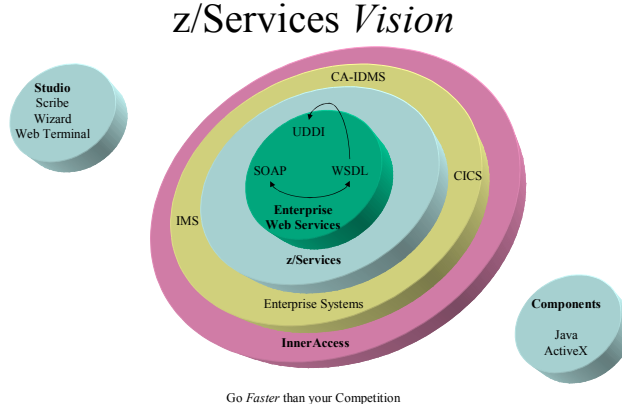
---

<sup>1</sup> EDC "North American Developer Survey 2002 Vol. 1"

<sup>2</sup> Peter Urban, AMR Research, "Web Service Vendor Hype Conflicts with User Feedback", ebizChronicle.com September 9, 2002

<sup>3</sup> Aberdeen Group "Web Services Development: On IT's Critical Path", August 2002

<sup>4</sup> Gartner Group, "The Most Important Concerns of Enterprise IT Managers", Joseph Feiman, August 2002



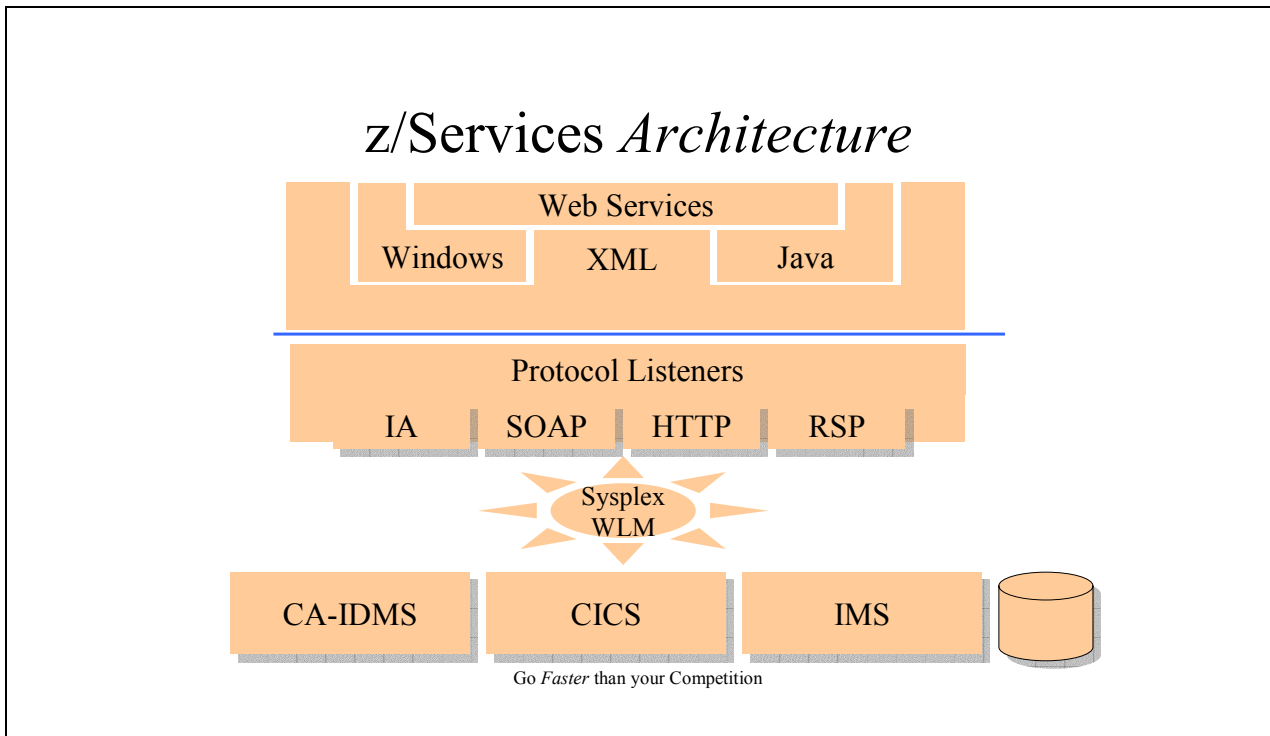
The z/Services vision has evolved over the life cycle of our products. The foundation products provide enterprise connectivity and extensions, InnerAccess transforms enterprise transactions into components, and z/Services publishes components as web services. Adjacent to these connectivity and extension services is the z/Services studio that enables enterprise programming expertise to be deployed on web enablement projects.

z/Services has 3 major facets:

- Components to deliver enterprise access to web based or PC based applications
- Development tools that enable enterprise programmers (CICS, IMS TM, CA-IDMS) to deliver web services based applications without developing Java and Windows programming expertise
- z/Services Server resides on the enterprise server and provides
  - Introspection Services
  - Execution Services
  - Web Services
    - Automatic WSDL generation
    - SOAP listener
    - Terminal API
    - Security infrastructure

## **z/Services Architecture**

### **z/Services Server**

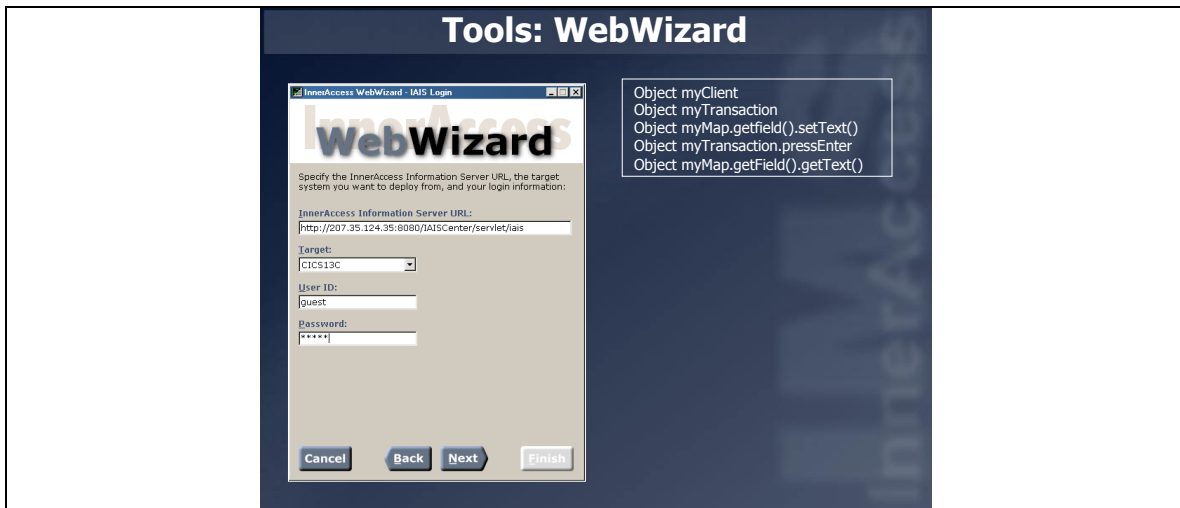


z/Services server, the component that resides on the enterprise mainframe provides a number of pieces that are critical to the web services puzzle. The server responds to component requests for Introspection Services and also provides execution services to components. The SOAP listener provides Simple Object Access Protocol support to provide the remote procedure interface for remote applications. Rexx Server Pages (RSP) support enables the operations staff to automate enterprise tasks and for the z/Services server to be managed. Rexx Server Pages can also be used to deliver HTTP or XML content from datasets or other mainframe data sources.

### **Introspection Services**

Screen introspection is a patent pending process designed to automatically discover the transactions, communication areas, map sets, and maps used by your online systems. The z/Services Studio provides the developer with complete access to all transactions in production through z/Services introspection.

The Studio Wizard publishes the transaction by integrating one of the templates and the traditional 3270 interface to create an enhanced web interface. A deployed Java Server Page (JSP) requires 5-6 lines of code to be deployed. The Studio Wizard enables the developer to publish a single transaction, selected transactions, or every transaction in the production environment.



**z/Services Web Wizard automatically generates ASP/JSP pages for CICS, IMS, and IDMS transactions.**

## Execution Services

z/Services provides execution services through both XML and SOAP interfaces. z/Services support for CICS, IMS and IDMS transactions does not require 3270 emulation nor screen scraping. z/Services provides 4 primary interfaces to support transactional services from your enterprise complex:

- CICS-BMS transactions: extensions to the 3270 Bridge have been written to provide enhanced functionality, provide field level access, and to avoid the extended path lengths and resource consumption associated with screen scraping
- CICS-Comm Area transactions: EXCI is used to provide non-BMS transaction support and to provide linkage from the SOAP listener to the CICS region
- IMS transactions: complementary functions similar to MFS provide OTMA connectivity to IMS systems
- IDMS transactions: the z/Services for IDMS adapter provides functions similar to the 3270 Bridge exit for CICS.

Additional capabilities have been designed into the execution services architecture that will be implemented in the near future including:

- Native 2 phase commit support
- Native WLM (Work Load Manager) support for scalability and throughput for z/Services throughout the enterprise sysplex

## Terminal API

The z/Services Terminal API provides development support for 3 key functions:

1. development support for examining the native visual transaction interface using Introspection Services

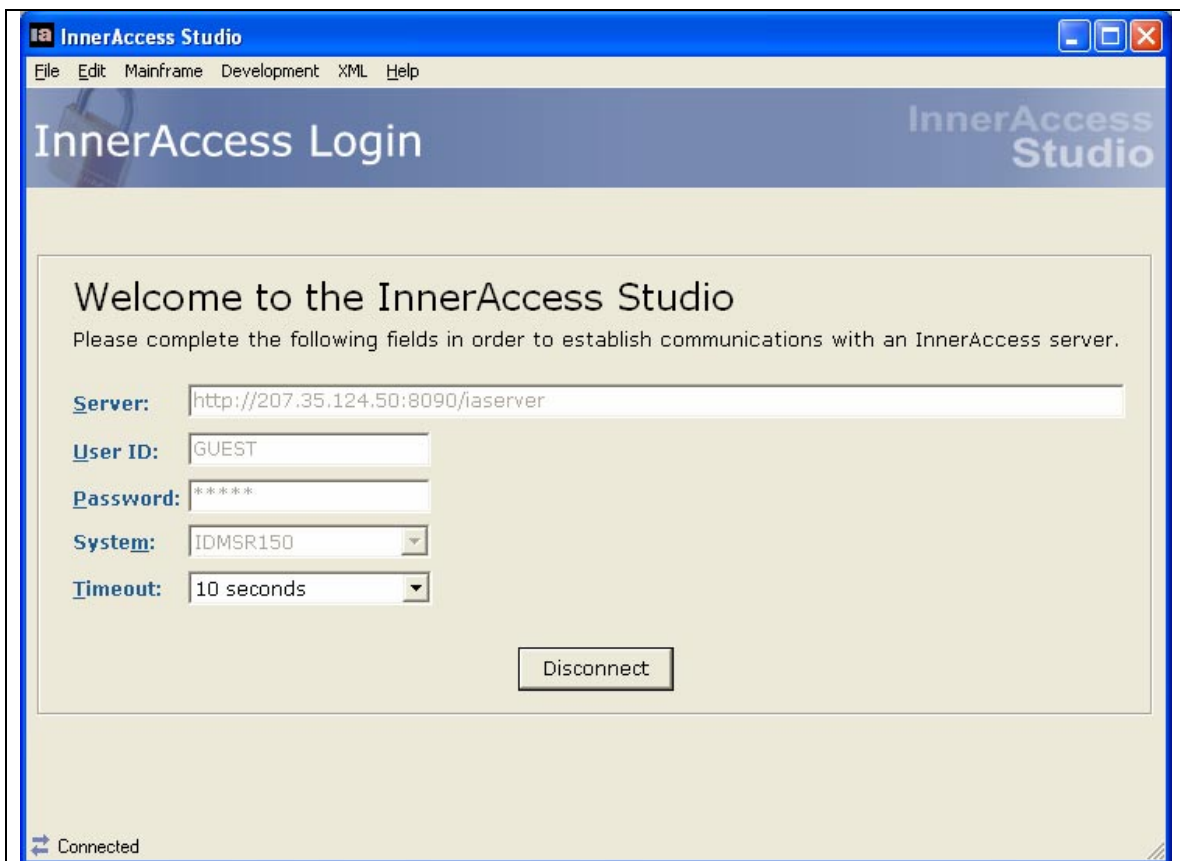
2. support the z/Services Scribe for combining multiple transactions into a single highly customized component
3. provides the programmatic interface to access transactions, maps, and fields. This reduces the traditional 3 component model into a single component and API for all CICS, IMS, and IDMS transactions.

## Development Suite

### Studio

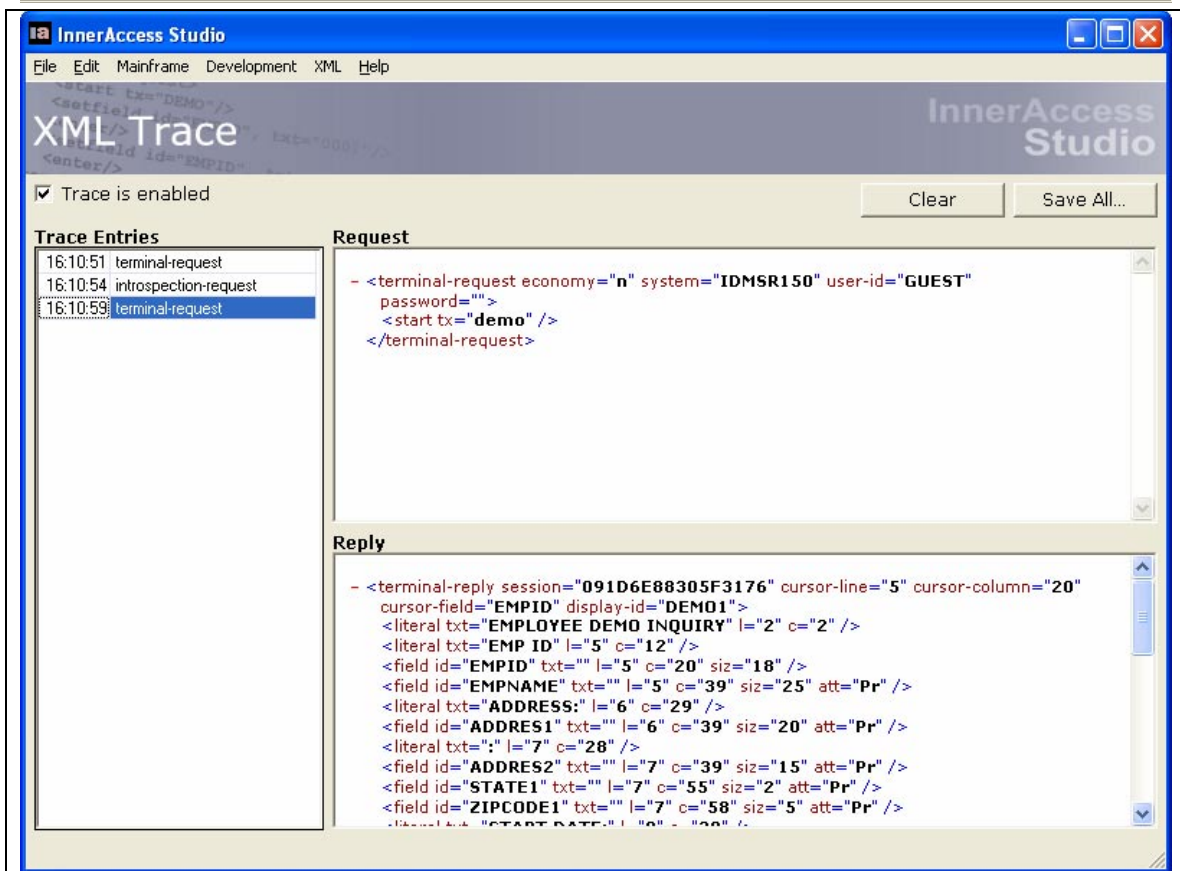
The developer's workbench is the key component to rapid development and deployment of web enabled and web services applications. The studio has several key components:

- Map Browser
- Wizard
- Scribe
- Web Terminal
- XML Console



### Map Browser

The Map Browser provides a graphical view to all maps.



## Wizard

The z/Services Wizard uses Introspection services to automatically discover all transactions, maps, and fields from a specific production region. The developer may choose to publish a single transaction, multiple selected transactions, or each and every transaction discovered by Introspection services.

The Wizard will merge a visual template with the XML returned from Introspection Services. The XML returns the complete specifications for each mapset, map, field, field description and transaction requested by the developer. The default templates are easily modified to provide a visual interface that access enterprise systems.

The Wizard performs a 1 web page for one transaction returned from Introspection Services. The simplicity of the Terminal API enables non-expert ASP and JSP developers to easily merge several sequential enterprise transactions into a single interface for the web-user.

Template design is easily implemented using either ASP or JSP technologies. Templates provide consistent, convenient, and user friendly interfaces to complex enterprise systems.

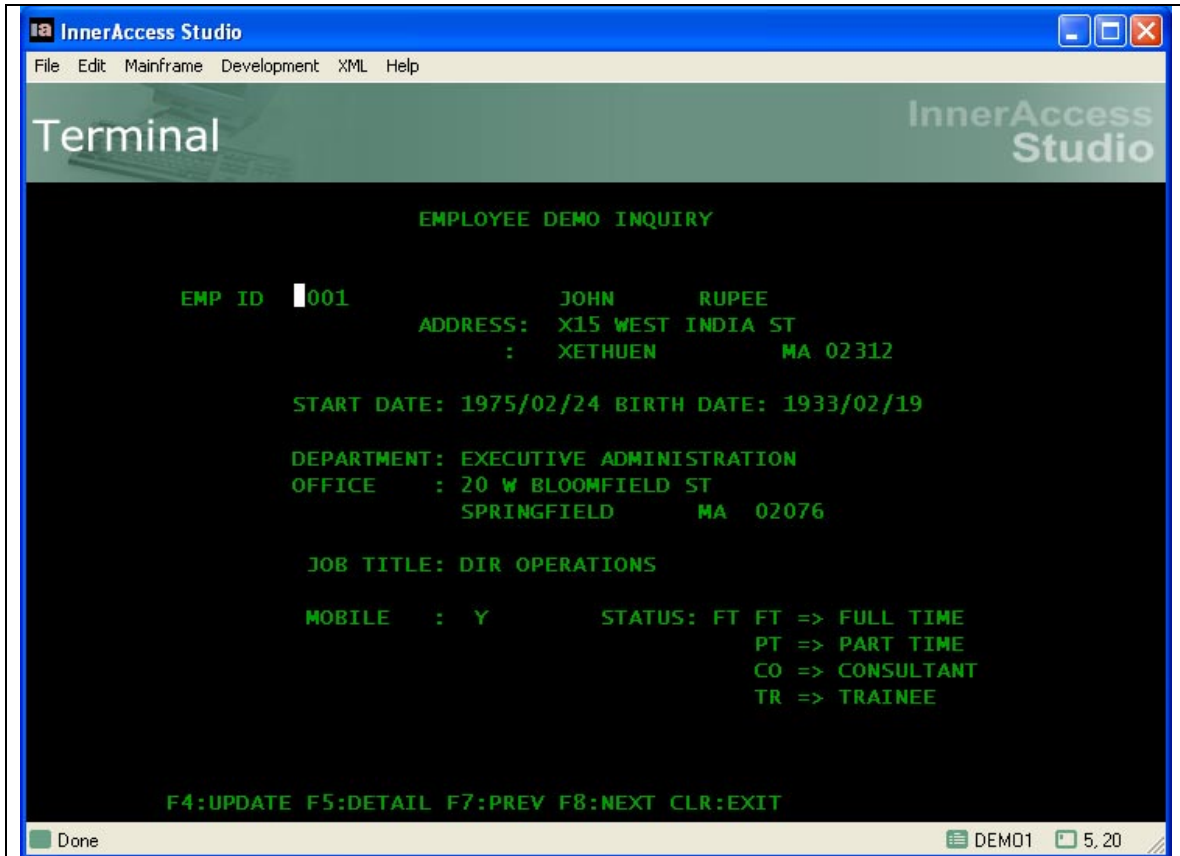
```
<%@ page language="java" isThreadSafe="true" session="true" import="inneraccess.*" errorPage="error.jsp"%>
<jsp:useBean id="iac" class="inneraccess.InnerAccessClient" scope="session"/>
```





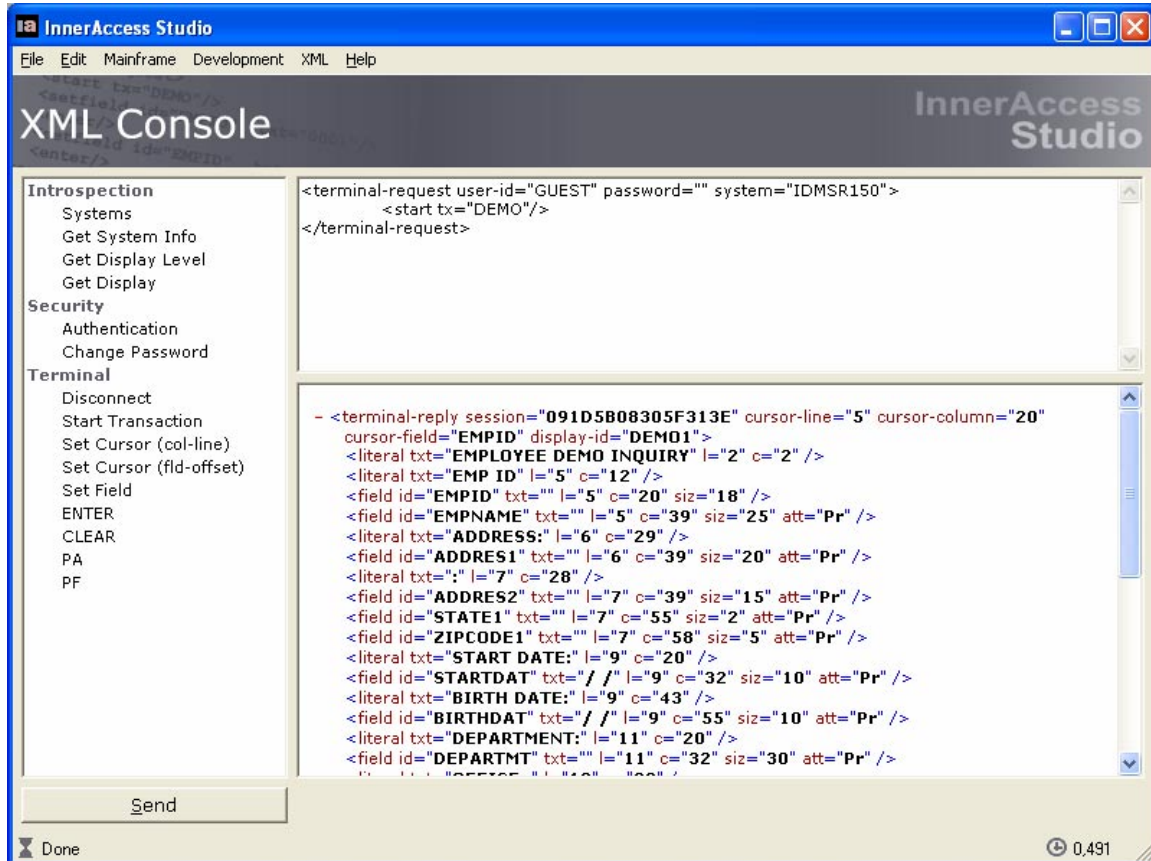
## Web Terminal

The Web Terminal uses Introspection Services to display screen based transactions interactively without having to use terminal emulation nor write a single line of code. The Web Terminal through the Terminal API provides the foundation for process simplification and automation within the z/Services Scribe tool.



## XML Console

The inner communication between each component of z/Services is performed with XML. Component calls to transaction execution services use the XML based SOAP protocol. The open architecture and use of standards within the z/Services framework allows for development extensions. To support extension development the XML communications between components can be traced and exposed with the z/Services XML Console.



z/Services XML console provides tracing of XML messages between components.

## Web Services

There are 3 critical components that are integral to the deployment of web services:

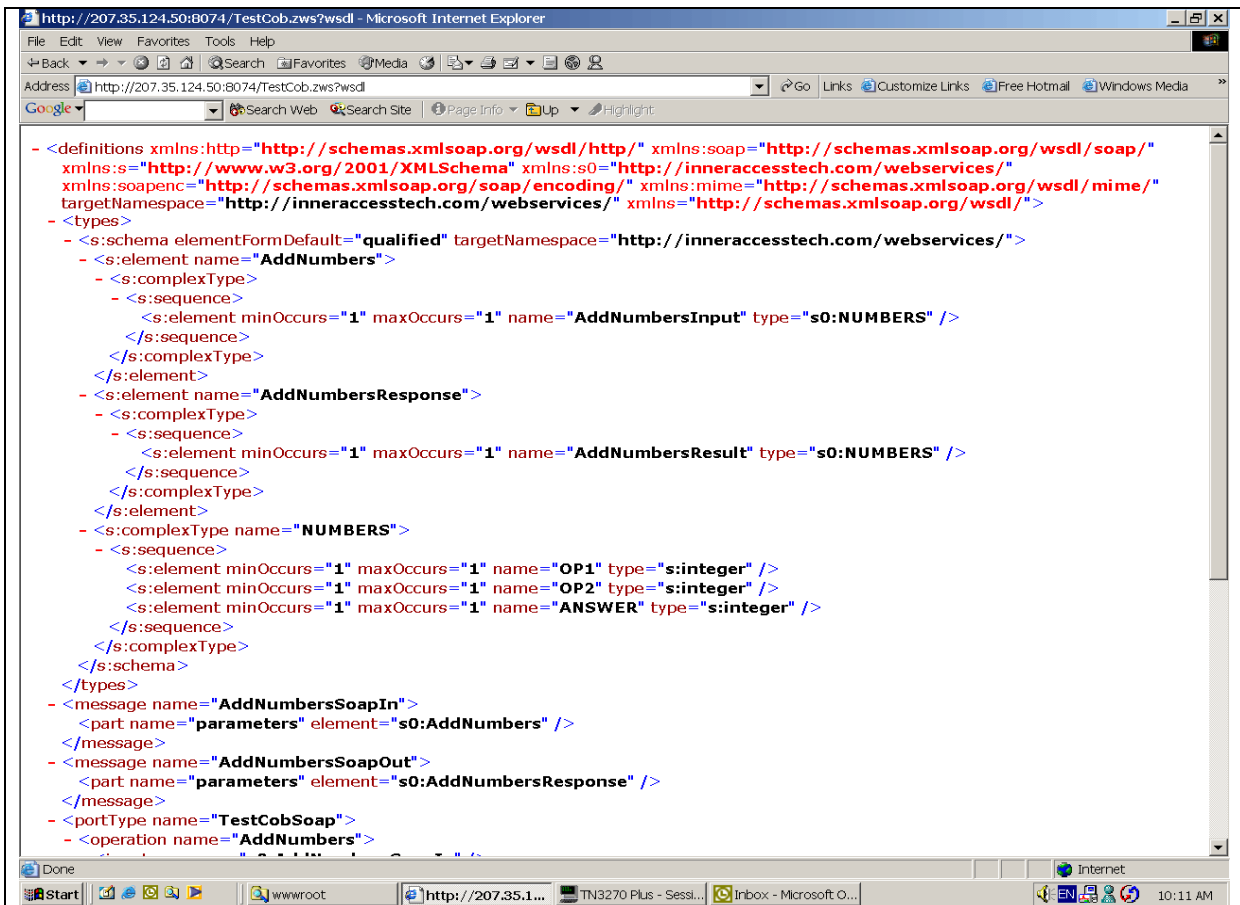
1. UDDI – Universal Description, Discovery, and Integration services and tools are publicly available from IBM and Microsoft. UDDI services enable requesting applications to discover the interface for your web service. Web Services are described with Web Services Description Language (WSDL)
2. WSDL – z/Services uses Introspection services to examine the COMM AREA for the requested transaction and automatically generate the WSDL definitions that are required for UDDI registration. Applications that request web services can

import the WSDL from the UDDI server to generate proxies (or stubs) to request your web service using the Simple Object Access Protocol (SOAP)

3. SOAP services – z/Services' SOAP listener provides the following functions:
  - Accepts XML based SOAP requests
  - Parses the SOAP envelope and generates a COMM AREA for use by the requested transaction
  - Uses z/Services execution services to execute the transaction
  - Formats the returned results from the transaction into a SOAP response for the requesting transaction

z/Services has been extensively tested with both the IBM WSAD (Websphere Application Developer) and with Microsoft Visual Studio .Net. For testing purposes several transactions at InnerAccess Technologies Inc. have been published as web services on the Microsoft (<http://test.uddi.microsoft.com>) as well as the IBM (<https://uddi.ibm.com/testregistry/registry.html>) Test Registries.

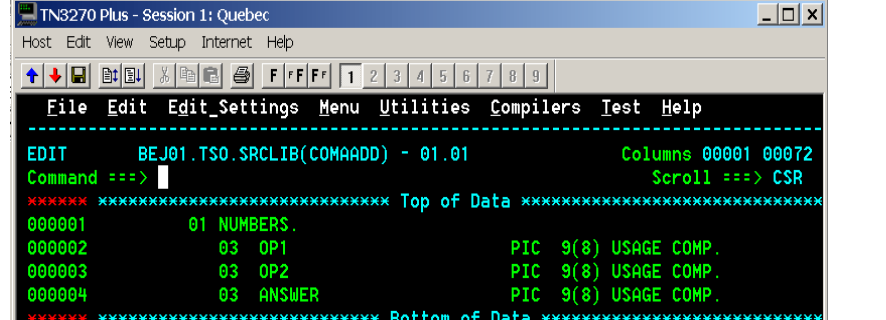
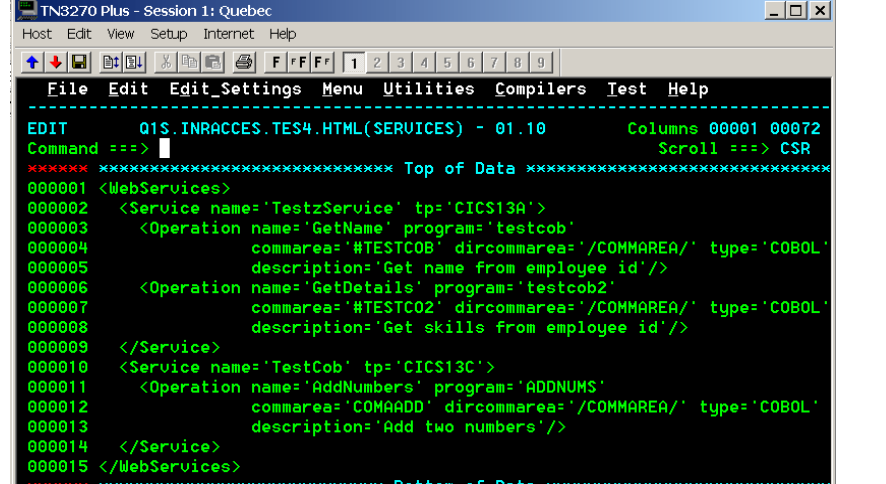

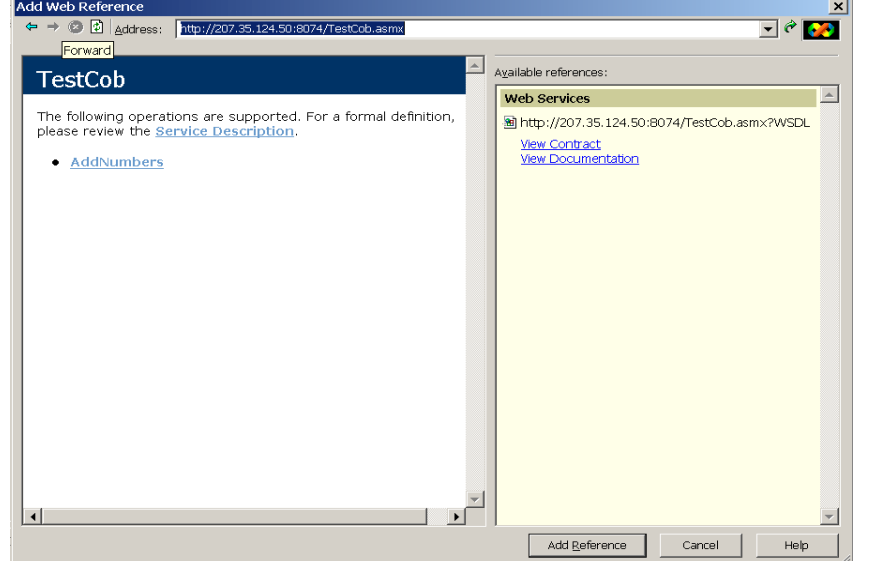
The automatically generated WSDL for our test application can be reviewed at:  
<http://207.35.124.50:8074/TestCob.zws?wsdl>



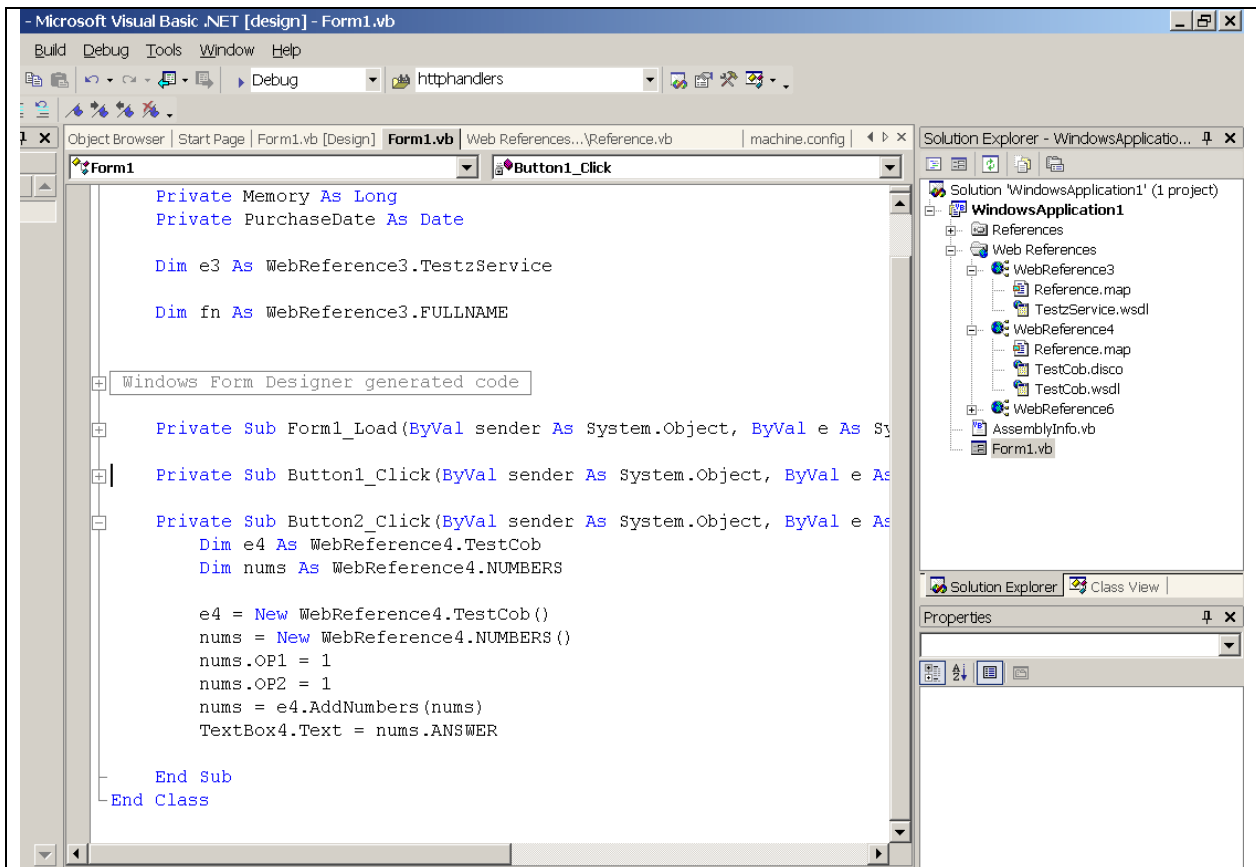
```
- <definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:s0="http://inneraccesstech.com/webservices/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
targetNamespace="http://inneraccesstech.com/webservices/" xmlns="http://schemas.xmlsoap.org/wsdl/">
- <types>
- <s:schema elementFormDefault="qualified" targetNamespace="http://inneraccesstech.com/webservices/">
- <s:element name="AddNumbers">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="AddNumbersInput" type="s0:NUMBERS" />
</s:sequence>
</s:complexType>
</s:element>
- <s:element name="AddNumbersResponse">
- <s:complexType>
- <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="AddNumbersResult" type="s0:NUMBERS" />
</s:sequence>
</s:complexType>
</s:element>
- <s:complexType name="NUMBERS">
- <s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="OP1" type="s:integer" />
<s:element minOccurs="1" maxOccurs="1" name="OP2" type="s:integer" />
<s:element minOccurs="1" maxOccurs="1" name="ANSWER" type="s:integer" />
</s:sequence>
</s:complexType>
</s:schema>
</types>
- <message name="AddNumbersSoapIn">
<part name="parameters" element="s0:AddNumbers" />
</message>
- <message name="AddNumbersSoapOut">
<part name="parameters" element="s0:AddNumbersResponse" />
</message>
- <portType name="TestCobSoap">
- <operation name="AddNumbers">
```

z/Services automatically generates WSDL specifications for your transactions published as Web Services.

The steps for generating a z/Services web service include:

<p>Linking the appropriate Comm Area to the correct transaction</p>	 <pre> EDIT      BEJ01.TSO.SRCLIB(COMAADD) - 01.01          Columns 00001 00072 Command ==&gt;   Scroll ==&gt; CSR ***** Top of Data ***** 000001      01 NUMBERS. 000002      03 OP1          PIC 9(8) USAGE COMP. 000003      03 OP2          PIC 9(8) USAGE COMP. 000004      03 ANSWER      PIC 9(8) USAGE COMP. ***** Bottom of Data *****         </pre>
<p>Generating the services definition file</p>	 <pre> EDIT      Q1S.INRACCES.TES4.HTML(SERVICES) - 01.10  Columns 00001 00072 Command ==&gt;   Scroll ==&gt; CSR ***** Top of Data ***** 000001 &lt;WebServices&gt; 000002   &lt;Service name='TestzService' tp='CICS13A'&gt; 000003     &lt;Operation name='GetName' program='testcob' 000004       commarea='#TESTCOB' dircommarea='/COMMAREA/' type='COBOL' 000005       description='Get name from employee id' /&gt; 000006     &lt;Operation name='GetDetails' program='testcob2' 000007       commarea='#TESTCO2' dircommarea='/COMMAREA/' type='COBOL' 000008       description='Get skills from employee id' /&gt; 000009   &lt;/Service&gt; 000010   &lt;Service name='TestCob' tp='CICS13C'&gt; 000011     &lt;Operation name='AddNumbers' program='ADDNUMS' 000012       commarea='COMAADD' dircommarea='/COMMAREA/' type='COBOL' 000013       description='Add two numbers' /&gt; 000014   &lt;/Service&gt; 000015 &lt;/WebServices&gt; ***** Bottom of Data *****         </pre>
<p>Generating the WSDL from the service definition file</p>	
<p>Publishing the web service</p>	

The last step of successful web services development is the creation of the application that consumes the service. Integration of z/Services with IBM's Websphere Application Developer or Microsoft's .Net developer platform is easy:



## **z/Services Recommendations**

A philosophy of elegance and simplicity is the key to z/Services. Straightforward API's with a singular focus on enterprise connectivity and extensions provide the platform for the web enablement and deployment of transactions as web services. The roadmap for successful deployment of web services includes a logical introspection of your infrastructure, skills, expertise, and of your trading partners (internal and external).

### ***Application Selection***

Initial applications for web services require strategic planning between the organization and IT. Gartner reports that only 30% of business processes are encapsulated within existing ERP systems. The goal of complete business-to-business systems integration is not likely to be realized in the near future. However, selected core business processes integrated into your value chain will provide immediate ROI and competitive advantages to the early adopters of z/Services.

Historically only 55% of CRM projects have met their objectives. The 2 key obstacles to success in CRM projects are uncontrolled scope and inadequate adoption of the people who have to use the CRM systems. The reasons for both obstacles are beyond the scope of this whitepaper. The lessons learned can be applied to web services:

- Keep the scope limited to very specific areas
- Understand the impact to the business processes
- Communicate clearly to the people involved in the process being changed

Selected initial web services objectives should include the exercising of new web services skills and transforming them into expertise. Quick ROI can be realized when there is a larger element of control that can be exercised over the user community. B2E initiatives deployed throughout an organization's intranet provide the opportunity to develop the expertise that marries existing enterprise skills with integration and internet capabilities.

Logical continuations include CRM Self Service applications that enable data and transaction access by selected trading partners in your value chain. Trading your experiences and expertise for business systems integration can provide exceptional opportunities to both IT and the organization as a whole.

The critical success factors to web services will include the ability to rapidly prototype and develop enterprise quality solutions. The z/Services suite provides a unique opportunity to re-use existing business rules that maintain the integrity of the information with your organization and deploy them with the realistic expectations of:

- Increased efficiencies realized in business processes
- Increased development productivity
- Reduction in costs
- Competitive Advantages