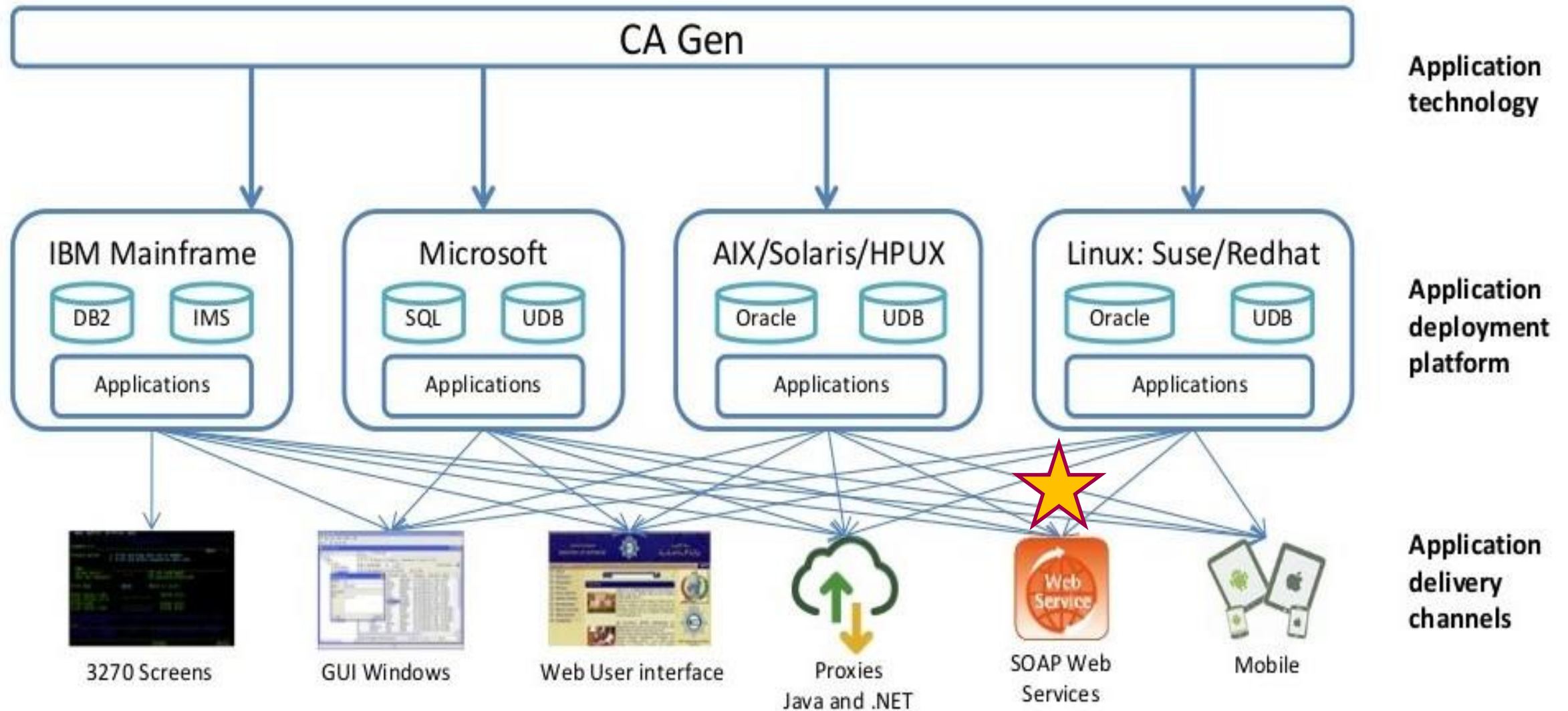


CA Gen Web Services

Business on Network



Gen Delivery Capability



What is Web Service

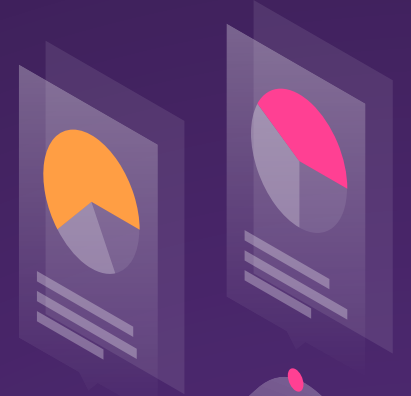
A collection of open protocols and standards which are widely used for exchanging data between systems or applications.

— Web Service Business Benefits

- Exposing Business Functionality on the network
- Sharing and Exchange Data between applications
- A Standardized Protocol which everybody understands
- Reduction in cost of communication



Type of Web Service & Protocol



SOAP Web Services.

Stands for Simple Object Access Protocol. (SOAP)

An intermediate language so that applications built on various programming languages could talk quickly to each other and avoid the extreme development effort.

RESTful Web Services.

Stands for REpresentational State Transfer. (REST)

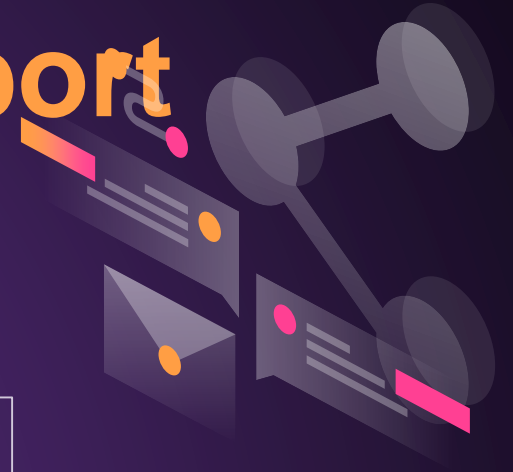
Used to build Web services that are lightweight, maintainable, and scalable.

WSDL.

Stands for Web Services Description Language (WSDL).

Is an XML-based file which tells the client application what the web service does and gives all the information required to connect to the web service.

Web Services Gen Support



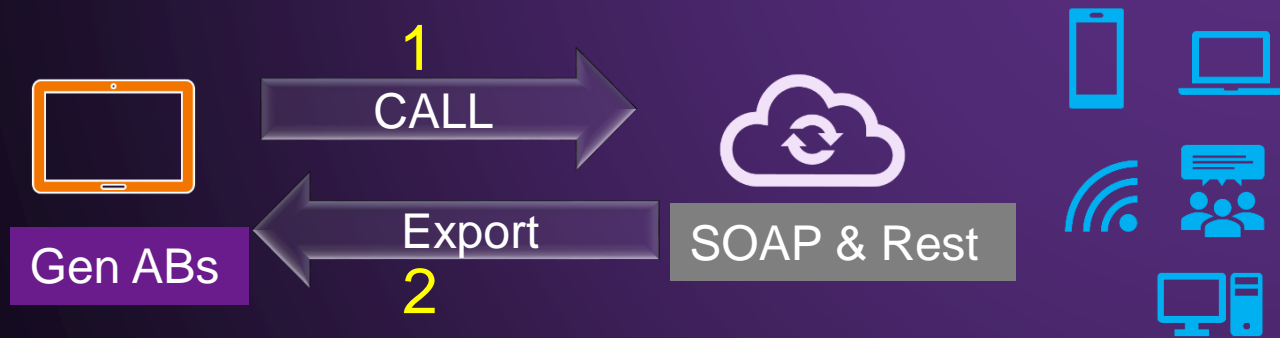
Gen Support

Consume (Call)

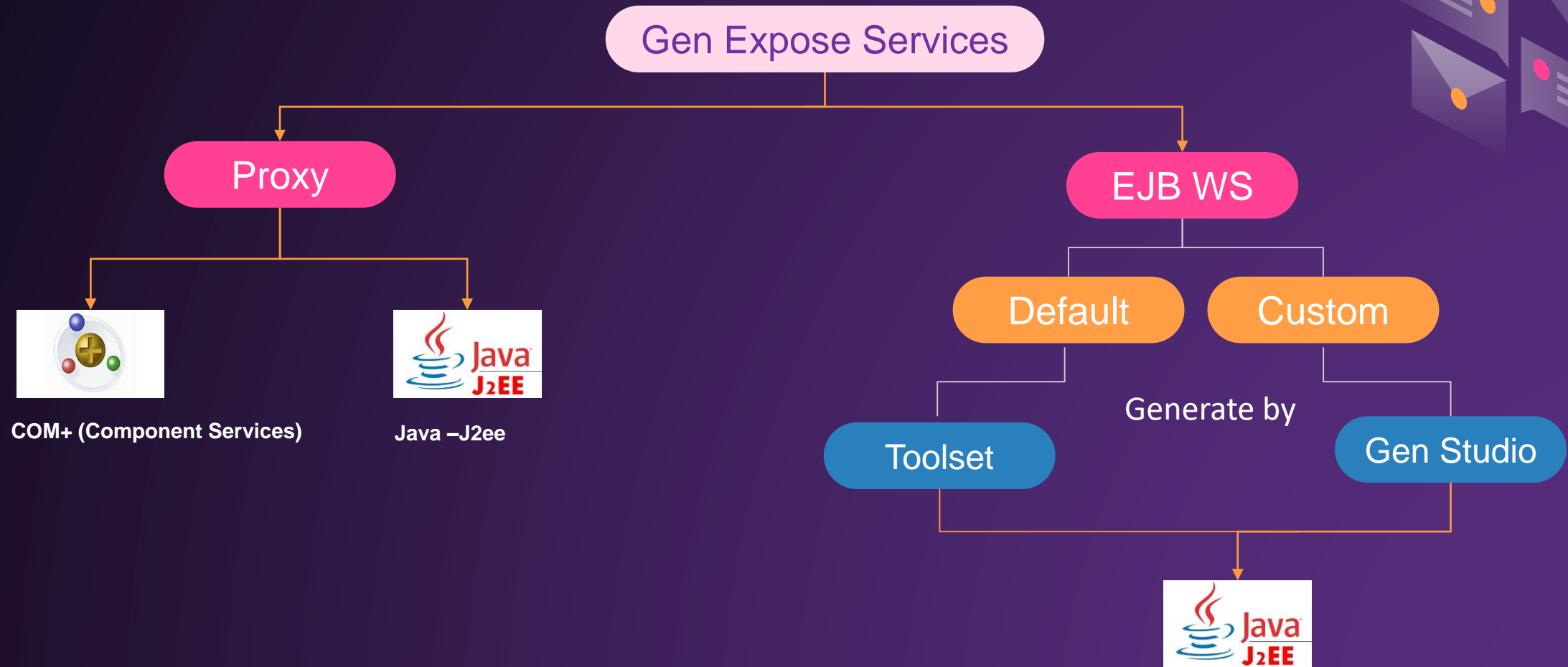
Program allow Gen to call other applications.

Expose (Export)

Program from Gen to allow other application to call Gen



Gen Expose Web Services Options



Gen WS

Proxy

- Implement to use since 2005
- Generate DLLs files via Gen Proxy server
- Export DLLs files to .NET or any middleware to create WSDL
- .NET team create reference for DLLs to send & receive data
From Gen proxy server
- Using middleware (CFB) to configure the connection
Between DLLs & Gen server
- .NET team use WCF to generate SOAP WSDL service (endpoint URL)

The screenshot shows the 'Proxy Code Generation' dialog box. At the top, 'Component Type' is set to 'Server Load Module'. Below this is a list of components with a tree view: 'Component - GS02WS' (selected), 'Interface - IGS02WS', and 'Method - GS02_SAUDI_POST_TRACKING'. To the right of the list are buttons: 'Expand', 'Contract', 'Select All', 'Deselect All', and 'Properties...'. Below the list are three dropdown menus: 'Operating System' (JVM), 'Target Proxy Type' (Java), and 'Type of Installation' (Local). There is a section for 'Additional Programming Interfaces' with checkboxes for 'Asynchronous' and 'XML'. Below that are three checked checkboxes: 'Automatically generate missing UUIDs', 'Use Classic Style API for Java Proxy', and 'Install after generation'. At the bottom right are four buttons: 'Generate', 'Save', 'Cancel', and 'Help'.

Common Format Buffer (CFB)

Is an encoded byte stream that CA Gen uses to exchange view data during the processing of a cooperative flow.

In addition to the import and export view, a CFB contains various pieces of control data that is used to process the cooperative flow

New Gen WS EJB WS

- Expose SOAP WSDL service via Gen “EJB Web Services” TP Monitor Server Direct
- .NET team consume WSDL service direct to adopting new WSDL for NIC customers



Default

Custom

Generation Defaults

Target Environment:

Operating System:

JVM

DBMS(TD):

JDBC

Language:

JAVA

TP Monitor:

EJB Web Services

Communications:

JavaRMI

Type of installation:

Local

DBMS Drive for local install:

C



CA Gen Studio

File Edit Construction Window Help

*Interfaces *Web Services

*gs:[GENERAL SERVICES INT] (opened)

- Web Service Definitions
 - saudipostDev
 - saudipostSoapPort
 - GS02_SAUDI_POST_TRACKING_[GS02_SAUDI_POST_TRACKING]

PSStep Interface Designer Web View Generation

Parameter	Gen Name	Gen Attribute	XSD Data T...	Mapping S...	Mapping
clientPassword	clientPassword	Text(256)	string	<None>	
clientId	clientId	Text(256)	string	<Variable>	
command	command	Text(80)	string	<None>	
dialect	dialect	Text(8)	string	<Literal>	Arabic
exitState	exitState	Numeric(10,0)	int	<None>	
nextLocation	nextLocation	Text(256)	string	<Variable>	
InSpTrackingShipment	InSpTrackingShipm...				
TrackingNumber	TrackingNumber	Text(35)	string	<Variable>	
InSpTrackingShipmentDet...	InSpTrackingShipm...				
LkEventCode	LkEventCode	Text(8)	string	<Variable>	
EventDate	EventDate	Date	date	<Variable>	
EventTime	EventTime	Time	time	<Literal>	
EventCreatorArabic	EventCreatorArabic	Text(70)	string	<Variable>	
EventCreatorEnglish	EventCreatorEnglish	Text(50)	string	<Variable>	
EventReceiverArabic	EventReceiverArabic	Text(70)	string	<Variable>	
EventReceiverEnglish	EventReceiverEnglish	Text(50)	string	<Variable>	
Remarks	Remarks	Text(200)	string	<Variable>	
InActionTypeSamisComm...	InActionTypeSamis...				
Flag	Flag	Text(1)	string	<Variable>	
InIssec1SecurityInfo	InIssec1SecurityInfo				
ID	CitizenId	Numeric(10,0)	int	<Variable>	
Locaction	LocationId	Numeric(4,0)	int	<Variable>	

Advanced Technical Topic

Comparison



EJB-WS vs Proxy

Gen EJB Web Services	Gen Proxy
Easy to generate & expose direct from Gen server	Normal Work but needs some experiences
No extra files out side the Gen Model	Each server have extra 2 DLLs files out Gen
No More middleware “CFB” & Proxy generation	CFB Mandatory
No extra Gen licenses to purchase	(CFB & Proxy) licenses mandatory
Reducing I/O resources due to stop CFB	Using I/O resources for CFB
Expose standard WSDL	Closed Solution for CA GEN
Support new .NET technology (.core)	Not support .NET technology (.core)
Support any API management software solution (IBM, Google, CA)	Needs work around to support API management
Support diagram trace utility (Debug)	Not Support direct
No extra middleware to connect NIC central applications by Client like (Angular, react, Vue.js, Blazor, Android and IOS)	Needs work around

New Gen Web Services

EJB WS

Pros



- Allow CA Gen servers to be easily consumed (call) by any external applications None Gen
- Generate J2EE for procedure steps as WS-I conformant Web Services Standard
- Allow to generate two types of WSDL (Default & Custom)
- Quickly deploy Web Services from any existing application function
- No work around solution here as its feature built in CA:Gen
- No more knowledge transfer need to Gen Developers
- No extra development Code to generate or expose WSDL
- No extra files out side Gen to generate or implement
- No injection files add to ear after assembly step
- Support diagram trace utility (Debug)
- Standard WS-Security built-in Application Server with out extra code (drag- drop)
- Support multiple application servers (WebSphere & WebLogic & JBOSS & Wildfly)

Cons



- Once one hour enchantment in Application Server (WebSphere) to support Soap WSDL
- Performance under evaluation
- Currently Support Expose SOAP WSDL will add more in the futures
- Generate Java J2EE only will add more in the futures

How Gen web services Display Inside WebSphere #1

Enterprise Applications

[Enterprise Applications](#) > CWServicesEAR

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

General Properties

Name
CWServicesEAR

Application reference validation
Issue warnings

Detail Properties

- [Target specific application status](#)
- [Startup behavior](#)
- [Application binaries](#)
- [Class loading and update detection](#)
- [Request dispatcher properties](#)
- [View Deployment Descriptor](#)
- [Last participant support extension](#)

References

- [Shared library references](#)
- [Shared library relationships](#)

[Back](#)

Modules

- [Metadata for modules](#)
- [Manage Modules](#)

Web Module Properties

- [Session management](#)
- [Context Root For Web Modules](#)
- [JSP and JSF options](#)
- [Virtual hosts](#)

Enterprise Java Bean Properties

- [Default messaging provider references](#)

Web Services Properties

- [Service providers](#)
- [Service provider policy sets and bindings](#)
- [Reliable messaging state](#)
- [Provide JMS and EJB endpoint URL information](#)
- [Publish WSDL files](#)
- [Provide HTTP endpoint URL information](#)

Database Profiles

- [SQLJ profiles and pureQuery bind files](#)

Web Services Properties

- [Service providers](#)
- [Service provider policy sets and bindings](#)
- [Reliable messaging state](#)
- [Provide JMS and EJB endpoint URL information](#)
- [Publish WSDL files](#)
- [Provide HTTP endpoint URL information](#)

How Gen web services Display Inside WebSphere #2

All JAX-WS service providers and other service providers are displayed. JAX-RPC services are not displayed.

+ Preferences

Start Listener Stop Listener



Select	Name ↕	Type ↕	Module ↕	Status ↻
You can administer the following resources:				
<input type="checkbox"/>	AS00 WS ME TRAFFIC ACCIDENT MENU	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS01 WS CR ACCIDENT DETAIL	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS02 WS CR ACCIDENT VEHICLE	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS03 WS CR ACCIDENT PERSON	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS04 WS CR ACCIDENT DESCRIPTION	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS05 WS MN ACCIDENT DAMAGE	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS06 WS ACCIDENT DAMAGE VEH DET	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS07 WS PUBLIC PRIVATE DAMAGE	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS08 WS QM ACCIDENT QUERY MENU	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS09 WS LI ACCIDENT MATCHES	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS10 WS LI BY ACCIDENT LOCATION	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS11 WS TRAFFIC ACCIDENT DETAILS	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS12 WS DI INVOLVED VEHICLE DET	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS13 WS DI INVOLVED PERSON DET	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS14 WS DI ACCIDENT DESCRIPTION	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS15 WS PR ENTER IDS	JAX-WS	AS00SM.jar	
<input type="checkbox"/>	AS16 WS LI ACTIVITY LIST	JAX-WS	AS00SM.jar	

Gen web services Security

Depend on Active directory policy

You can secure web services by configuring the message-level WS-Security policy set and bindings for a stand-alone security token that is either a Lightweight Third Party Authentication (LTPA) token **OR** a Username token.

Policy Set Attachments

Attach a policy set to the service, endpoints, or operations. Access the Policy Sharing link to allow clients to acquire the provider policy. Complete the attachment by providing system-specific configuration when you assign the appropriate binding.

+ Preferences

Attach ▾ Detach Policy Set Assign Binding ▾



Select	Service/Endpoint/Operation ▾	Attached Policy Set ▾	Binding ▾	Policy Sharing ▾
You can administer the following resources:				
<input type="checkbox"/>	AF40_WS_NPV_VIOLATOR_FINE_DTLS	NIC Applications Policy Set	NIC-JAX WS-Policy-Binding	Disabled
<input type="checkbox"/>	AF40_WS_NPV_VIOLATOR_FINE_DTLSPort	NIC Applications Policy Set	NIC-JAX WS-Policy-Binding	Disabled (inherited)
<input type="checkbox"/>	Af40WsNpvViolatorFineDtIsCall	NIC Applications Policy Set	NIC-JAX WS-Policy-Binding	Disabled (inherited)

Total 3

SoapUI WS Display

The image shows the SoapUI WS Display interface. The top bar indicates 'Request 1' and the URL 'https://172.29.2.103:9443/CB01SM/CB01_EXPOSE_WEB_SERVICES'. The left pane shows the raw XML of the request, and the right pane shows the raw XML of the response. The status bar at the bottom indicates 'response time: 17ms (453 bytes)'.

Request XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:cb0="http://tempuri.org/CB01ExposeWebServices/">
  <soapenv:Header/>
  <soapenv:Body>
    <cb0:Cb01ExposeWebServicescall>
      <!--Optional:-->
      <Cb01ExposeWebServicesImport command="?" clientId="?" clientPassword="?">
        <!--Optional:-->
        <InCobonWs>
          <Sell>3</Sell>
        </InCobonWs>
      </Cb01ExposeWebServicesImport>
    </cb0:Cb01ExposeWebServicescall>
  </soapenv:Body>
</soapenv:Envelope>
```

Response XML:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:a="http://tempuri.org/CB01ExposeWebServices/">
  <soapenv:Body>
    <a:Cb01ExposeWebServicescallResponse xmlns:a="http://tempuri.org/CB01ExposeWebServices/">
      <Cb01ExposeWebServicesExport exitStateMsg="" exitStateType="">
        <CobonWs>
          <TotalCobon>8</TotalCobon>
        </CobonWs>
      </Cb01ExposeWebServicesExport>
    </a:Cb01ExposeWebServicescallResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Auth Headers (0) Attachments (...) WS-A WS-RM JMS Headers JMS Property (0)

response time: 17ms (453 bytes)

Headers (6) Attachments (...) SSL Info (1 certs) WSS (0) JMS (0)

1:1

CB01_EXPOSE_WEB_SERVICES

IMPORTS: ...

EXPORTS: ...

LOCALS:

ENTITY ACTIONS:

SET cobon_ws total_cobon TO in cobon_ws sell + 5

Example #1

Generate new EJB web services from Gen server
URL

https://172.29.2.103:9443/CB01SM/CB01_EXPOSE_WEB_SERVICES?wsdl

Endpoint

https://172.29.2.103:9443/CB01SM/CB01_EXPOSE_WEB_SERVICES/CB01EXPOSEWEBSERVICES.wsdl

Business Sys	↓	↓	↓	↓	COBON REGISTER SYSTEM
Server Mangr	-	-	↓	-	CB01SM
Proc Step	-	-			CB01_EXPOSE_WEB_SERVICES

WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
- <wsdl:definitions targetNamespace="http://tempuri.org/Gs02SaudiPostTracking/" name="GS02_SAUDI_POST_TRACKING"
  xmlns:ns1="http://schemas.xmlsoap.org/soap/http" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://tempuri.org/Gs02SaudiPostTracking/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <wsdl:types>
    - <xs:schema targetNamespace="http://tempuri.org/Gs02SaudiPostTracking/" xmlns:tns="http://tempuri.org/Gs02SaudiPostTracking/"
      elementFormDefault="unqualified" attributeFormDefault="unqualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:element name="Gs02SaudiPostTrackingcall" type="tns:Gs02SaudiPostTrackingcall"/>
        <xs:element name="Gs02SaudiPostTrackingcallResponse" type="tns:Gs02SaudiPostTrackingcallResponse"/>
        - <xs:complexType name="Gs02SaudiPostTrackingcall">
          - <xs:sequence>
            <xs:element name="Gs02SaudiPostTrackingImport" type="tns:Gs02SaudiPostTrackingImport" minOccurs="0"/>
          </xs:sequence>
        </xs:complexType>
        - <xs:complexType name="Gs02SaudiPostTrackingImport">
          - <xs:sequence>
            + <xs:element name="InSpTrackingShipment" minOccurs="0">
              - <xs:element name="InSpTrackingShipmentDetail" minOccurs="0">
                - <xs:complexType>
                  - <xs:sequence>
                    <xs:element name="LkEventCode" type="xs:string"/>
                    <xs:element name="EventDate" type="xs:string"/>
                    <xs:element name="EventTime" type="xs:string"/>
                    <xs:element name="EventCreatorArabic" type="xs:string"/>
                    <xs:element name="EventCreatorEnglish" type="xs:string"/>
                    <xs:element name="EventReceiverArabic" type="xs:string"/>
                    <xs:element name="EventReceiverEnglish" type="xs:string"/>
                    <xs:element name="Remarks" type="xs:string"/>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            - <xs:element name="InActionTypeSamisCommonFields" minOccurs="0">
              - <xs:complexType>
                - <xs:sequence>
                  <xs:element name="Flag" type="xs:string"/>
                </xs:sequence>
              </xs:complexType>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:schema>
    </wsdl:types>
  </wsdl:definitions>
```



Example #2

Microsoft Blazor call direct CA Gen Central Application through EJB Web Services

Blazor Demo

Home |

Test Services

Tracking Number : *

tracking number

Event Code : *

event code

Event Date : *

mm / dd / yyyy

Event Time : *

--Select--

Event Creator Arabic : *

event creator arabic

Event Creator English :

event creator english

Event Receiver Arabic : *

event receiver arabic

Event Receiver English :

event receiver english

Remarks :

remarks

Operator Id : *

operator id

Location Id : *

location id

Submit

Clear



Expose Gen web services

Demo

New Gen Call Web Services

CALL EXTERNAL

- Statement allows Gen to call external (None Gen) web services directly from an action diagram.
- The web services can be of the type HTTP or HTTPS.

How to

- Understand the Format of the Parameters
- Create an empty Action Diagram
- Add a Call External Statement in an Action Diagram
- Match Service Parameters to Gen Views
- Configure the Application for Secure Web Services
- Certified your web Services

```
CALL TO WEBSERVICE
IMPORTS:
  Entity View import mobile_user (optional,transient,import only)
    user_name (mandatory)
    mobile_no (mandatory)
    misc_details (mandatory)
EXPORTS:
  Work View  export ief_supplied (transient,export only)
    text
LOCALS:
ENTITY ACTIONS:

CALL EXTERNAL Web Service "Insertmobileuser"
```

Method Name Insertmobileuser

Matched input:

Parameter	View
string Insertmobileuser.Insertmobileuser.desc...	IMPORT MOBILE_USER.MISC_DETAILS
string Insertmobileuser.Insertmobileuser.Mobi...	IMPORT MOBILE_USER.MOBILE_NO
string Insertmobileuser.Insertmobileuser.una...	IMPORT MOBILE_USER.USER_NAME

Add Delete

Matched Output:

Parameter	View
string InsertmobileuserResponse.Insertmobile...	EXPORT IEF_SUPPLIED.TEXT

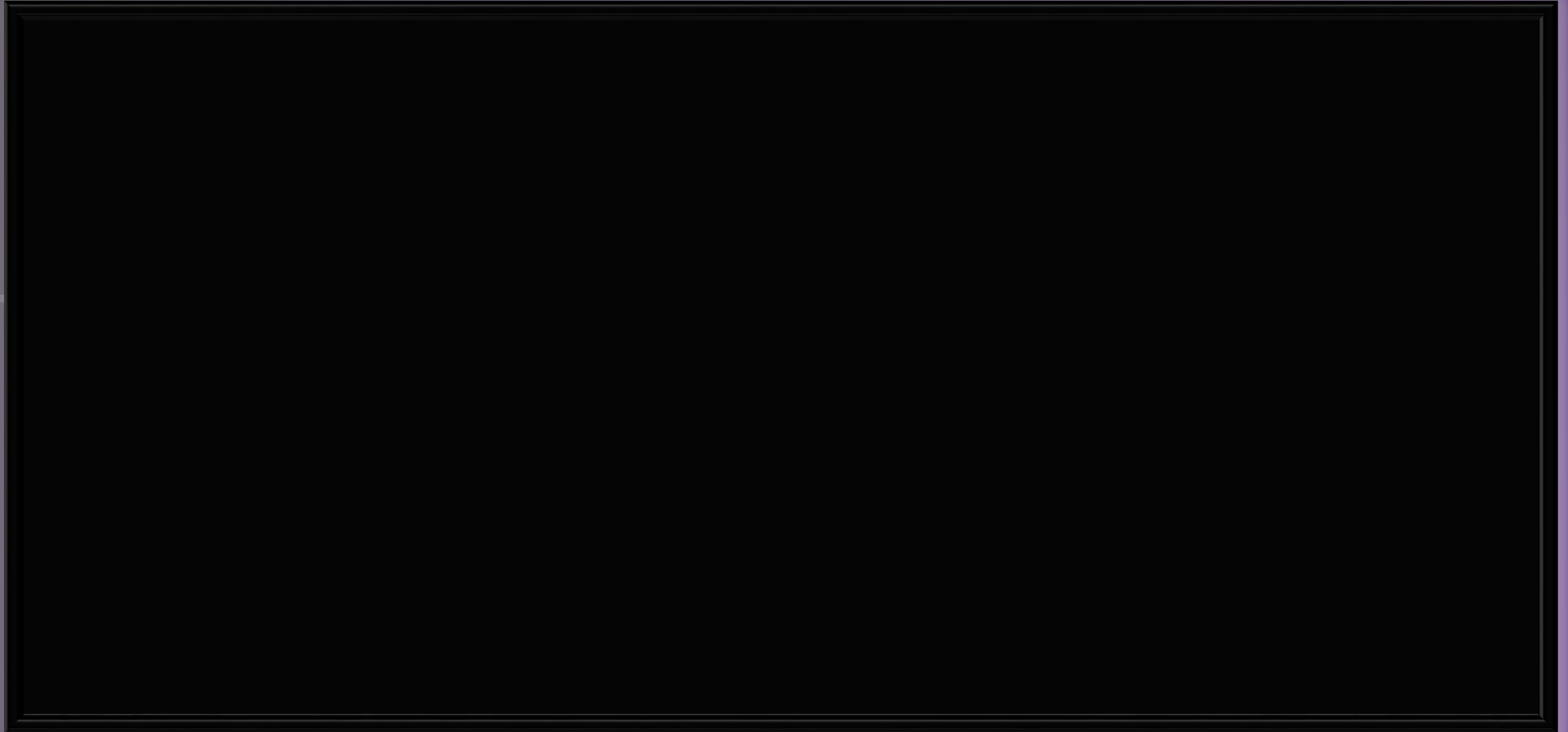
Add Delete

OK Cancel Help

New Gen CALL EXTERNAL

Demo

<https://youtu.be/wuBd2v6jfyQ>



CA Gen Web Services



Prepared By
Tarek Kamal



31 March 2021 /v04-WorkShop