

SOA in CA Gen

Dalia Soliman
Director, Development
CA



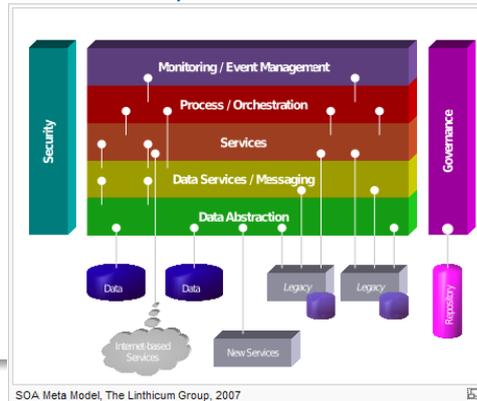
Agenda

- > What is SOA?
- > What are Web Services?
- > Web Services Features in CA Gen r8
- > Web Services in CA Gen Beyond r8
- > Questions & Answers



SOA (Service-Oriented Architecture)

OASIS defines SOA as: *A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.*



SOA Principles

> An architecture whose main principles are:

- Technology agnostic.
- Abstracted and modular software services.
- Loose Coupling of Services.
- Orchestration is used to associate individual SOA objects.
- Adherence to a service contract: uses Meta Data to describe the characteristics of the Services and the data to drive them.
- Interoperability between services.

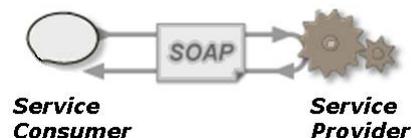
SOA and CBD

- > Component Based Development (CBD) sets good standards to architect your product.
- > Following CBD principals are a step in the right direction for SOA.



Web Services

- > Defined by the [W3C](#) as: *"a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards."*



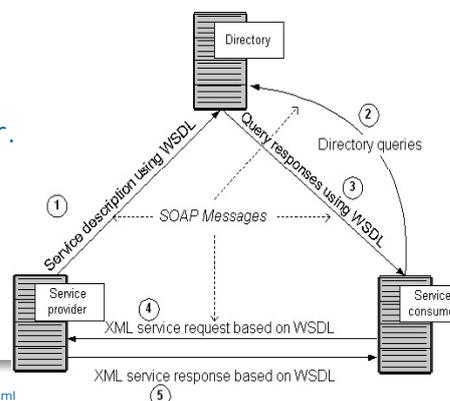
Web Services Essentials

- > **XML:** eXtensible Markup Language.
- > **WSDL:** Web Service Definition Language.
- > **SOAP:** Simple Object Access Protocol. XML over HTTP.
- > **Web Service Operation:** Web Method or module that performs a specific function or service.
- > **UDDI:** Universal Description, Discovery and Integration is a platform-independent, XML-based registry for businesses to list themselves. It is sponsored by OASIS.



UDDI Principles

- > A service provider describes a service using WSDL and publishes it to a directory.
- > A service consumer locates a service and determines how to communicate with it based on the WSDL.
- > The service consumer sends a request to the service provider.
- > The service provider provides the expected response to the service consumer.



http://service-architecture.com/web-services/articles/web_services_explained.html



.NET Web Services

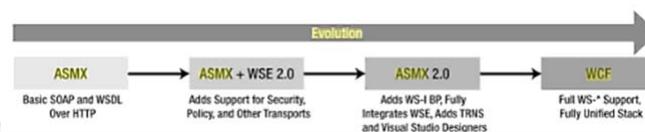
> Microsoft has 2 flavors of Web Services:

1. .ASMX Web Service:

- Original, simple, and widely adopted.
- Configuration is limited.

2. Windows Communication Foundation:

- Better configuration, performance, more Protocols, and not restricted to IIS.
- Less industry adoption probably due to complexity.



Java Web Services

> Java offers 2 ways of implementing Web Services:

- Java Class programming model:
the Web Service is deployed in a Web Container.
- Enterprise JavaBeans model:
the web service is implemented as a stateless session bean that gets deployed in an EJB container.

> Web Services are now easier in Java:

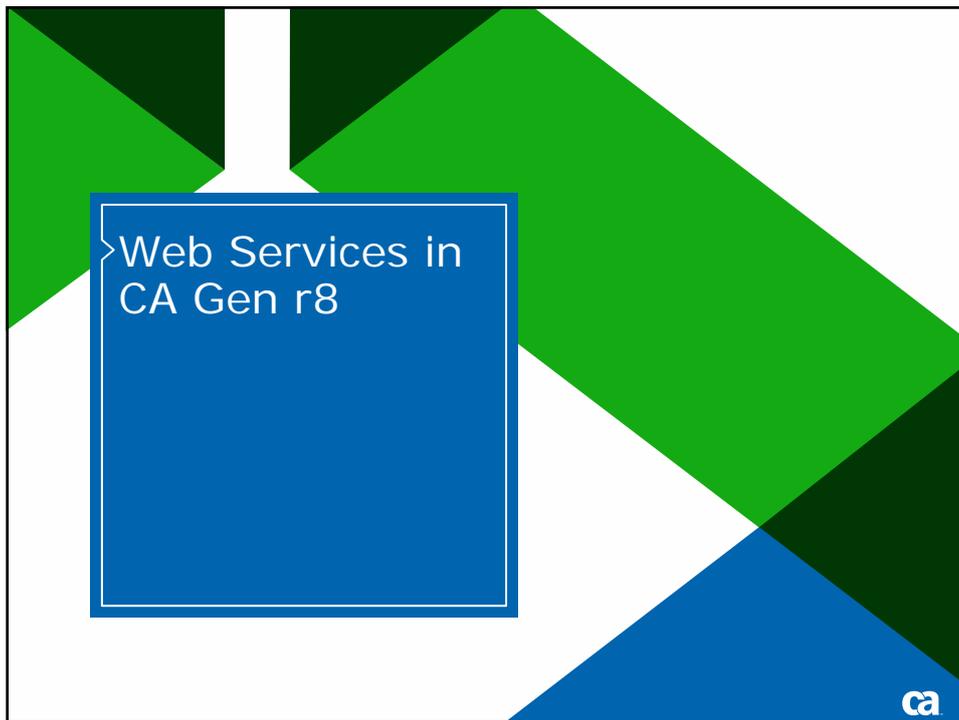
- The EJB 3.0 Specification added annotations to facilitate creating Web Services.
- Deployment tools generate a service endpoint interface, as well as a WSDL document, using JAX-WS rules for Java WSDL mapping.



Java Web Services - Technologies

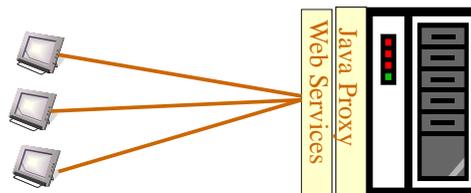
- > **JAX-WS:** Java API for XML-Based Web Services. An API used for both SOAP based and RESTful Java Web services. It is included in Java SE 6.
- > **JAXB:** Java Architecture for XML Binding. It provides a convenient way to process XML content using Java objects by binding its XML schema to Java representation.
- > **Metro:** is a framework for developing Web Services. It contains JAX-WS and WSIT. It is bundled with Glassfish and is used in many Application Servers e.g. WebLogic, JBoss.
- > **WSIT:** Web Services Interoperability Technology. APIs to create web service clients and services that interoperate between the Java platform and Microsoft's WCF and .NET.





Prior to CA Gen r8 - Web Service Wizard

- > Prior to CA Gen r8, Web Services were natively supported in CA Gen through the Web Service Wizard.
- > This plug-in fully automates the process of exposing new and existing CA Gen back-end servers as Web Services.
- > Uses the Java Proxies underneath the covers.



Web Services in CA Gen r8

CA Gen r8 adds the following features in r8:

- > .NET Proxy – Web Services
- > EJB Web Services
- > Customized Web Service Interfaces
- > Web Service Access



.NET Proxy –
Web Services

.NET Proxy Based Web Services

- > The Web Services capabilities are built into the CA Gen .NET Proxy.
- > You do not need any extra licenses or generation options.
- > This capability has been in the product for a few releases.
- > CA Gen r8 is the first release where it is officially certified and supported.



Basics of .NET Proxy Web Services

- > The .NET Proxy's Public methods are defined with the attribute WebMethod.
- > The WebMethod attribute exposes proxy methods as part of the XML Web service.
- > IIS publishes it as a Web Service.
- > CA Gen creates a .aspx test harness to test Web Service.
- > You can write your client code using either ASMX or WCF to call the CA Gen .NET Proxy Web Service.





EJB Web Services

- > CA Gen r8 introduces a new generation option: EJB Web Services.
- > This option is covered by the Enterprise Java Beans (EJB) license.

Generation Defaults

Target Environment:

Operating System: JVM

DBMS(TD): <NONE>

Language: JAVA

TP Monitor: EJB Web Services

Communications: JavaRMI

Type of installation: Local

DBMS Drive for local install: C

Override Bus Sys Target Environment with above defaults.
 Run Consistency Check for each item generated.
 Generate source code with trace (Gen All).
 Include Drop statements in DDL (Gen All).
 Qualify tables and indices with owner ID (DDL).
 Create Storage Group in DDL (DB2 only).
 Delete generated source after remote install.
 Create RI Alter Primary/Foreign Keys/Triggers in DDL.
 Process modules marked for Compatibility.

OK Save Cancel Help

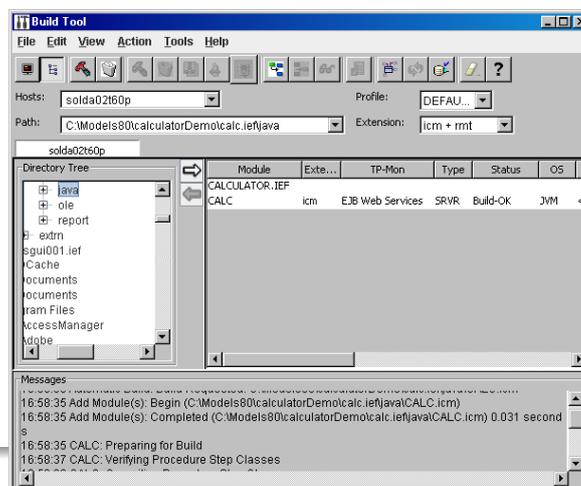
EJB Web Services - Generation

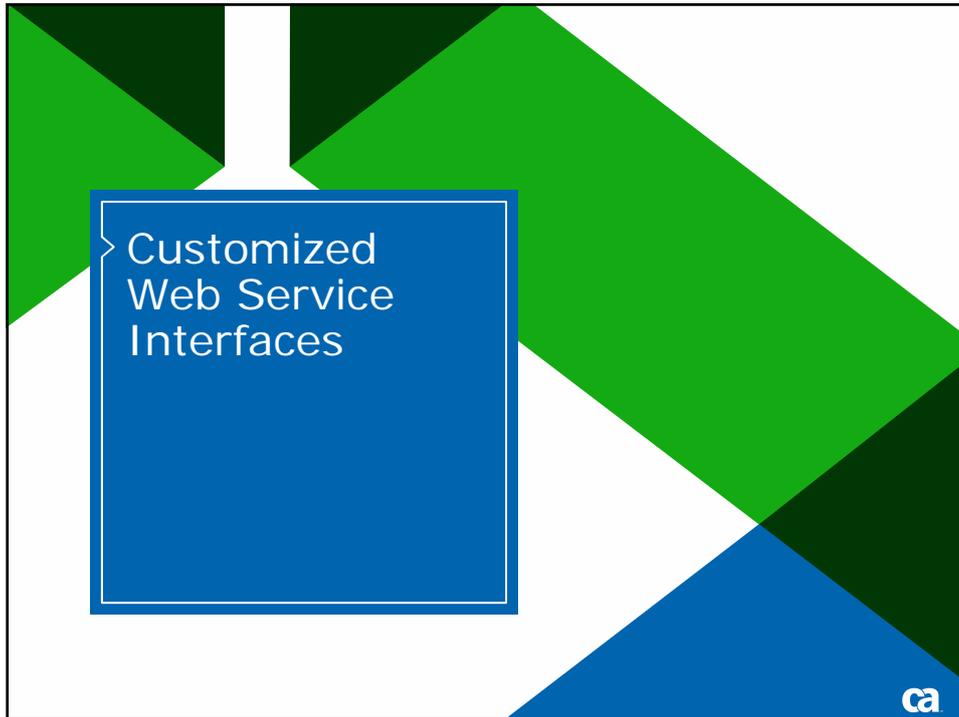
- > Selecting this option at generation time adds annotations to the EJBs according to the EJB 3.0 Specification.
- > Application Servers create Web Service interfaces based on the annotations using JAX-WS at deployment time.
- > Some Application Servers such as JBoss and WebLogic use the JAX-WS code from Metro.
- > Other Application Servers, such as WebSphere, use Axis.



EJB Web Services Build and Assemble

- > The Build Tool is used to build and assemble the generated artifacts.
- > Each Server Procedure Step results in an EBJ.
- > All the EJBs for a Server Load Module are packaged in a JAR file.





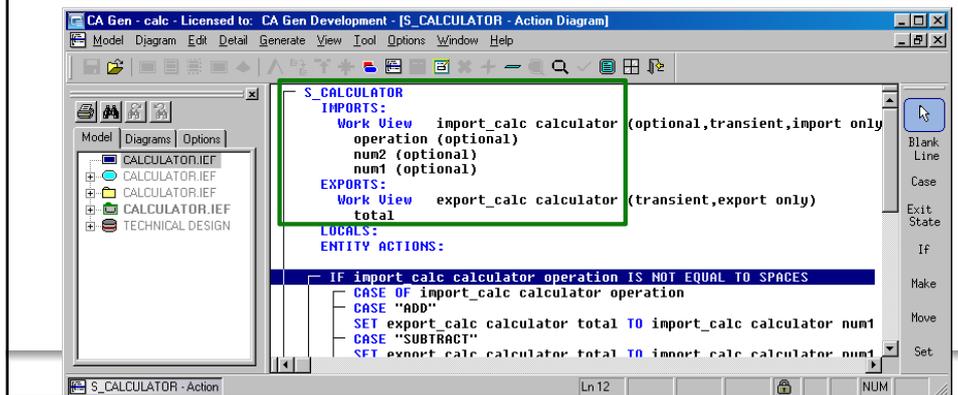
Customized Web Service Interfaces

- > The Web Service Definition and Operations generated using EJB Web Services use the Name properties associated with a Server PStep: Pstep and attribute names.
- > The PStep Interface Designer can be used to create customized versions of these properties of the Server PSteps.

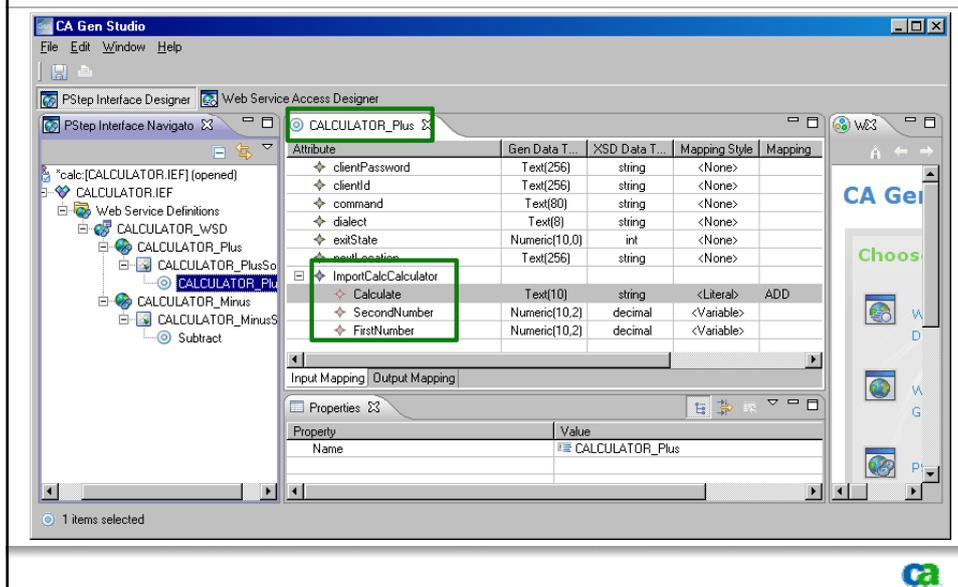


Original PStep in the Toolset

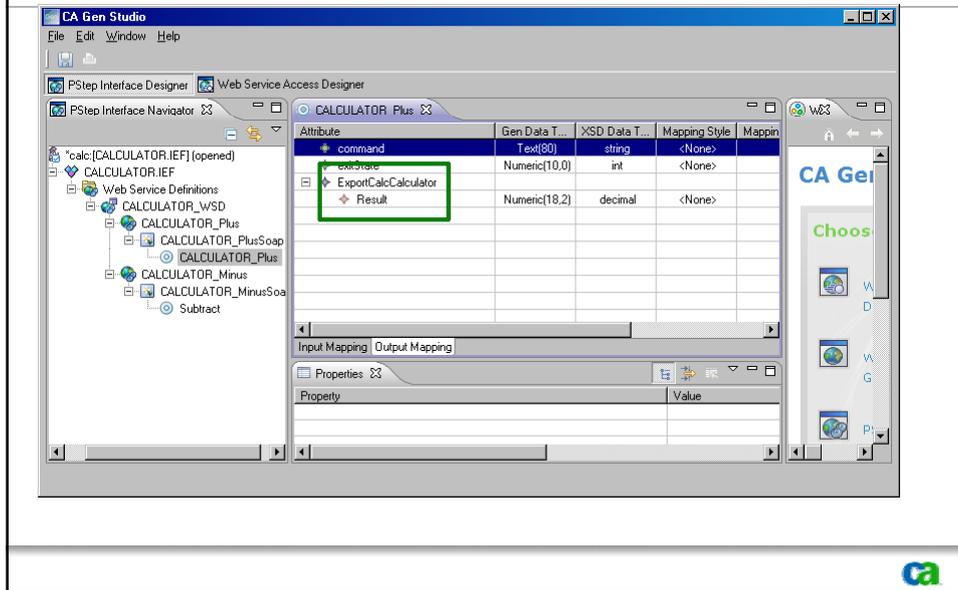
- > Notice the PStep and Attribute names in the Import and Export Views of S_CALCULATOR.
- > You can customize them in the PStep Interface Designer.



Customized Interface in Gen Studio-Imports

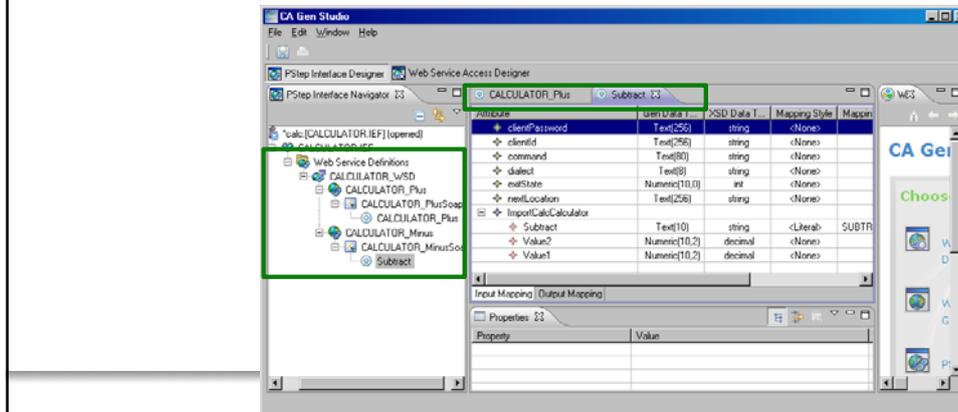


Customized Interface in Gen Studio-Exports



PStep Interface Designer – Design Component

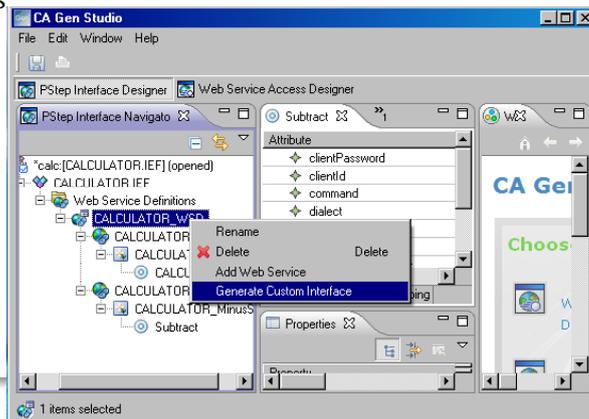
- > PSteps from different Load Modules/EJBs can be added to the same Web Service Definition.
- > A PStep can have one or more custom interfaces. They will be saved in the model.



PStep Interface Designer – Generation Component

> Main steps:

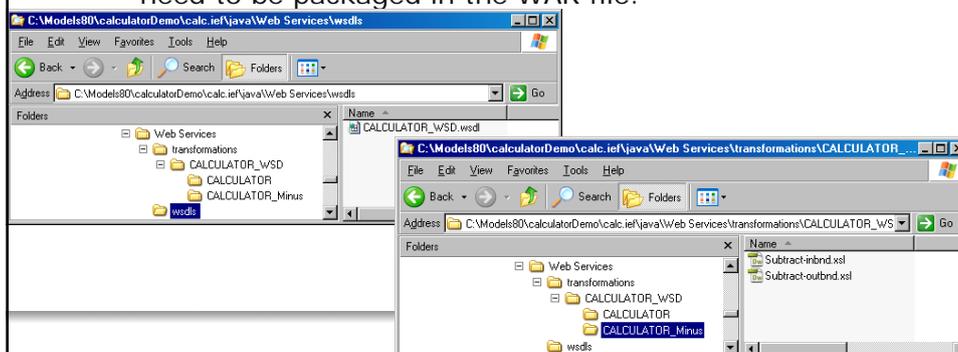
1. Create the Web Service Definition
2. Add the necessary Web Service Operations
3. Customize names
4. Set values
5. Generate Custom Interface



PStep Interface Designer – Generated Artifacts

> The PStep Interface Designer will generate:

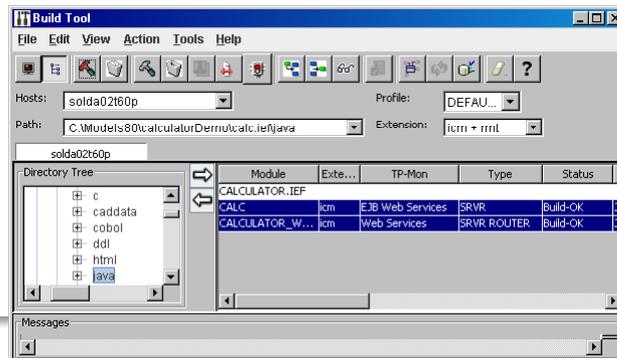
- **WSDL:** contains the definition of the custom interfaces.
- **XSL:** 2 transformation sheets to convert to and from a custom interface to the PStep interface.
- **ICM:** of type Server Router. It defines the artifacts that need to be packaged in the WAR file.



PStep Interface Designer – Build

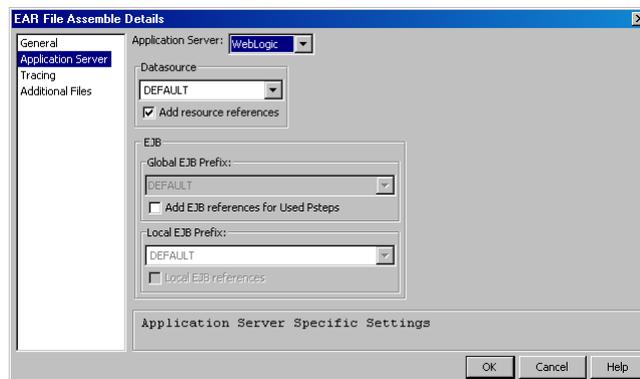
> The CA Gen Build Tool will be used to build the following:

- JAR: containing the EJB Web Services
- WAR: containing mainly the WSDL and XSL sheets.



PStep Interface Designer – Assemble

> You go through the normal Assemble process to create an EAR file.



PStep Interface Designer – Execution Component

- > A Web Service consumer calls a custom Operation.
- > The Gen Web Service Runtime Router transforms the incoming SOAP message and invokes the EJB Web Service.
- > The EJB Web Service executes the service logic and returns a SOAP message to the Gen Web Service Runtime Router.
- > The Router transforms the outgoing SOAP message and sends it back to the service consumer.



Web Services in
CA Gen Beyond
r8



SOA in CA Gen Beyond r8



- > CA Gen will continue to expand its SOA and Web Services capabilities.
- > The next steps planned are:
 - CA Gen CICS Web Services
 - Web Services as a Transport



CA Gen CICS Web Services

- > CA Gen r8 contains the infrastructure work to support CICS Web Services.
- > This would allow generated CICS Servers to be exposed as Web Services.
- > Since this communication would bypass the traditional CFB and COMMAREA, the 32K limit would be removed.
- > The limits would be dictated by some other much greater limit due to language choice or system limitations.
- > To participate, server managers would have to be regenerated and re-installed at a minimum.



Web Services as a Transport

- > CA Gen Servers will be exposed as Web Services (via .Net Proxies and EJBs in CA Gen r8, plus CICS beyond Gen r8).
- > CA Gen Clients will have the ability to consume CA Gen Web Services.
- > The 32K limit will be eliminated in this transport mode.



Summary



Summary

- > What is SOA?
- > What are Web Services?
- > SOA Features in CA Gen r8
- > SOA in CA Gen Beyond r8



Disclaimer

CA reserves the right to modify any plans regarding any future deliverables addressed in this presentation



Questions and Answers

