



Software Group

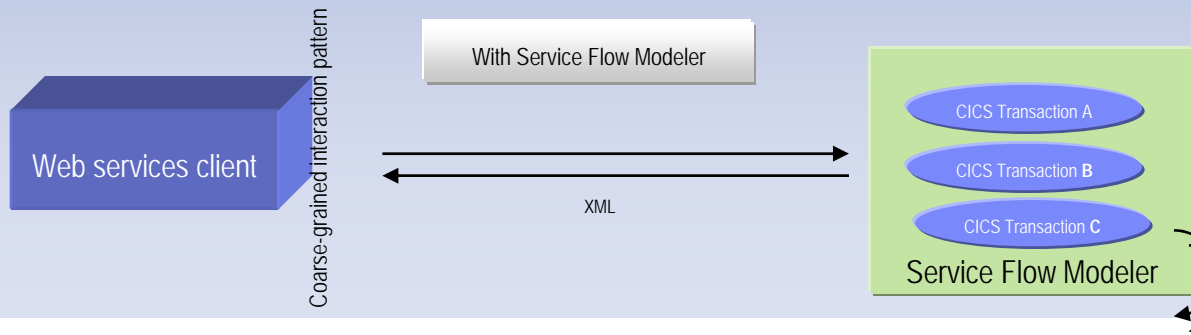
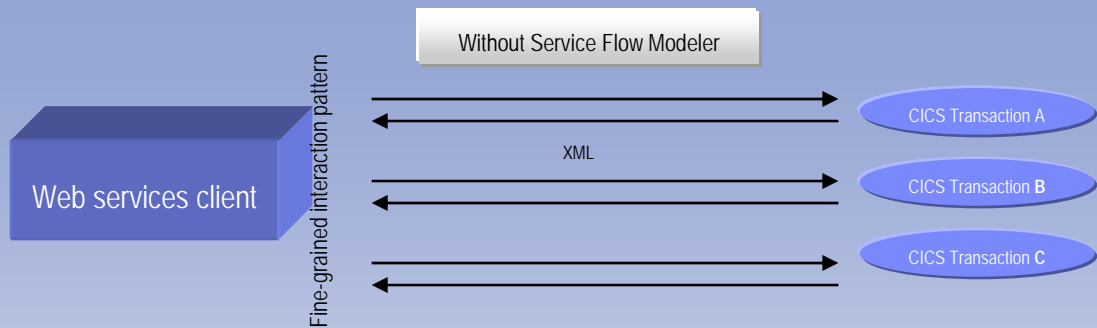
# *The CICS Service Flow Modeler*



# Why a Service Flow Feature?

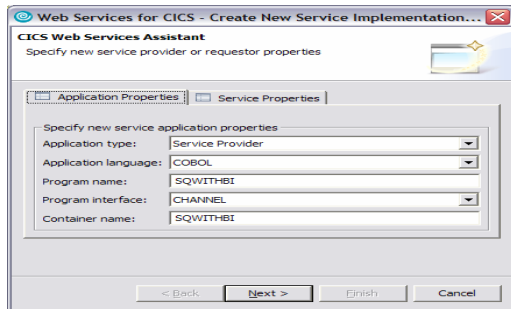
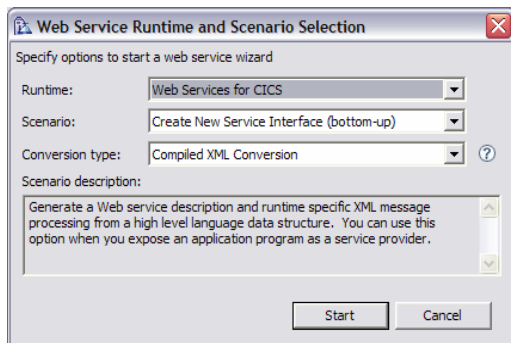
- **Many leading enterprises have chosen IBM Mainframes running CICS software to provide service functions**
  - The proven strengths of CICS running on IBM mainframes has, over time, led to a huge investment in CICS application code and skills that can be reused
  - Performance and scalability of CICS business processing has been proven over 30 years
- **To be agile in today's fast-moving marketplace, companies must embrace modern architectures and new technologies while leveraging both people and computing assets**
  - CICS maintains and adds value by enabling traditional, proven assets and processes to be consumed as Web Services
  - Unlock critical IT assets and re-purpose them to participate in a service oriented architecture
  - Opening access to existing fine-grained applications as coarse-grained business functions, while maintaining QoS
  - Provide a layer of abstraction between service consumer and application implementation / user interface
  - Foster SOA skills in traditional developers
- **Increase Productivity**
  - Build libraries of annotated components representing current assets
  - Rapidly assemble new applications out of existing components using graphical tools
  - Exploit existing developer skills and literacy

## ...continued



# What is XSE?

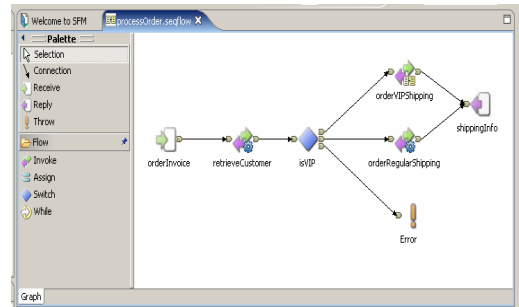
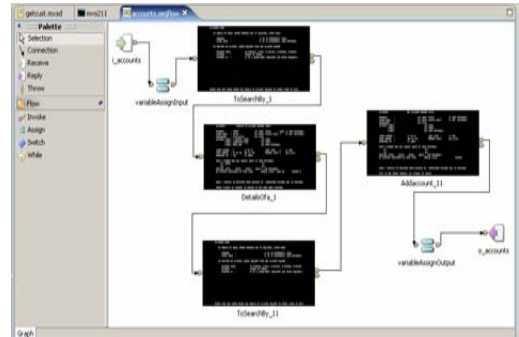
- **XML Services for the Enterprise is a tool to adapt COBOL-based applications so that they become Web services and can process and produce XML messages.**
- **It allows you to:**
  - Create a New Web Service Interface
  - Map to an Existing Web Service Interface
  - Create a New Web Service Implementation
- **Supported runtimes:**
  - Web Service for CICS (CICS TS 3.1)
  - SOAP for CICS (CICS TS 2.2, 2.3, 3.1)
  - IMS SOAP Gateway
  - Batch, TSO, and USS



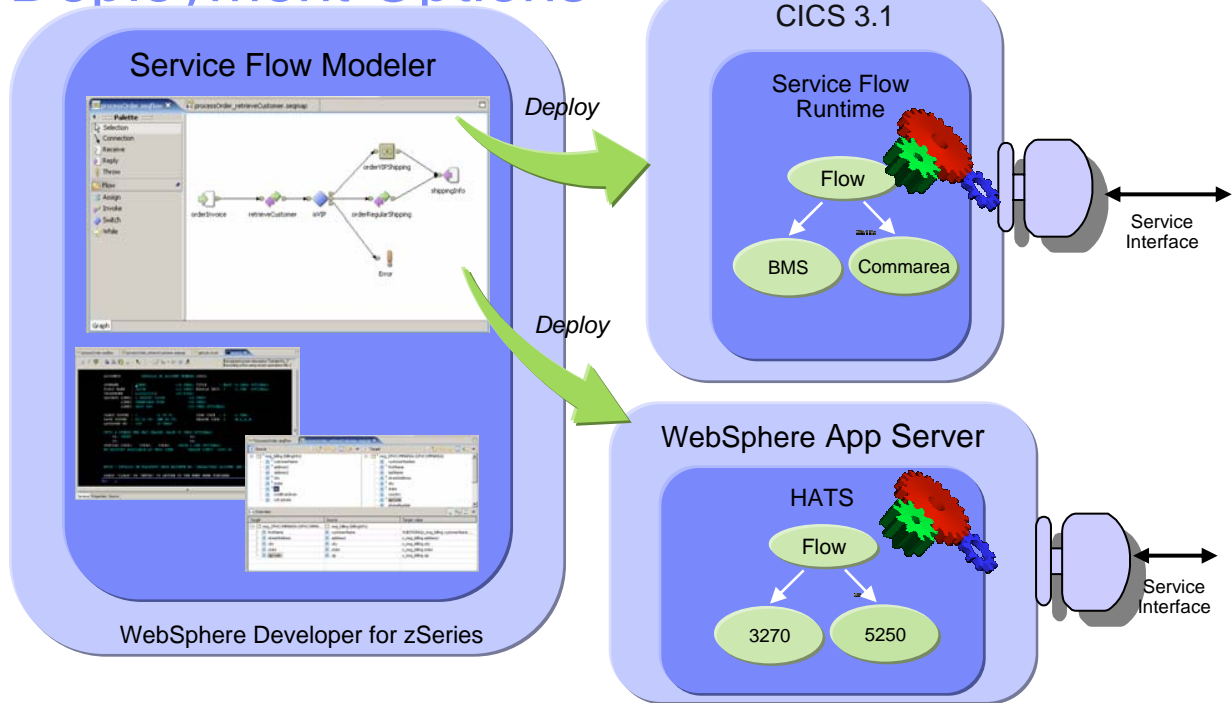
*The simplest mechanism for creating atomic Web Services*

# What is SFM?

- **Service Flow Modeler is a tool to build service flows out of your existing **Commarea and Terminal** based CICS applications.**
- **It allows you to:**
  - Model business or application processes
  - Implement business processes by aggregating multiple transaction invocations, terminal interactions, and sub-flows
  - Deploy these aggregations to runtimes in CICS Transaction Server V3.1 or WebSphere Application Server.
  - Choose to deploy business process as a web service.
- **Development concepts consistent with other SOA development tasks**



# WDz's Service Flow Modeler Deployment Options



# Choosing the Right Tool for CICS SOA

Feature	Host Access Transformation Services Creates web user interfaces from existing terminal based applications	WDz – Service Flow Modeler Assembles multiple commarea and terminal based applications into a single Web service	WDZ – XML Services for the Enterprise XML and Web services enablement for existing commarea-based applications
Builds Web User Interfaces	Yes	No	No
Exposes screen applications as web services	Yes	Yes	No
Exposes commarea programs as web services	No	Yes	Yes
Supports assembly of multiple interfaces	No	Yes	No
z/OS Apps Supported	Any with 3270 interface	CICS BMS and Commarea	CICS Commarea IMS Apps
WAS runtime	Yes	No	No
CICS/IMS native runtime	No	Yes (CICS)	Yes (CICS and IMS)
Skills	Eclipse/J2EE	Integration Developer COBOL/CICS/Eclipse	Integration Developer COBOL/IMS/CICS/Eclipse

# Components of the Service Flow Feature

## What is the SFF

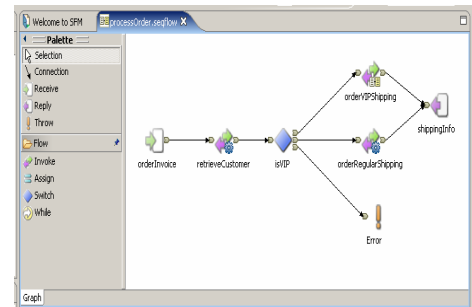
- The CICS Service Flow Feature is a no-charge, orderable feature for CICS TS v3.1
- A business service integration adapter for all CICS applications

## Tooling

- A graphical modeling integrated development environment
- WebSphere Developer for zSeries v6.0.1
  - Service Flow Modeler
  - XML Services for the Enterprise (XSE)
- Entitlement to 10 LIMITED product licences

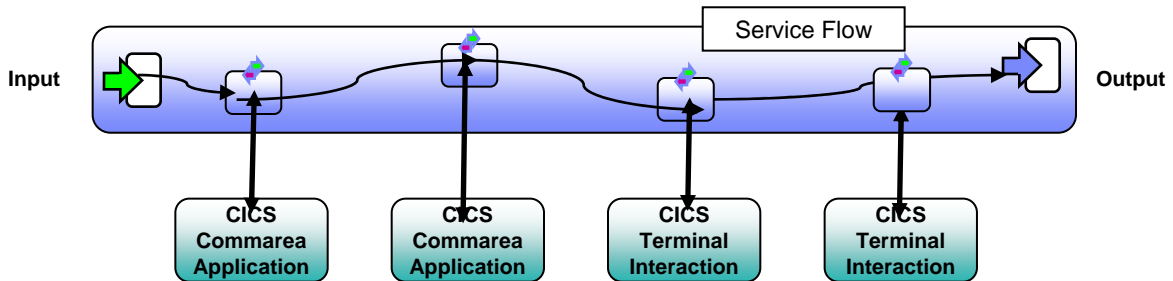
## Runtime

- CICS Service Flow Runtime
- Extends the CICS TS v3.1 environment.
- Offers adapters to invoke CICS commarea based programs and terminal-oriented transactions















# What is a Service Flow?

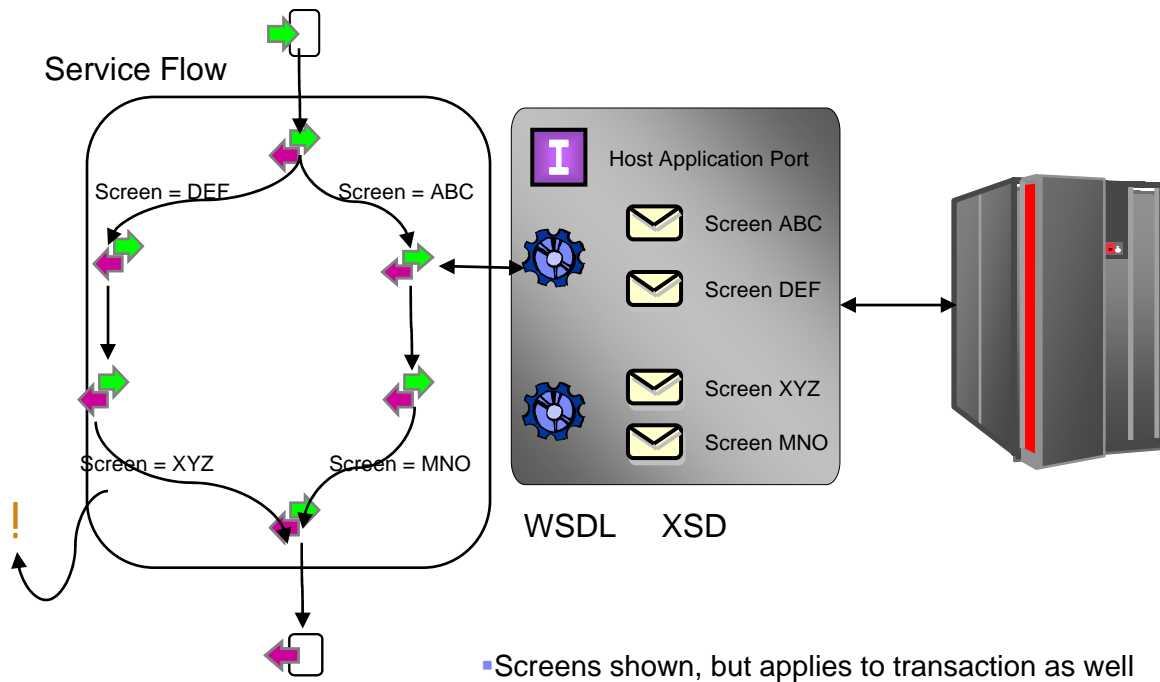


- A service flow is a non-interruptible micro-flow that is constructed from a collection of nodes that represent the invocation of CICS resources
- The flow describes the navigation of the nodes and allows data mapping between the nodes
- A single request may cause the execution of many CICS resources
- Allows for the development of coarse grained services from fine grained resources

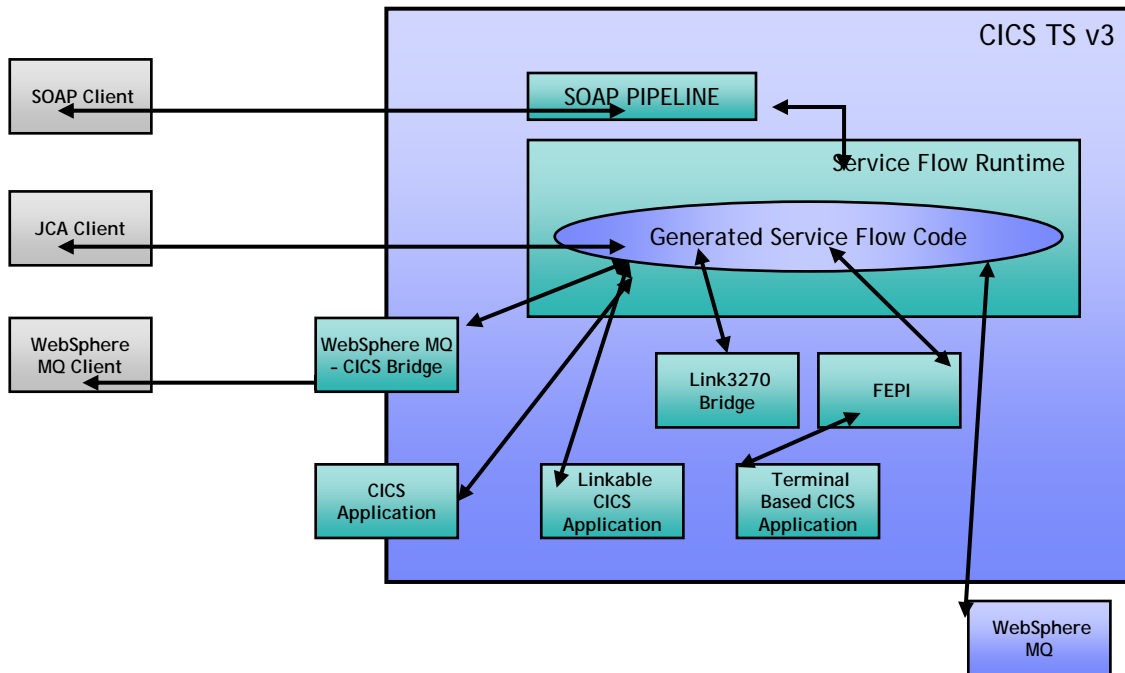
# Components of a Service Flow

	<b>Receive</b>	<b>Input node for flow</b>
	<b>Reply</b>	<b>Output node from flow</b>
	<b>Throw</b>	<b>Throw fault node</b>
	<b>Invoke</b>	<b>Generic invoke (target not yet defined)</b>
	<b>Invoke Operation</b>	<b>Invoke a non-terminal operation</b>
	<b>Invoke Terminal</b>	<b>Invoke a terminal screen operation</b>
	<b>Invoke Flow</b>	<b>Invoke a flow</b>
	<b>Assign</b>	<b>Map data between messages in the flow</b>
	<b>Switch</b>	<b>Flow control decision node</b>
	<b>While</b>	<b>Flow control loop node</b>

# Development Model



# Deployment Model



# Using Service Flow Modeler

- **Service Flow Project**

- Create, store and maintain required resources.
  - Interface Definitions, Terminal applications, Non-Terminal applications

- **Service Flow Model terminology**

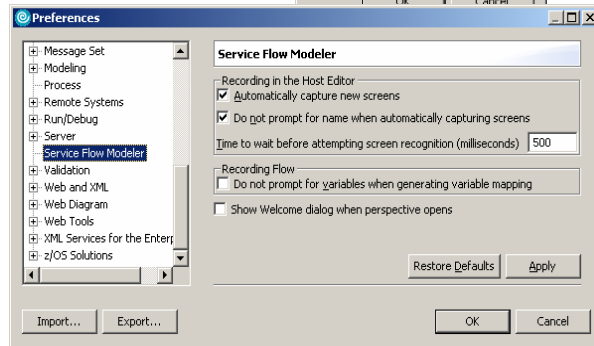
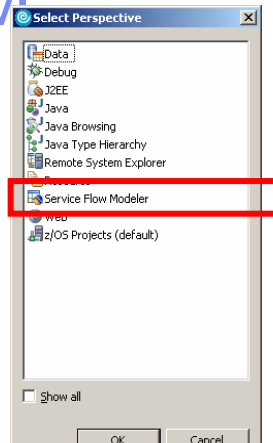
- **Messages** - data structures
  - Screens, transaction inputs, transaction outputs, and web service messages
- **Message mappings** - any transformation of data between messages
- **Operations** - EIS interaction
  - Screen submission, transaction invocation, and sub-sequence flow invocation
- **Flows** – composition of multiple operations and the message mapping

- **Flow behavior is represented graphically**

- Data movement is done through Message Mappings

# Gaining Perspective on SFM

- **The Service Flow Modeler Perspective is used for development**
- **Access through the menu option Window > Open Perspective > Other or the toolbar button**
- **Contains simplified project view and necessary supporting views to access all functions available in SFM**
- **SFM Preferences are also available which pertain to various functions**
  - Access through the menu option Window > Preferences > Service Flow Modeler



The screenshot displays the Service Flow Modeler interface for a project named 'LookUpPhoneFlow.seqflow'. The interface is divided into several key areas:

- Navigator:** Located at the top left, it shows a hierarchical tree of the project structure, including folders for Flows, Mapping, Generation, and Messages. A red box highlights the 'Navigator' label.
- Editor:** The central workspace where the workflow is designed. It features a Palette on the left with categories like Selection, Connection, Receive, Reply, Throw, Flow, Assign, Invoke, Switch, and While. The main diagram shows a flow starting with 'inmsg', followed by a connector, then 'getAccountNum', a connector, 'getPhoneFlow', a connector, 'Assign', and finally 'outmsg'. A large blue box labeled 'Editor' is overlaid on this area.
- Properties View:** Located at the bottom right, it contains a table for viewing and editing properties of the selected element. The table has columns for 'Property' and 'Value'. A large blue box labeled 'Properties View' is overlaid on this area.
- Outline:** Located at the bottom left, it provides a detailed view of the selected element's structure. A large blue box labeled 'Outline' is overlaid on this area.

Additional interface elements include a menu bar (File, Edit, Flow, View, Palette, Navigate, Search, Project, Run, Window, Help), a toolbar with various icons, and a status bar at the bottom indicating 'Writable'.

# Development Tools

The screenshot displays the IBM Business Process Manager (BPM) development environment. The main window shows a process flow diagram for 'processOrder.seqflow'. The flow starts with an 'orderInvoice' event, followed by a 'retrieveCustomer' activity. A decision diamond 'isVIP' branches the flow into 'orderVIPShipping' and 'orderRegularShipping', both leading to a 'shippingInfo' event. An 'Error' event is also shown at the end of the flow.

Red arrows indicate the mapping between the process flow and the message maps. One arrow points from the 'retrieveCustomer' activity to the 'msg\_billing (billingInfo)' message map in the 'Source' pane. Another arrow points from the 'isVIP' decision diamond to the 'msg\_DFHCOMMAREA (DFHCOMMAREA)' message map in the 'Target' pane.

The 'Operations Editor' on the left shows the 'Operations' tree with 'Ports' expanded, including 'Input', 'Output', and 'Faults'.

The 'Structure' pane for 'getcust.mxsd' shows the following structure:

Structure	Type
getcust.mxsd	
Messages	
msg_STOCK	STOCK
msg_DFHCOMMAREA (DFHCOMMAREA)	DFHCOMMAREA
customerNumber	xsd:string
firstName	xsd:string
lastName	xsd:string
streetAddress	xsd:string
city	xsd:string
state	xsd:string
country	xsd:string
zipCode	xsd:string
phoneNumber	xsd:string
returnCode	xsd:string
numberOfStoc...	xsd:short
quantityOfShares	xsd:short
4trk5vmhsk	xsd:string

The 'Source' pane shows the 'msg\_billing (billingInfo)' message map with the following structure:

- customerName
- address1
- address2
- city
- state
- zip
- creditcardnum
- ccExpDate

The 'Target' pane shows the 'msg\_DFHCOMMAREA (DFHCOMMAREA)' message map with the following structure:

- customerNumber
- firstName
- lastName
- streetAddress
- city
- state
- country
- zipCode
- phoneNumber

The 'Overview' table at the bottom shows the mapping between the source and target message maps:

Target	Source	Target value
msg_DFHCOMMAREA (DFHCOMMAREA)	msg_billing (billingInfo)	
firstName	customerName	SUBSTRING(s_msg_billing.customerName ...
streetAddress	address1	s_msg_billing.address1
city	city	s_msg_billing.city
state	state	s_msg_billing.state
zipCode	zip	s_msg_billing.zip



# Mapping Editor

**Compose Mapping Expression**

Use source or target fields and functions to create value for target

Mapping: ca\_userid --> ca\_userid  
Value of target field: t\_msg\_CAPLACEORDER.ca\_order\_request.ca\_userid

s\_msg\_PLACEORDERREQUEST.ca\_userid

Apply Condition

s\_msg\_PLACEORDERREQUEST.ca\_userid IS NOT NULL

Source fields » String Functions

OK Apply Cancel

# Specialized Terminal Tools

The screenshot displays the IBM Worklight Studio interface for a multi-screen application. The main window shows a sequence of screens: 'accounts.seqflow', 'processOrder.seqflow', 'processOrder\_retrieveCustomer.seqmap', 'getcust.mxsd', and 'mvs211'. A 'Palette' on the left contains actions like Selection, Connection, Receive, Reply, Throw, Flow, Invoke, and Assign. A 'Structure' pane shows a tree view of fields and screens. A 'Properties' pane shows details for a field.

The 'Screen Operations Editor' is overlaid on the 'DetailsOfa\_1' screen. It shows a tree view of screen interactions, with 'Extract: Field337' selected. The editor's main area displays the screen's content, which includes account details and a list of other users who may charge.

**Screen Content (DetailsOfa\_1):**

```

ACCOUNTS          DETAILS OF ACCOUNT NUMBER 10001
SURNAME           : JONES           (10 CHRS) TITLE   : MAST (4 CHRS OPTIONAL)
FIRST NAME        : JACOB           (12 CHRS) MIDDLE INIT: J (1 CHR OPTIONAL)
TELEPHONE         : 1122223333      (10 DIGS)
ADDRESS LINE1: 2 PARTNY CLOSE      (14 CHRS)
LINE2: CHANDLERS FOR                (4 CHRS)
LINE3: SA99 4SS                     (4 CHRS)
CARD ISSUED: 1 (1 TO 5)             CARD CODE: 6 (1 CHR)
DATE ISSUED: 02 02 99 (MM DD YY)    REASON CODE: L (M,L,S,R)
APPROVED BY: JJO (3 CHRS)
SPECIAL CODE1: CODE2: CODE3: (EACH 1 CHR OPTIONAL)
NO HISTORY AVAILABLE AT THIS TIME    CHANGE LIMIT 1000.00 STATUS B
NOTE:- DETAILS IN BRACKETS SHOW MAXIMUM NO. CHARACTERS ALLOWED AND IF OPTIONAL
PRESS "CLEAR" OR "ENTER" TO RETURN TO THE MENU WHEN FINISHED
  
```

**Screen Operations Editor Tree:**

- DetailsOfa\_1
  - Screen Interaction 0
    - ToSearchBy\_1
      - Screen Interactions
        - Screen Interaction 0
          - Extract: Field337
          - AID key: [clear]

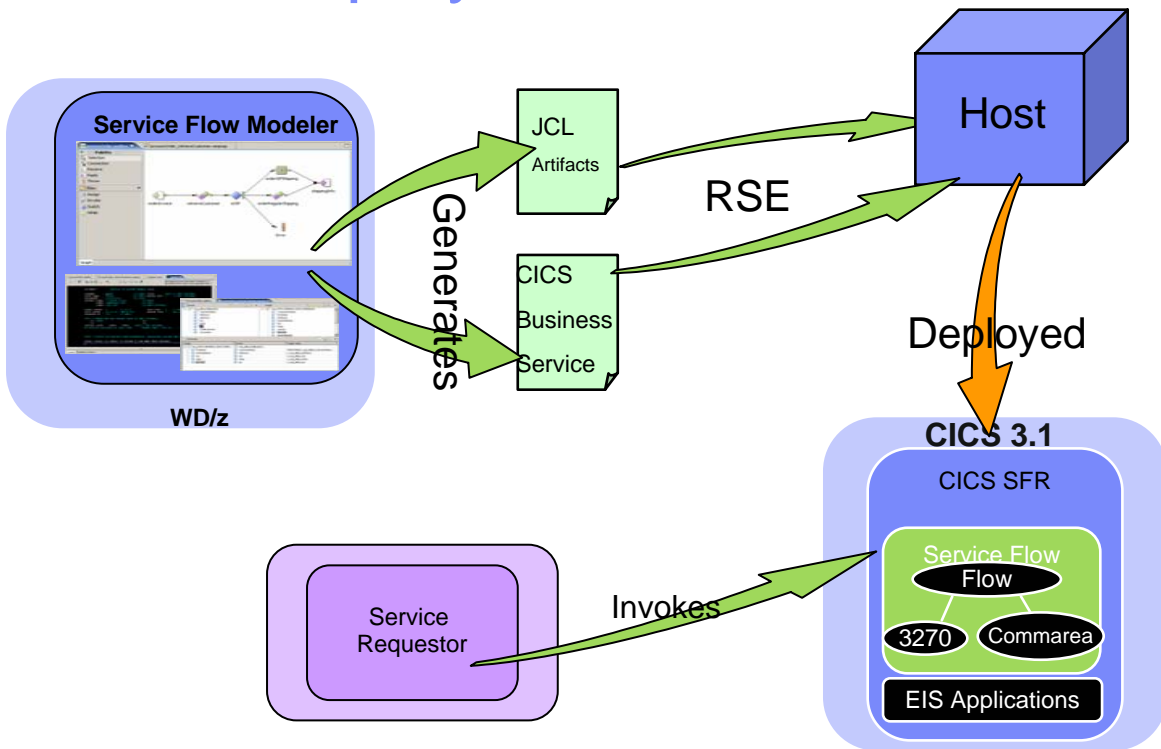
**Properties Pane:**

Property	Value
Bidirectional	false
Contents	ACCOUNT
Length	17
Name	Field2
Position	2
Protected	true

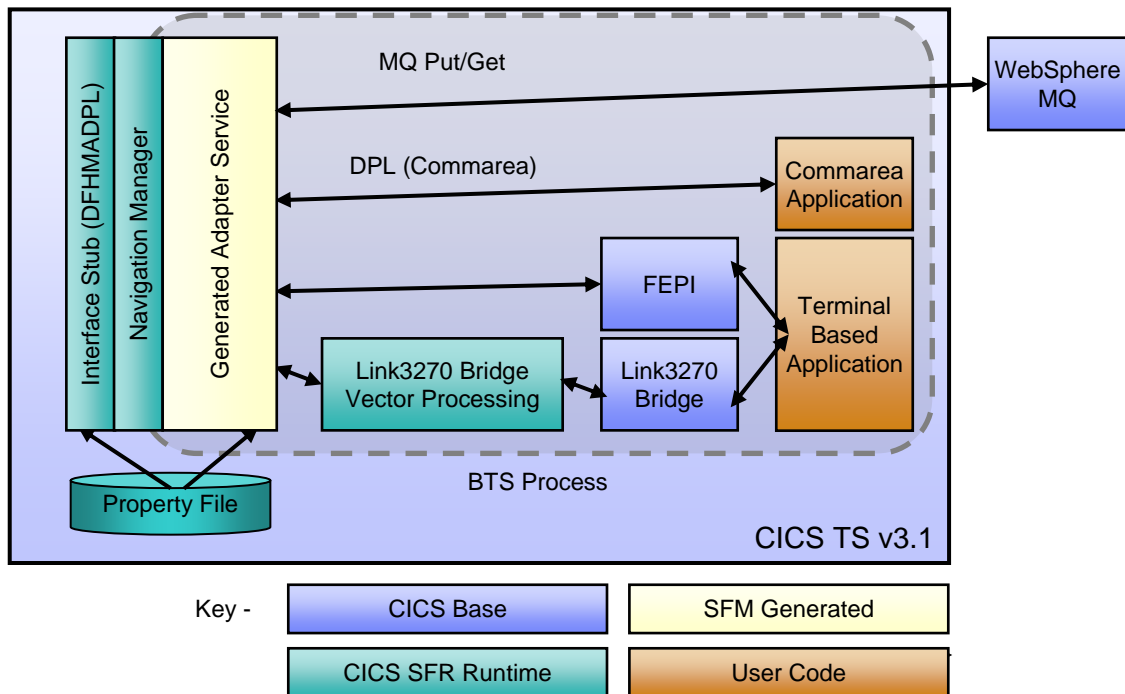
**Field Properties Table:**

Property	Value
Field	DetailsOfa_Inst...
Name	Field337

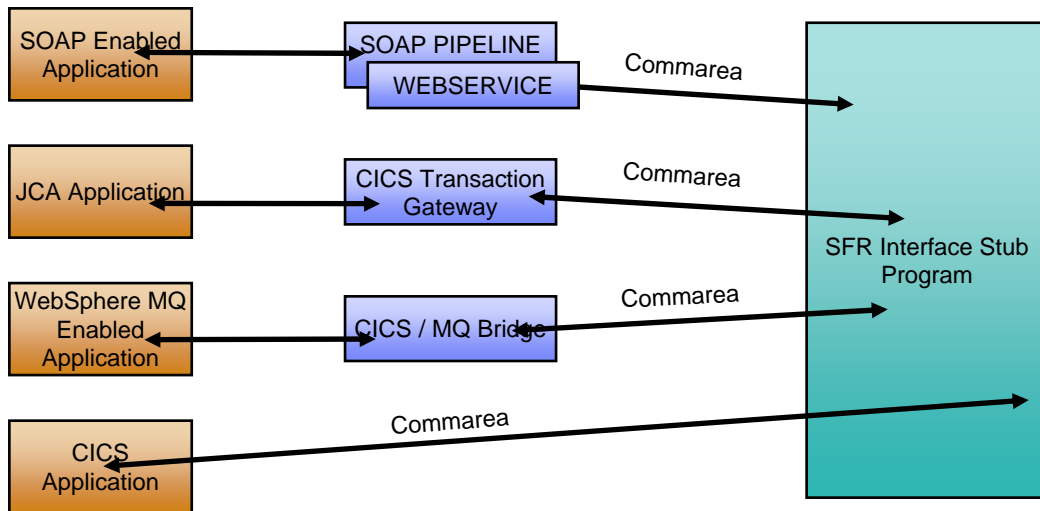
# Model, Deploy and Invoke



# Runtime Architecture

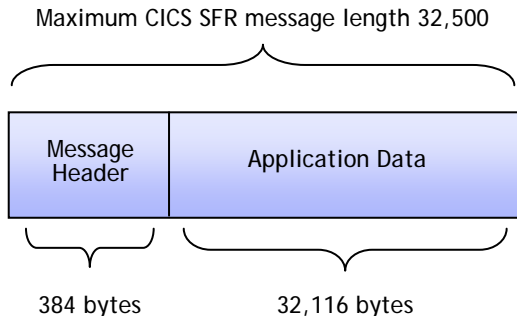


# Invoking a CICS Business Service



# Invoking a CICS Business Service

- **Flow Invoked through program DFHMADPL**
- **Message passed from Service Requestor to CICS SFR**



- **Message Header contains request and reply**
  - Request Name, Data Length
  - Return Code, Error Message

# CICS Service Flow Feature and IBM SOA with WID

- **Common artifacts**
  - WSDL
  - XSD
- **Common Developer experience**
  - Eclipse based tooling
  - Common modeling constructs
    - Receive/Reply
    - Invoke
    - Switch
- **Easy integration of Service Flow**

....An example with  
IBM WebSphere Integration Developer (WID) follows ....

# Import deployed WSDL in WebSphere Integration Developer

**Import SFM generated WSDL**

**Invoke SFM's Webservice using imported WSDL**

**Process Server Business Process**

**Service Interface**

**Service Interface**

**Similar development units (such as assignments)**

Business Integration - CO - IBM WebSphere Integration Developer

Business Integrat... Navigator | Assembly Diagram: CatalogOrder | BPD

Business Integrat... | Interface Part... | Client | Reference Part... | ItemRetriever | catalogPortType | fallbackOrder | Variables | ProcessOrderPart... | ProcessOrderResp... | InputItem | outputItem | catpp | operatorMessage | Combination Set...

Properties | Problems (Server) Console Debug

Details

From: variable

- ProcessOrderResponse : operation(Response)
- outputItem : short
- inputItem : string
- ProcessOrderParameters : operation()
- catout : SERVICEOUTPUTAREA
- o\_orderItem : serviceoutputarea\_orderItem
- navigator\_output\_area : serviceoutputarea\_navigator\_output\_area
- o\_catalogorder : serviceoutputarea\_navigator\_output\_area\_o\_catalogorder

type

Query: [navigator\_output\_area.catalogorder.itemMessage

variable

- ProcessOrderResponse : operation(Response)
- o\_orderItem : string
- outputItem : short
- inputItem : string
- ProcessOrderParameters : operation()
- catout : SERVICEOUTPUTAREA
- catpp : SERVICEOUTPUTAREA
- operatorMessage : string

Query: [status



# Where to get more information?

- **WDz Help -> Help Contents**
  - Developing z/OS applications and Web Services interfaces
  - Topics and Tutorials
- **CICS TS 3.1 Infocenter**
  - <http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/index.jsp>
  - Access to CICS
    - CICS Web Services
    - CICS Internet Guide
  - CICS Service Flow Runtime
- **IBM WebSphere Application Transformation Demos**
  - [http://websphere.dfw.ibm.com/atdemo/atdemo\\_wsed.html](http://websphere.dfw.ibm.com/atdemo/atdemo_wsed.html)
  - Viewlet Demos available for download

Thank  
YOU

# SFM Demo Descriptions

- Scenario 1: This is a demonstration of the rapid application development capabilities of the Service Flow Modeler component of WebSphere Developer for zSeries v6.0.1. In this demo, we quickly implement a new CICS application by aggregating existing COMMAREA programs, then expose that application as a Web service.
- Scenario 2: This is a demonstration of the rapid application development capabilities of the Service Flow Modeler component of WebSphere Developer for zSeries v6.0.1. In this demo, we quickly implement a new CICS application by automating the navigation of an existing 3270 terminal application, then expose that application as a Web service.
- Scenario 3: This is a demonstration of the flexibility and rich functionality of the Service Flow Modeler component of WebSphere Developer for zSeries v6.0.1. In this demo, we show how a business analyst can model a workflow which will implement a specified Web service definition. Then, we show how an application developer can implement that workflow out of existing CICS applications, both terminal and nonterminal. Finally, we show a systems programmer deploying this service to the CICS Service Flow Runtime.