WebSphere software

IBM

e business software

WSED POT

São Paulo - Brazil

Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

Purpose

- This is a no charge event to help customers to be familiarized with WebSphere Studio Enterprise Developer version 5 (WSED)
- See product details at http://www-3.ibm.com/software/ad/studioenterprisedev/
- WSED is the follow on of VisualAge COBOL and in near future for VisualAge Generator.
- Also WSED is the right tool to implement applications with WebSphere without Java skills
-

WebSphere software

IBM

Audience Pre-requisites

- Audience: Architects, Technical Specialist, and COBOL developers, responsible for building their applications using J2EE architecture to be deployed into WebSphere Application Server V5 with business logic in the mainframe (z/OS). Also Java developers interested on the STRUTS tools.
- Pre-requisites: Understand J2EE architecture, some z/OS experience or understanding. It is desired to have introductory knowledge of WebSphere Studio Application Developer. No Java skills are required. COBOL and 4GL programmers (CSP, VAGEN) are welcome

WebSphere software

Abstract

The presentations will cover

- " WebSphere Family Product Overview
- " Struts Tools
- " Web Diagram Editor
- " Enterprise Generation Language (EGL)
- " z/OS Application Development Tools
- " IDE for Enterprise Developers
- " XML Enablement Enhancements for z/OS
- " WebSphere Studio Asset Analyzer V2

WebSphere software

Lab1 - EGL Development and Test

- This lab exercise will take you through the steps of building a server program using the Enterprise Generation Language (EGL). The EGL server program will be part of a web banking application. The web client part of the application, which is built in LAB2, lets the user enter their account number, enter an amount to withdraw from that account, and then press a button to initiate the withdrawal. The EGL server program is called to perform the withdrawal and return the account's new balance. The web client then displays the confirmation page with the new account balance.

Lab2 - Struts Development with 4GL, Integration, and Test

- This lab will take you through the steps of using the Struts tools of WSED. You will build the client portion of a web application and integrate the Struts classes with the EGL server built in Lab1.
- The application is a web banking application. The client portion of this web application lets the user enter their account number and an amount to withdraw from that account, and then press a button to initiate the withdrawal. After withdrawal has completed, a confirmation page is returned that also shows the new account balance.
-

Lab 3 - Using XML Enablement

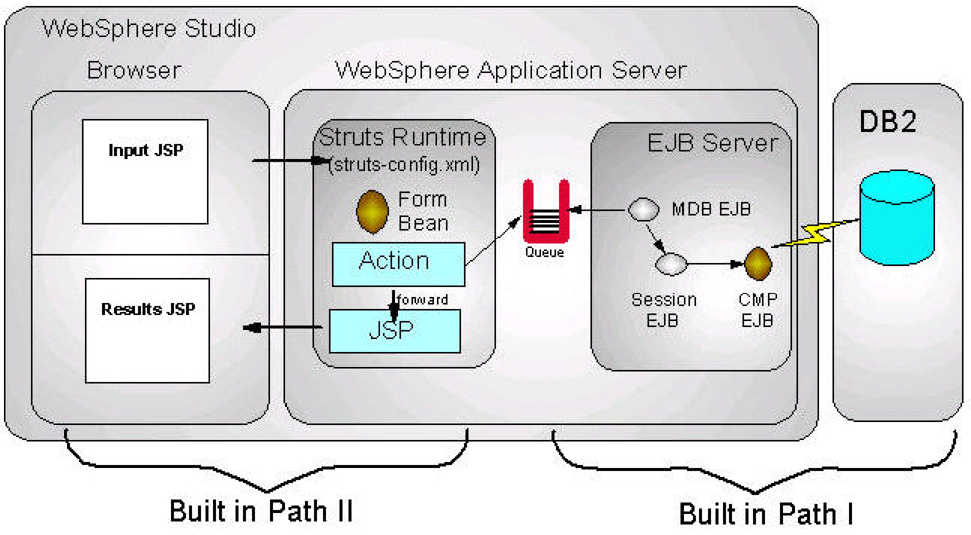
- This lab will take you through the steps of using the XML Enablement component of WSED to create converter programs that will enable existing COBOL applications to process XML data. The workshop will generate both inbound and outbound XML converters for two of the programs described in the white paper "XML For the Enterprise".
-

Lab 4 - Using the z/OS Application Development Tools to Work with Remote Systems

- This lab will take you through the steps of using the z/OS Application Development component of WSED to work with remote systems. It will familiarize you with the z/OS Application Development environment. In this workshop, we will be defining a remote z/OS system, setting up a MVS project, editing, compiling, and debugging a COBOL application.

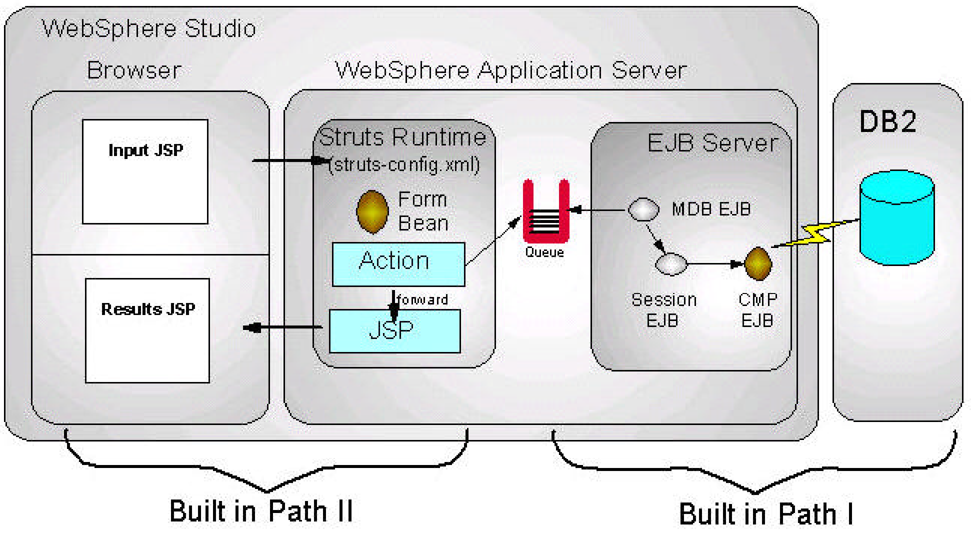
Lab 5 - Optional - Using J2EE and STRUTS

- **This is the Part II**



Lab 5 - Optional - Using J2EE and EJB

- **This is the Part I**



WebSphere software

IBM

e-business software

WebSphere Studio

Positioning the Family

Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

The Family WebSphere Studio

$$\begin{tikzcd} \text{SD} \arrow[r] & \text{AD} \arrow[r] & \text{IE} \arrow[r] & \text{ED} \end{tikzcd}$$

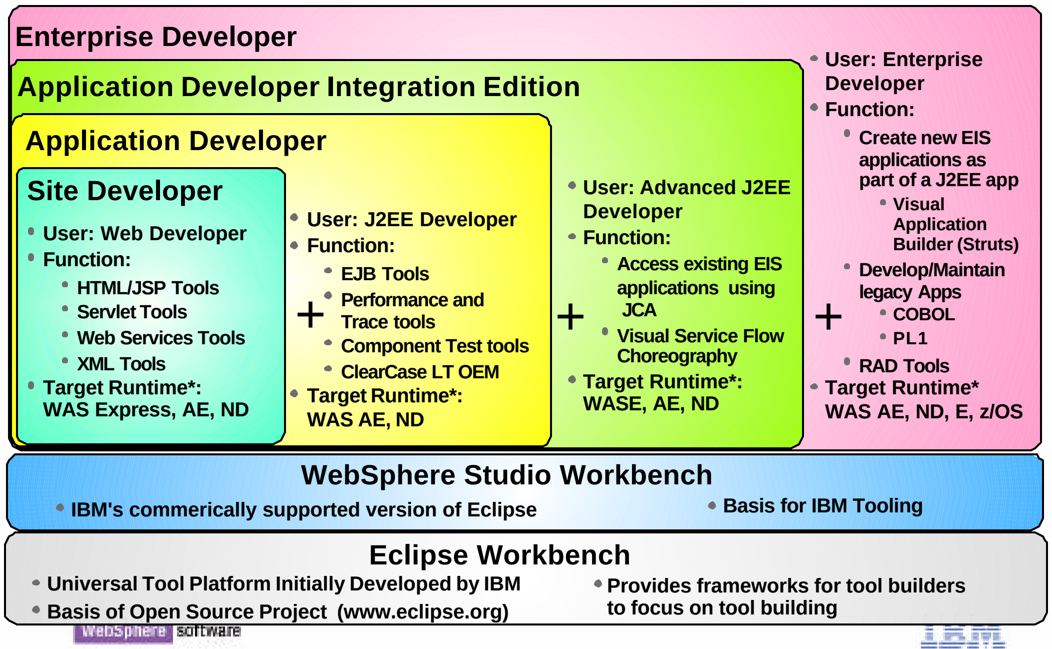
- AD: Profiing/Tracing Tools
- IE: Service Flow Composition
- ED: RAD Tools

- SD = Site Developer (WSSD)
- AD = Application Developer (WSAD)
- IE = Integration Edition (WSAD-IE)
- ED = Enterprise Developer (WSED)

WebSphere software

IBM

WebSphere Studio Family



Enterprise Developer

- User: Enterprise Developer
- Function:
  - Create new EIS applications as part of a J2EE app
    - Visual Application Builder (Struts)
  - Develop/Maintain legacy Apps
    - COBOL
    - PL1
  - RAD Tools
- Target Runtime*
  WAS AE, ND, E, z/OS

Application Developer Integration Edition

- User: Advanced J2EE Developer
- Function:
  - Access existing EIS applications using JCA
  - Visual Service Flow Choreography
- Target Runtime*:
  WASE, AE, ND

Application Developer

- User: J2EE Developer
- Function:
  - EJB Tools
  - Performance and Trace tools
  - Component Test tools
  - ClearCase LT OEM
- Target Runtime*:
  WAS AE, ND

Site Developer

- User: Web Developer
- Function:
  - HTML/JSP Tools
  - Servlet Tools
  - Web Services Tools
  - XML Tools
- Target Runtime*:
  WAS Express, AE, ND

+ + +

WebSphere Studio Workbench

- IBM's commerically supported version of Eclipse
- Basis for IBM Tooling

Eclipse Workbench

- Universal Tool Platform Initially Developed by IBM
- Basis of Open Source Project (www.eclipse.org)
- Provides frameworks for tool builders to focus on tool building

Customer Profile - WebSphere Studio Site Developer

- **Typical Customers for Site Developer:**
  - ►Customer building Web Applications which does does not require EJBs
  - ►Customers building/integrating Web Services into their web applications
  - ►Customers migrating from WebSphere Studio Classic
  - ►Customers migrating from VisualAge for Java Professional
- **Skills: Java, Web**

Customer Profile - WebSphere Studio Application Developer

- **Typical Customers for Application Developer**
  - ►Customers building J2EE Applications which requires EJBs
  - ►Customers who need Performance/Profiling Tools
  - ►Customers who need component test tools
  - ►Customers migrating from VisualAge for Java Enterprise
- **Skills: J2EE, Java, Web**

WebSphere software

IBM

Customer Profile - WebSphere Studio Application Developer Integration Edition

- **Typical Customers for Integration Edition**
  - ►Customers building J2EE Applications which requires enterprise integration using JCA Resource Adapters:
    - –CICS, IMS, HOD, SAP, etc
  - ►Customers building application which requires service flow composition
  - ►Customers migrating from VisualAge for Java Enterprise who used enterprise access builders
- **Skills: J2EE, Java, Web**

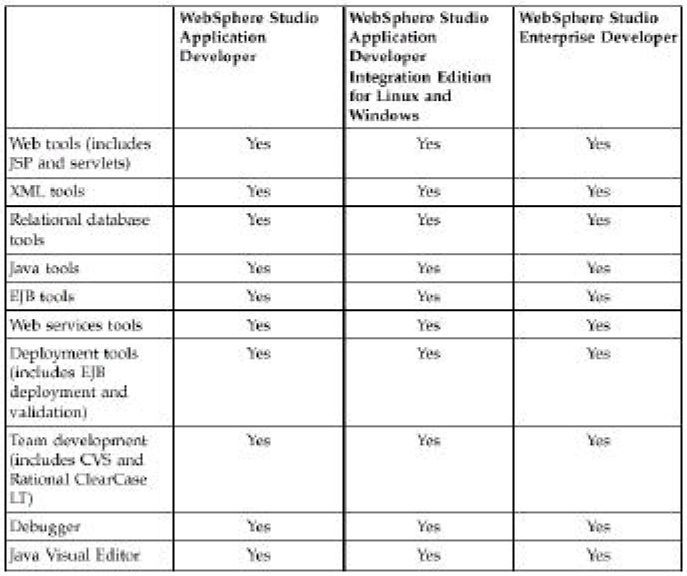
WebSphere software

IBM

Customer Profile - WebSphere Studio Enterprise Developer

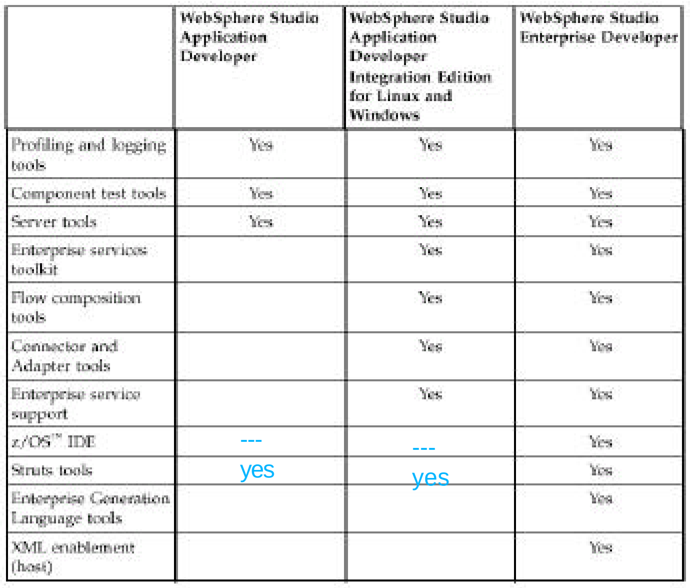
- **Typical Customers for Enterprise Developer**
  - ►Customers building new EIS appliations which require J2EE xxx
  - ►Java developers who rapidly build Struts based applications
  - ►Non-Java programmers who want to build J2EE applications using a 4GL language
  - ►Customers maintaining legacy applications written in COBOL or PL/I
  - ►Customers developing legacy applications written in COBOL or PL/I
  - ►Customers migrating from VisualAge Generator, VA COBOL, or VA PL/I
- **Skills: Java, COBOL, PL/I, J2EE**

Product Configurations



	WebSphere Studio Application Developer	WebSphere Studio Application Developer Integration Edition for Linux and Windows	WebSphere Studio Enterprise Developer
Web tools (includes JSP and servlets)	Yes	Yes	Yes
XML tools	Yes	Yes	Yes
Relational database tools	Yes	Yes	Yes
Java tools	Yes	Yes	Yes
EJB tools	Yes	Yes	Yes
Web services tools	Yes	Yes	Yes
Deployment tools (includes EJB deployment and validation)	Yes	Yes	Yes
Team development (includes CVS and Rational ClearCase LT)	Yes	Yes	Yes
Debugger	Yes	Yes	Yes
Java Visual Editor	Yes	Yes	Yes

Product Configurations...

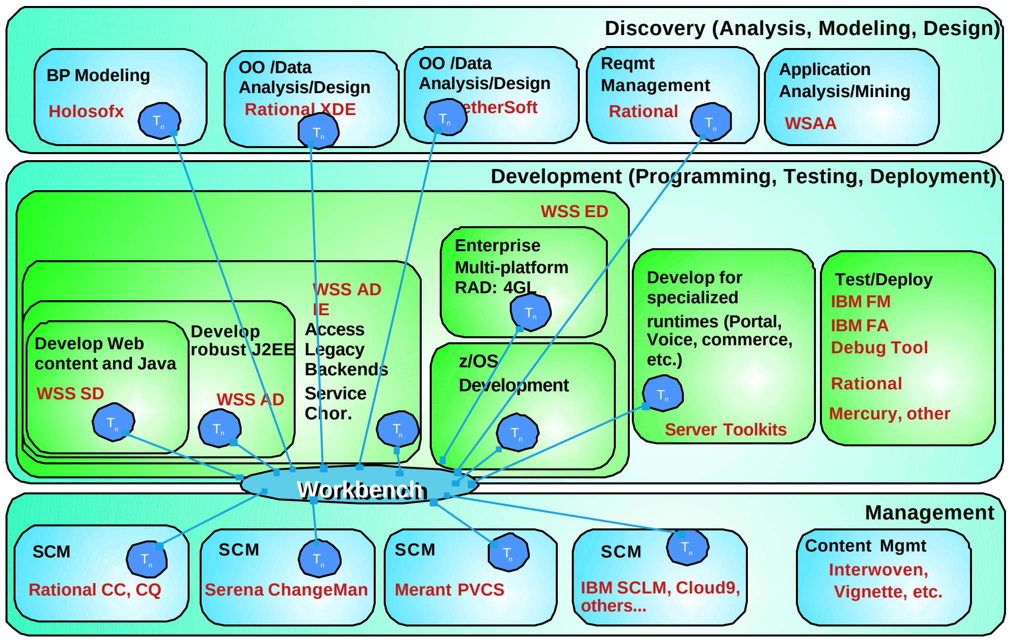


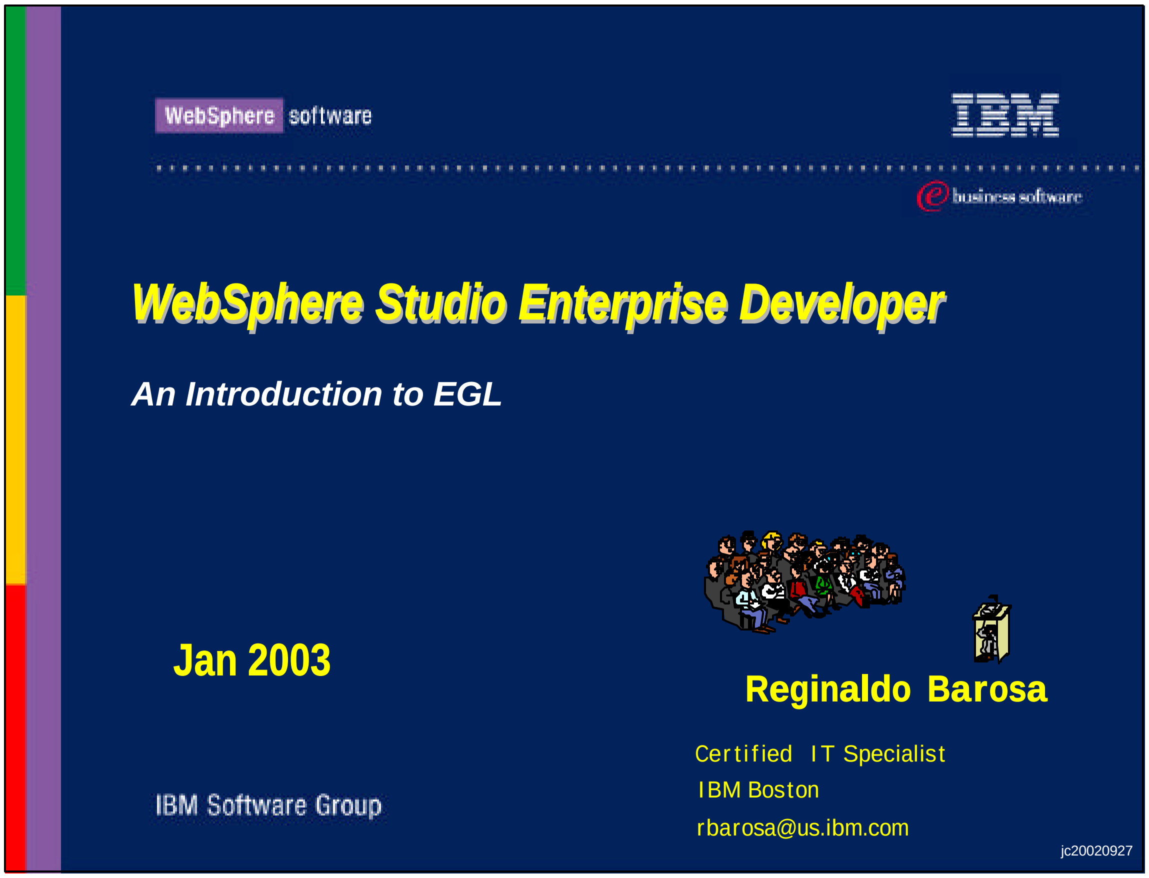
	WebSphere Studio Application Developer	WebSphere Studio Application Developer Integration Edition for Linux and Windows	WebSphere Studio Enterprise Developer
Profiling and logging tools	Yes	Yes	Yes
Component test tools	Yes	Yes	Yes
Server tools	Yes	Yes	Yes
Enterprise services toolkit		Yes	Yes
Flow composition tools		Yes	Yes
Connector and Adapter tools		Yes	Yes
Enterprise service support		Yes	Yes
z/OS™ IDE	---	---	Yes
Struts tools	yes	yes	Yes
Enterprise Generation Language tools			Yes
XML enablement (host)			Yes

WebSphere software

IBM

WebSphere Studio: The Comprehensive Platform




WebSphere software

IBM

e business software

WebSphere Studio Enterprise Developer

An Introduction to EGL

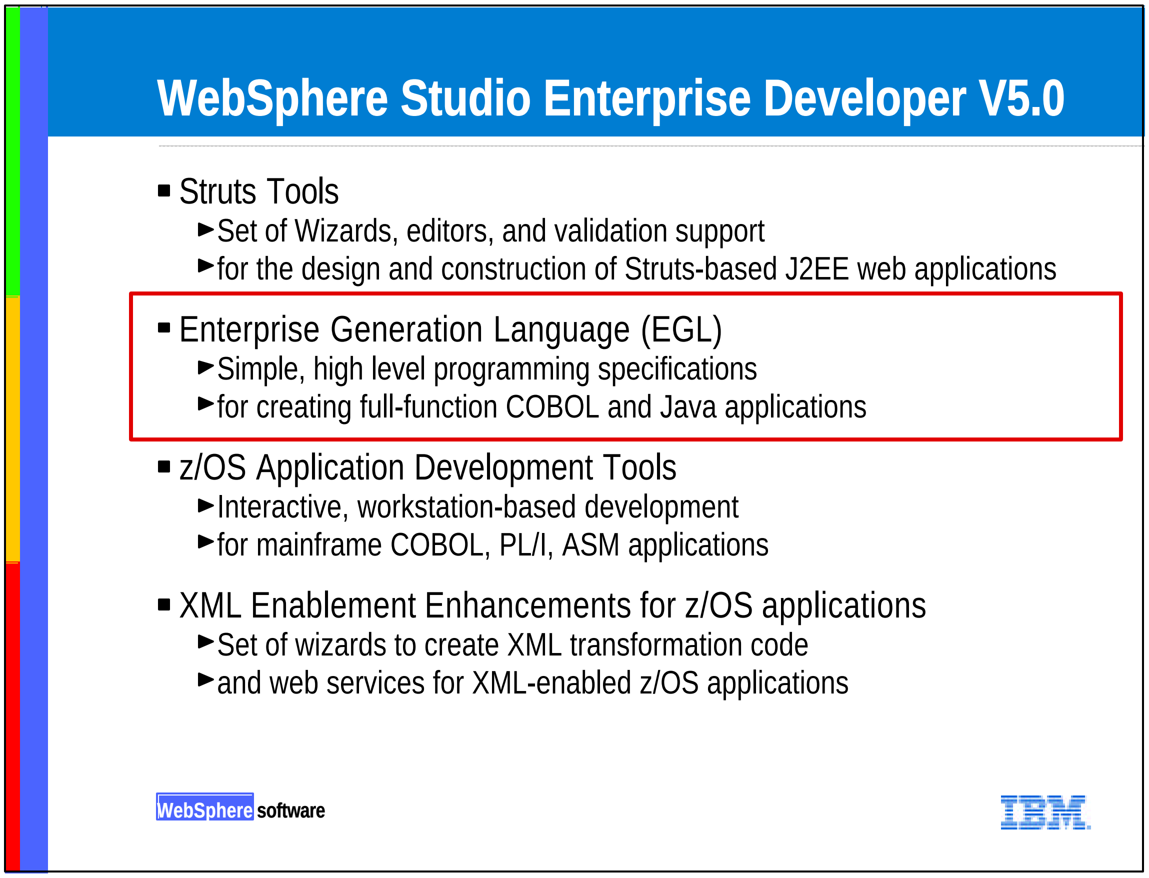
Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

jc20020927

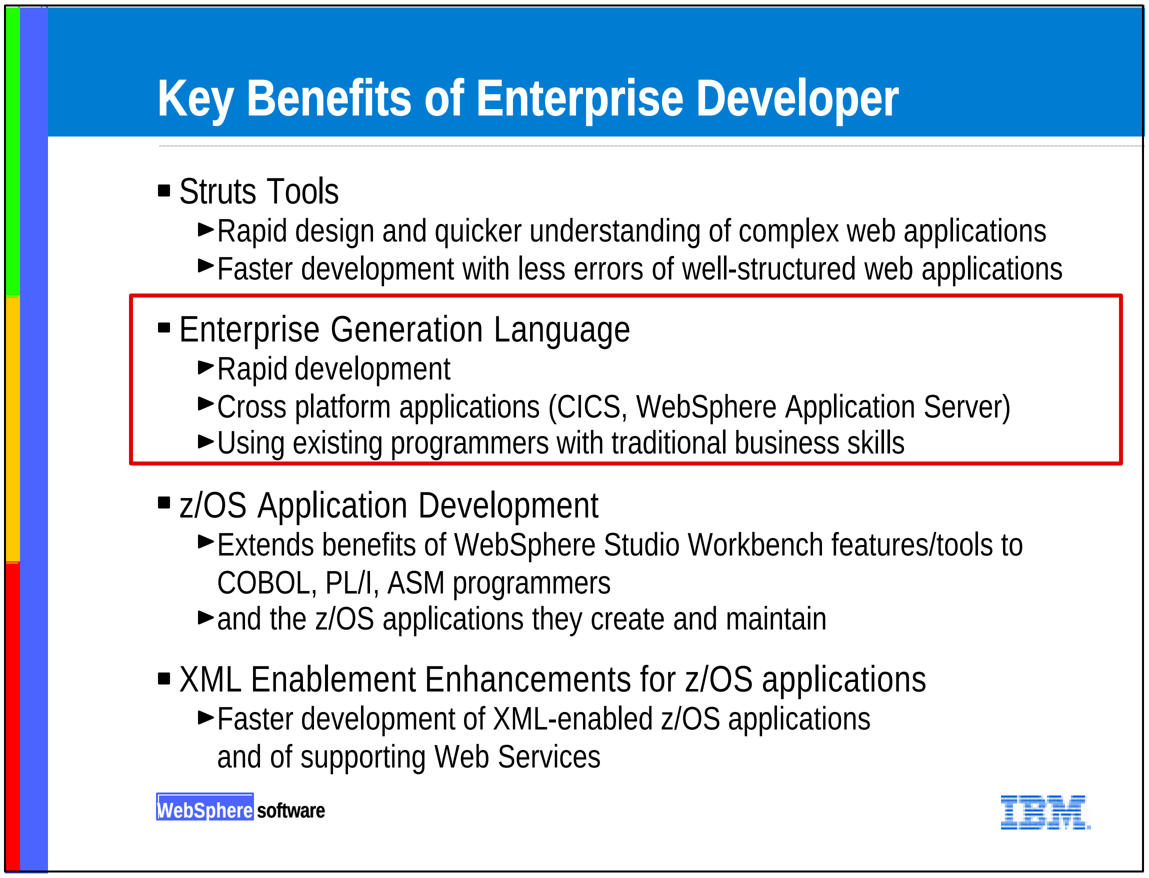


WebSphere Studio Enterprise Developer V5.0

- Struts Tools
  - ►Set of Wizards, editors, and validation support
  - ►for the design and construction of Struts-based J2EE web applications
- Enterprise Generation Language (EGL)
  - ►Simple, high level programming specifications
  - ►for creating full-function COBOL and Java applications
- z/OS Application Development Tools
  - ►Interactive, workstation-based development
  - ►for mainframe COBOL, PL/I, ASM applications
- XML Enablement Enhancements for z/OS applications
  - ►Set of wizards to create XML transformation code
  - ►and web services for XML-enabled z/OS applications

WebSphere software

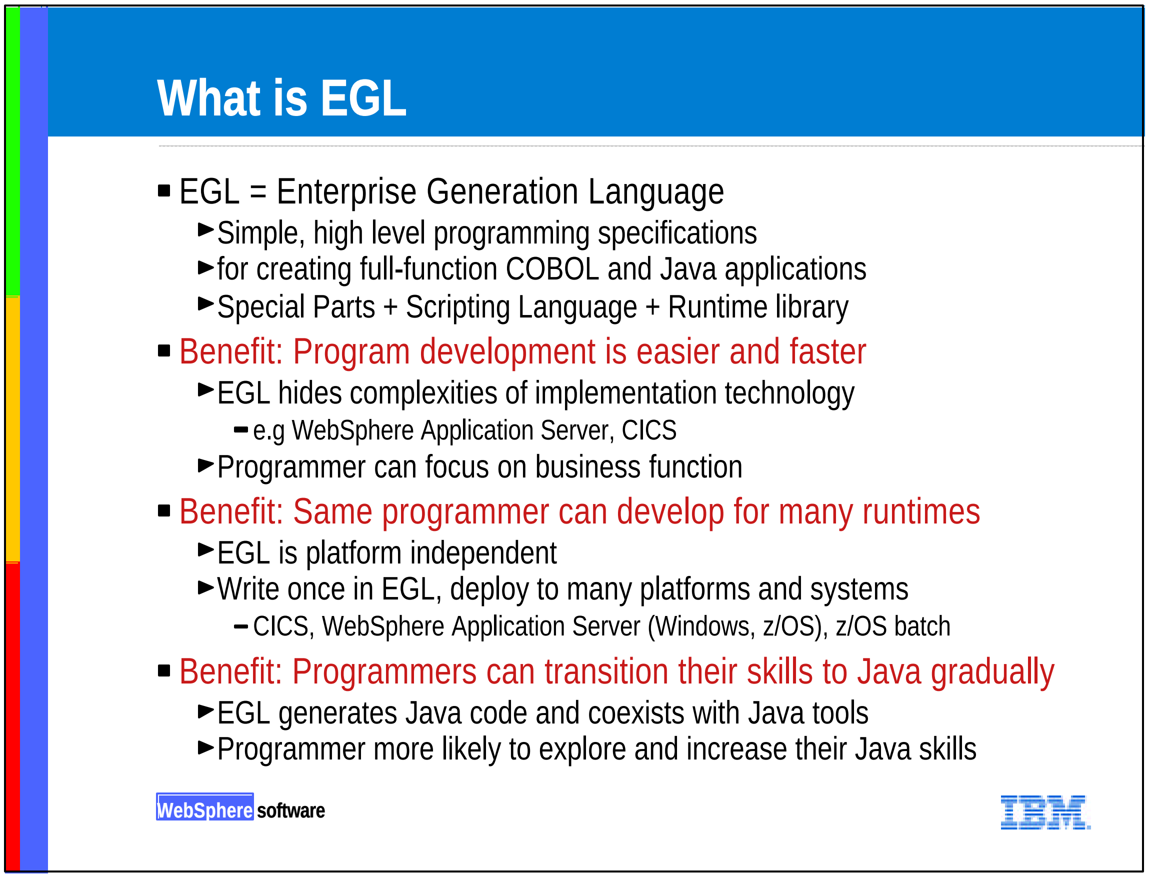
IBM



Key Benefits of Enterprise Developer

- Struts Tools
  - ►Rapid design and quicker understanding of complex web applications
  - ►Faster development with less errors of well-structured web applications
- Enterprise Generation Language
  - ►Rapid development
  - ►Cross platform applications (CICS, WebSphere Application Server)
  - ►Using existing programmers with traditional business skills
- z/OS Application Development
  - ►Extends benefits of WebSphere Studio Workbench features/tools to COBOL, PL/I, ASM programmers
  - ►and the z/OS applications they create and maintain
- XML Enablement Enhancements for z/OS applications
  - ►Faster development of XML-enabled z/OS applications and of supporting Web Services

WebSphere software



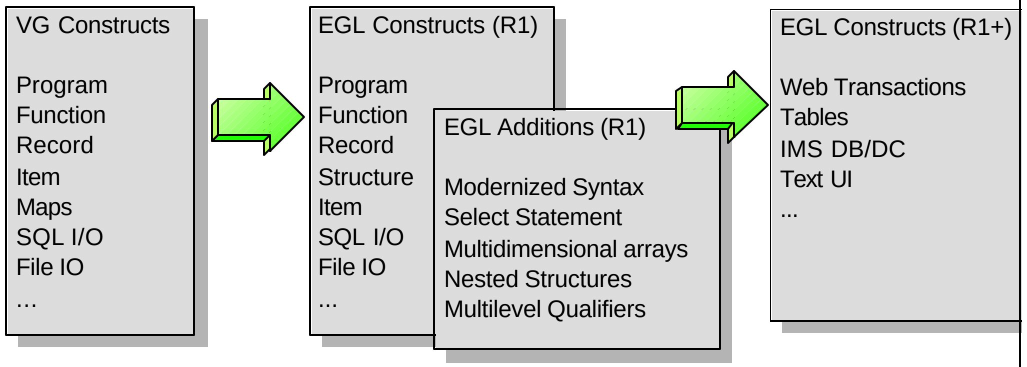
What is EGL

- EGL = Enterprise Generation Language
  - ►Simple, high level programming specifications
  - ►for creating full-function COBOL and Java applications
  - ►Special Parts + Scripting Language + Runtime library
- Benefit: Program development is easier and faster
  - ►EGL hides complexities of implementation technology
    - – e.g WebSphere Application Server, CICS
  - ►Programmer can focus on business function
- Benefit: Same programmer can develop for many runtimes
  - ►EGL is platform independent
  - ►Write once in EGL, deploy to many platforms and systems
    - – CICS, WebSphere Application Server (Windows, z/OS), z/OS batch
- Benefit: Programmers can transition their skills to Java gradually
  - ►EGL generates Java code and coexists with Java tools
  - ►Programmer more likely to explore and increase their Java skills

WebSphere software

Enterprise Generation Language

- ►Migration path from VisualAge Generator's 4GL (end of 2003)
  - with significant additions and enhancements necessary for robust e-business Web Application development
- ►A new name, linked to IBM's WebSphere strategy
  - reflects the broad platform AD support that Enterprise customers need



Bringing forward and enhancing the core technologies created over the last 20+ years from CSP to VisualAge Generator

WebSphere software IBM

High Level Specifications

- Simple specifications
  - ►faster development
  - ►hides technology
- Deployment flexibility
  - ►run on platform of choice
  - ►lowers skill requirement

ADD CUSTINFO

WSED
MQ Connect Queue
MQ Open Queue
MQ Put
Intercept errors
MQSeries

CALL CUSTPGM DATA

WSED
Marshal data
Convert data
Invoke CUSTPGM
(CICS, IP, APPC, ..)
Communications

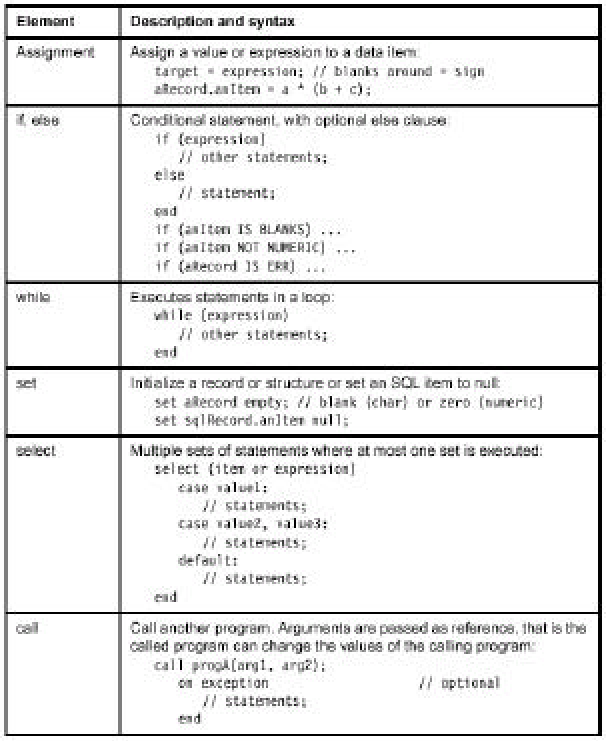
INQUIRY CUST_TABLE

WSED
Declare Cursor
Open Cursor
Fetch
Close
SQL

Benefit: Same programmer can develop for multiple platforms

WebSphere software IBM

EGL elements



Element	Description and syntax

Assignment	Assign a value or expression to a data item:
	`target = expression; // blanks around = sign`
	`aRecord.anItem = a * (b + c);`

if, else	Conditional statement, with optional else clause:
	`if (expression)`
	`// other statements;`
	`else`
	`// statement;`
	`end`
	`if (anItem IS BLANKS) ...`
	`if (anItem NOT NUMERIC) ...`
	`if (aRecord IS ERR) ...`

while	Executes statements in a loop:
	`while (expression)`
	`// other statements;`
	`end`

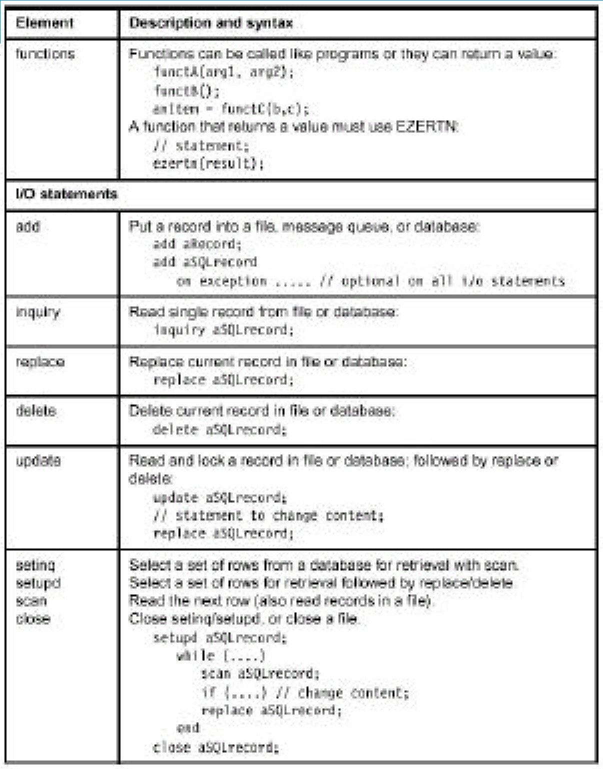
set	Initialize a record or structure or set an SQL item to null:
	`set aRecord empty; // blank (char) or zero (numeric)`
	`set sqlRecord.anItem null;`

select	Multiple sets of statements where at most one set is executed:
	`select (item or expression)`
	`case value1:`
	`// statements;`
	`case value2, value3:`
	`// statements;`
	`default:`
	`// statements;`
	`end`

call	Call another program. Arguments are passed as reference, that is the called program can change the values of the calling program:
	`call progA(arg1, arg2);`
	`on exception          // optional`
	`// statements;`
	`end`

WebSphere software

EGL elements...



Element	Description and syntax

functions	Functions can be called like programs or they can return a value:
	`functA(arg1, arg2);`
	`functB();`
	`anItem = functC(b,c);`
	A function that returns a value must use EZERTN:
	`// statement;`
	`ezertn(result);`

I/O statements	

add	Put a record into a file, message queue, or database:
	`add aRecord;`
	`add aSQLrecord`
	`on exception ..... // optional on all i/o statements`

inquiry	Read single record from file or database: `inquiry aSQLrecord;`
replace	Replace current record in file or database: `replace aSQLrecord;`
delete	Delete current record in file or database: `delete aSQLrecord;`

update	Read and lock a record in file or database; followed by replace or delete:
	`update aSQLrecord;`
	`// statement to change content;`
	`replace aSQLrecord;`

setinq
setupd
scan
close
	Select a set of rows from a database for retrieval with scan.
	Select a set of rows for retrieval followed by replace/delete.
	Read the next row (also read records in a file).
	Close setinq/setupd, or close a file.
	`setupd aSQLrecord;`
	`while (....)`
	`scan aSQLrecord;`
	`if (....) // change content;`
	`replace aSQLrecord;`
	`end`
	`close aSQLrecord;`

WebSphere software

EGL program example

```
ezefec =1;
registry.userid =logws.userid;
registry.password =logws.password;
registry.status =-1;
logws.status ="1";
if (logws.action ="inquire")
  registry-select();
  if (registry is nrf)
    logws.userid ="";
    logws.status ="0";
  end
  if (logws.password !=registry.password)
    logws.status ="0";
  end
else
   if (logws.action ="add")
     registry-add();
     if (registry is err)
        logws.status ="0";
     end
   else
    logws.status ="0";
   end
end
```

EGL Development Process

- Develop
  - Create EGL parts
  - Write EGL script
- Build
  - Generate 3GL from EGL specifications
  - Build runtime executables on target platform
- Test
  - Use integrated EGL debugger
- Run
  - zOS UNIX System Services, Windows NT, 2000, XP
    - tier 2 (Java on web application server) and tier 3 (Java)
  - CICS for MVS
    - tier 3 (COBOL)

WebSphere software

Develop

EGL Part Types

- Program
- Function

Logic

- Item
- Record
- Structure

Data

- Build Descriptor
- Linkage Options
- Resource Association
- Link Edit
- Bind Control

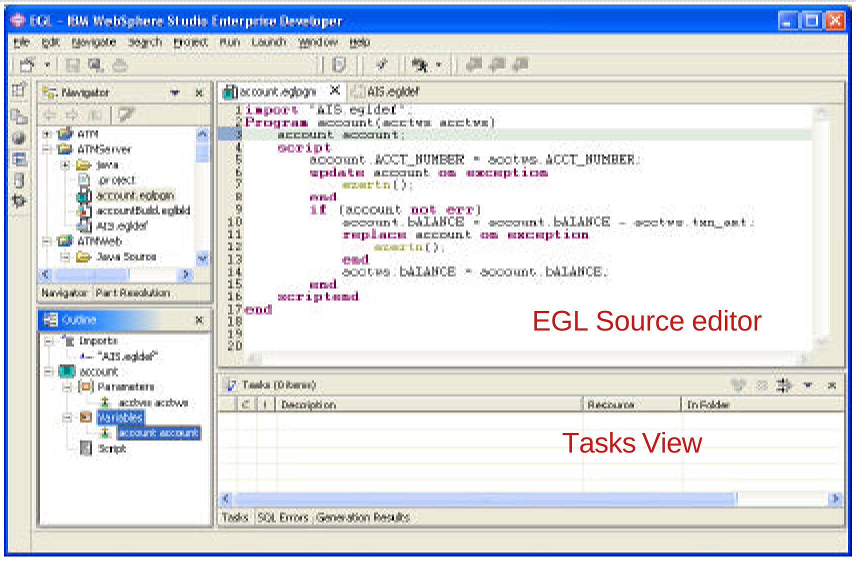
Control

EGL Parts *(continued)*

- Defined in any order
- Stored/grouped in 3 file types
  - .eglpgm - EGL program file
    - 1 program
    - and any associated parts
      - Functions, Data parts, .egldef imports
  - .egldef - EGL definitions file
    - Functions, Data parts
    - Reusable functions, shared data parts, managed parts
  - .eglbld - EGL build file
    - Control parts
    - Required for EGL generation process
- Authored with
  - EGL Parts editor: fill-in-the-blank GUI's
  - EGL Source editor - source text editor

WebSphere software

EGL Source editor and EGL Perspective

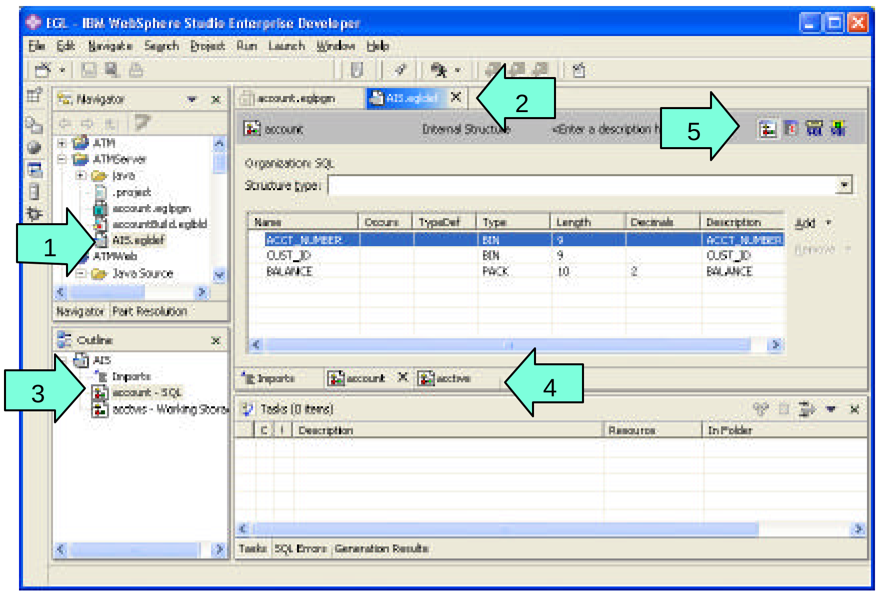


```
import "AIS.egldef";
Program account(acctvws acctvws)
    account account;
    script
        account.ACCT_NUMBER = acctvws.ACCT_NUMBER;
        update account on exception
            exertn();
        end
        if (account not err)
            account.BALANCE = account.BALANCE - acctvws.txn_amt;
            replace account on exception
                exertn();
            end
            acctvws.BALANCE = account.BALANCE;
        end
    scriptend
end
```

Benefit: Fast development for the experienced programmer

WebSphere software

Editing EGL files - EGL Parts Editor



Name	Occurs	TypeDef	Type	Length	Decimals	Description
ACCT_NUMBER			BIN	9		ACCT_NUMBER
CUST_ID			BIN	9		CUST_ID
BALANCE			PACK	10	2	BALANCE

C	!	Description	Resource	In Folder

Benefit: Easy development for the novice programmer

WebSphere software

IBM

Editing EGL files - Description

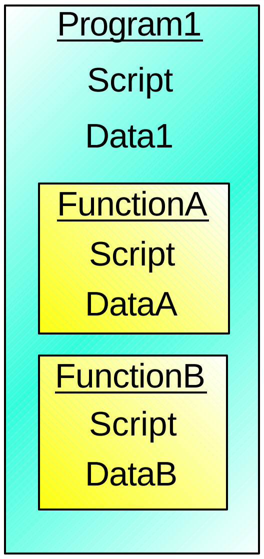
1. In the Navigator view, double click on an EGL file
   - ► File is opened with the editor last used (parts editor or source editor) and is opened in the Outline view (To change editor, right click on the EGL file, select Open with..., and select EGL Parts Editor or EGL Source Editor)
2. In the Editor, tabs at the top are for opened files
   - ► highlighted tab is active file
   - ► click on tabs to switch between files
3. Outline view shows parts defined in the opened EGL file
   - ► First part in EGL file is opened for editing when file opened
   - ► Double click on a part to open it in the Editor
4. In the Editor, bottom tabs are for opened EGL parts
   - ► tab with the "X" is the active part, click on "X" to remove part from Editor
   - ► click on tabs to switch between parts
5. Active part may have multiple property pages
   - ► Script, Signature, Variables, SQL properties, Default SQL
   - ► click on buttons to switch between pages

WebSphere software

IBM

EGL Logic parts

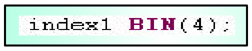
- Program
  - Main logical unit
    - Tested with EGL debugger
    - Becomes runtime executable
  - Use EGL script to define logic
  - Can receive parameters
  - Can have data areas
- Function
  - Callable logical unit
    - invoked from program or another function
    - reusable
  - Use EGL script to define logic
  - Can receive parameters, pass return value
  - Can have local data areas



WebSphere software

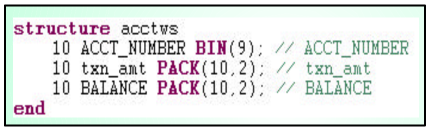
EGL Data parts

- Item
  - a memory area that cannot be subdivided
  - e.g.



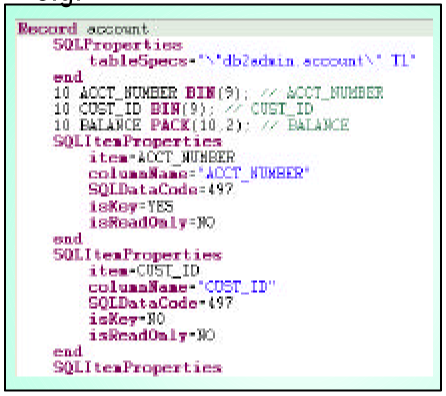
```
index1 BIN(4);
```

- Structure
  - a collection of memory areas
  - e.g.



```
structure acctws
    10 ACCT_NUMBER BIN(9); // ACCT_NUMBER
    10 txn_amt PACK(10,2); // txn_amt
    10 BALANCE PACK(10,2); // BALANCE
end
```

- Record
  - a structure with a type specified
    - Working Storage, SQL row, Message Queue, sequential file, etc.
  - e.g.



```
Record account
    SQLProperties
        tableSpecs="\"db2admin.account\" T1"
    end
    10 ACCT_NUMBER BIN(9); // ACCT_NUMBER
    10 CUST_ID BIN(9); // CUST_ID
    10 BALANCE PACK(10,2); // BALANCE
    SQLItemProperties
        item=ACCT_NUMBER
        columnName="ACCT_NUMBER"
        SQLDataCode=497
        isKey=YES
        isReadOnly=NO
    end
    SQLItemProperties
        item=CUST_ID
        columnName="CUST_ID"
        SQLDataCode=497
        isKey=NO
        isReadOnly=NO
    end
    SQLItemProperties
```

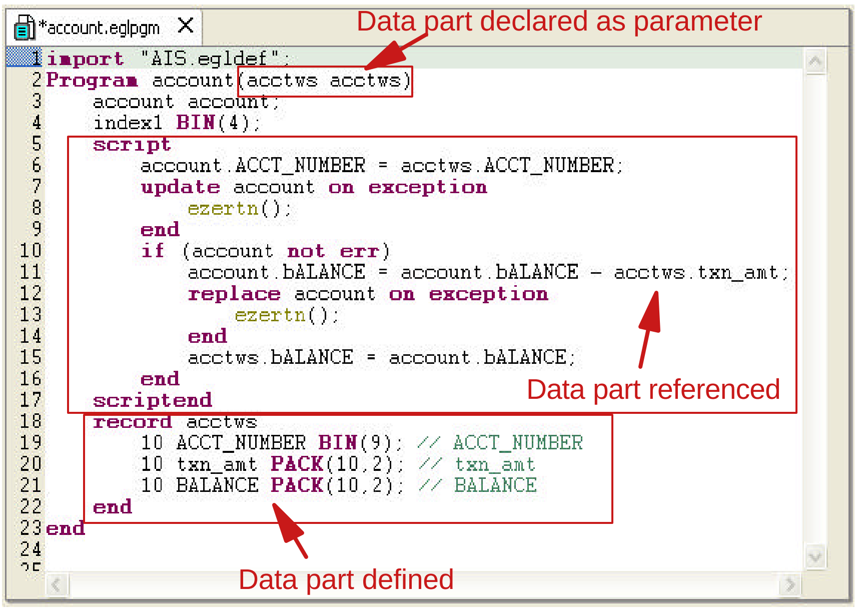
WebSphere software

Using Data parts

- Must be defined
  - ►Specify part characteristics
  - ►Within an .eglpgm or .egldef file
- Must be declared
  - ►Specify as parameter or variable within a logic part
  - ►e.g. Program account(acctws acctws) ...
    - (acctws acctws) is an example of a parameter declaration
    - (<u>acctws</u> acctws) is the name used as a qualifier in the script
    - (acctws <u>acctws</u>) refers to a data part definition
- Referenced in the script of a logic part
  - ►Examples
    - unqualified: balance = balance - txnAmt;
    - qualified: account.balance = acctws.balance
  - ►Defining or declaring a variable before scripting is not required
    - but this reduces the help that code assist can give

WebSphere software   IBM

Using Data parts example



```
import "AIS.egldef";
Program account(acctws acctws)
    account account;
    index1 BIN(4);
    script
        account.ACCT_NUMBER = acctws.ACCT_NUMBER;
        update account on exception
            ezertn();
        end
        if (account not err)
            account.bALANCE = account.bALANCE - acctws.txn_amt;
            replace account on exception
                ezertn();
            end
            acctws.bALANCE = account.bALANCE;
        end
    scriptend
    record acctws
        10 ACCT_NUMBER BIN(9); // ACCT_NUMBER
        10 txn_amt PACK(10,2); // txn_amt
        10 BALANCE PACK(10,2); // BALANCE
    end
end
```

WebSphere software   IBM

Script - Writing logic with EGL

1. Assignment
2. Keyword
3. I/O keyword
4. Function
5. EZE words

```
script
  ①account.ACCT_NUMBER = acctws.ACCT_NUMBER;
  ③update account on exception
     ④⑤ezertn();
    end
   ②if (account not err)
      ①account.bALANCE = account.bALANCE - acctws.txn_amt;
      ③replace account on exception
           ezertn();
        end
      ①acctws.bALANCE = account.bALANCE;
    end
scriptend
```

WebSphere software

IBM

Using Content Assist

- View and paste statement templates into the script
  - ►shows list of EGL statements
    - –variations in syntax also shown
- View and paste declared data parts into the script
  - ►Ctrl + Space, select declared record
  - ►type . (decimal point), then Ctrl + Space to get list of data items
- View and paste EZE words
  - ►Ctrl + Space, then *
- Content Assist activated by:
  - ►Context menu item, Content Assist
  - ►Ctrl + Space
- List shown will be narrowed to typed characters that match

Benefit: Less keystrokes = Faster development

WebSphere software

IBM

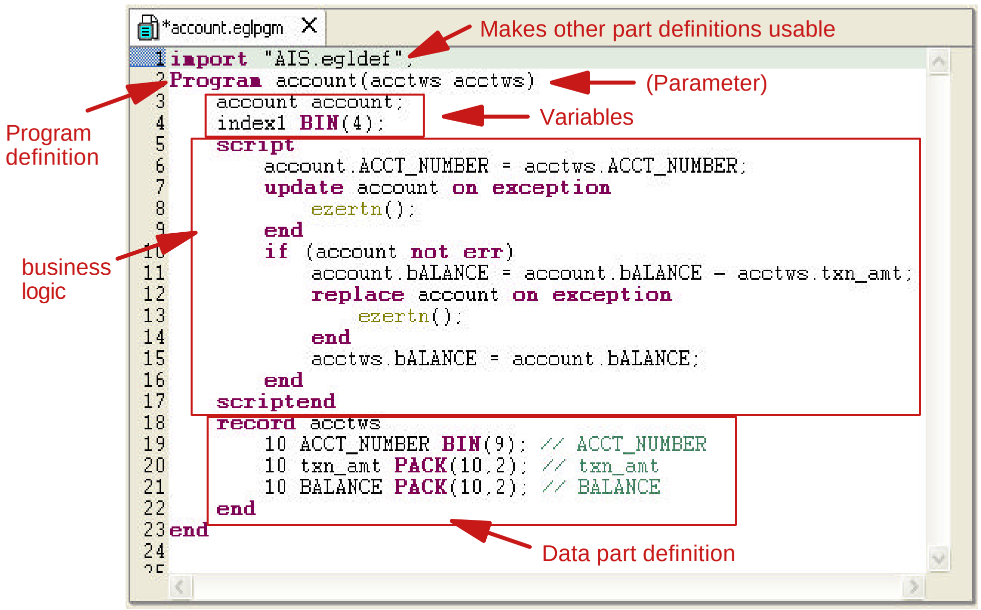
Writing SQL requests

- Access a table, a view, or a join of tables or views
  - UDB directly from CICS
  - all others via JDBC
- Begins with I/O keyword, may be terminated with "end"
  - add, inquiry, update, delete, replace, setinq, scan, sqlexec
  - update account on exception ezertn(); end
- Identify a record definition for SQL row
  - update account on exception ezertn(); end
- Identify SQL clause
  - optional, if not present, then default SQL clause is used
  - in most cases you can modify the default SQL
  - update account statementID=account-update on exception ...
- Identify what to do if an I/O error occurs
  - update account on exception ezertn(); end

WebSphere software

IBM

Program definition summary



```
import "AIS.egldef";
Program account(acctws acctws)
   account account;
   index1 BIN(4);
   script
      account.ACCT_NUMBER = acctws.ACCT_NUMBER;
      update account on exception
         ezertn();
      end
      if (account not err)
         account.bALANCE = account.bALANCE - acctws.txn_amt;
         replace account on exception
            ezertn();
         end
         acctws.bALANCE = account.bALANCE;
      end
   scriptend
   record acctws
      10 ACCT_NUMBER BIN(9); // ACCT_NUMBER
      10 txn_amt PACK(10,2); // txn_amt
      10 BALANCE PACK(10,2); // BALANCE
   end
end
```

WebSphere software

IBM

WebSphere software

More details

Script syntax rules

- Statement terminated with a semicolon
  - optional for "end" statement
- Statement can continue onto multiple lines
- Multiple statements per line allowed
- Comments
  - single line: text between // and EOL character
  - multi-line: text between /* and */
- Block-containing statements are terminated with end delimiter
  - IF, SELECT, WHILE, ...
- Brackets [ ] used for subscripts
- Parentheses ( ) used for grouping, as in a math expression
- Names in statements and throughout EGL are case-insensitive

WebSphere software

Statement Types

- Assignment
- Keyword
- Function
- EZE words

WebSphere software

Assignment statement

- Form: target = source;
- Target: data item, data item eze word, declared record or structure
- Source:
  - ►If target is data item or data item eze word, the source must be numeric or string expression
    - –complex series of symbols: z = a + b + c;
    - –data item or data item eze word: myDate = ezedte;
    - –function invocation: myItem = readFile(myKeyValue);
    - –literal: ezeuserid = "USER";
  - ►If the target is a structure, the source must be a structure
  - ►If the target is a record, the source must be a record
    - –myRecord01 = myRecord02

WebSphere software

Assignment statement examples

- Copy data from one area to another
  - registry.password = logws.password;
  - registry.userid = logws.userid;
- Places a value into a data area
  - The result of an arithmetic calculation
    - myDataItem = bigValue - 32;
  - A value returned from a function invocation
    - myItem = readFile(myKeyValue);
  - A literal
    - ezefec = 1;
    - registry.status = -1;
    - logws.status = "1";

WebSphere software IBM

Keyword Statements

call	transfers control to another program and optionally passes a series of values. Control returns to the caller when the called program ends. If the called program changes the passed data, the storage area available to the caller is changed,too.
if, else	if marks the start of a set of statements that run only if a logical expression resolves to true. else marks the start of an alternative set of statements that run only if the logical expression resolves to false.
select	select marks the start of multiple sets of statements, where at most only one of those sets is run.
set	Initializes the value of each structure item in a record or sets a structure item in a SQL row record to null
while	marks the start of a set of statements that run in a loop. The first run occurs only if a logical expression resolves to true, and each subsequent iteration depends on the same test.

WebSphere software IBM

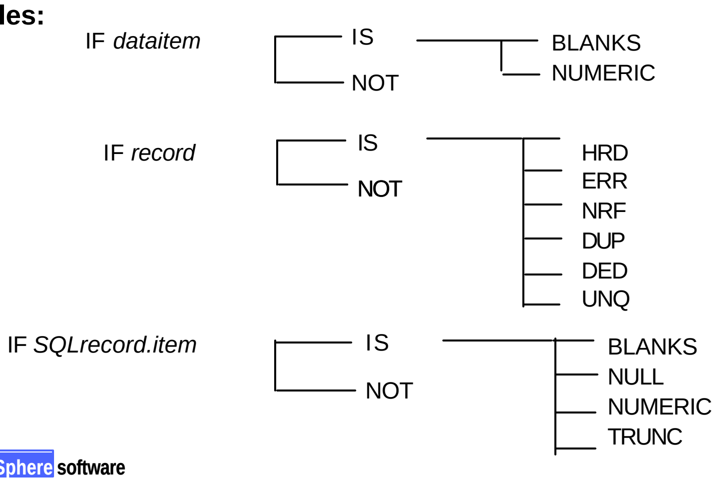
I/O Keyword Statements

add	places a record in a file, message queue, or database
close	detaches the file or message queue associated with a given record; or, in the case of a SQL record, releases the unprocessed rows that were selected by an update, setupd, or setinq statement.
delete	removes either a record from a file or a row from a database.
inquiry	reads either a single record from a file or a single row from a database.
replace	replace puts a changed record into a file or database.
scan	reads the next record from a file, message queue, or database.
scanback	scanback reads the previous record in the file that is associated with a specified EGL indexed record.
set	Establishes position in the file associated with an indexed record
setinq	selects a set of rows from a relational database for later retrieval with scan statements.
setupd	selects a set of rows from a relational database for later retrieval with scan statements; in this case, each scan locks a row for subsequent replacement or deletion.
sqlexec	lets you write an SQL data-definition statement (of type CREATE TABLE, for example), as well as data-manipulation statements of type DELETE, INSERT, or UPDATE but not others.
update	reads and locks a record/row from a file or in a relational database. Update is followed by a replace or delete against the same record.

Boolean operators - IS / NOT

- IS - tests true if the specified state is true
- NOT - that tests true if the specified state is false

Examples:



WebSphere software

Using Multiple EGL Files

- .egldef files enable part sharing
  - DB definitions, reusable functions, etc
  - less maintenance
  - control/ownership can be by another
- IMPORT statement
  - links to the named EGL definition file
- AccountRec example
  - Scenario 1: defined twice, maintain both
  - Scenario 2: defined once, definition shared

- Scenario 1

DetailPgm.eglpgm	ListPgm.eglpgm
ReadAccountFn	ListEmployeeFn
AccountRec	AccountRec

- Scenario 2

  DetailPgm.eglpgm
  - Import AIS.egldef
  - ReadAccountFn

  ListPgm.eglpgm
  - Import AIS.egldef
  - ListAccountFn

  AIS.egldef
  - AccountRec

WebSphere software

IBM

Typedef Example

- If this structure definition exists

```
Structure address
   10 streetAddress CHA(20);
   10 city CHA(15);
end
```

- Then address could be used as a typedef in a new structure

```
Structure personnel
   10 homeAddress address;
   10 workAddress  address;
end
```

- This is equivalent to defining personnel as

```
Structure personnel
   10 homeAddress;
      20 streetAddress CHA(20);
      20 city CHA(15);
   10 workAddress;
      20 streetAddress CHA(20);
      20 city CHA(15);
end
```

WebSphere software

IBM

Language Constructs

WebSphere software

IBM

Function Invocation statements

- Directs processing to:
  - another script
  - function eze word
- Invoke with no arguments
  - callMax();
  - EZERTN();
- Invoke with arguments
  - CallMax(num1,num2);
- Invoke inline as value
  - LargerNum = callMax(num1, num2);

WebSphere software

IBM

EZE Words

- Used in logic
  - ezefec = 1;
  - ezertn();
- Provide access to many system-provided values
  - date and time
  - runtime environment information
- Provide useful functions
  - such as mathematical and string operations

WebSphere software IBM

EZE Words (continued)

EZEFEC

- **Controls continuation after hard I/O errors**
- **If EZEFEC is set to 1 and an error routine is specified**
  - **hard I/O error is bypassed and processing continues**
  - **Application is responsible for reporting error to application user**
- **Example: EZEFEC = 1;   /* Should be in every program**

EZESYS

- **Identifies environment in which program is running**
- **Not available in a Java wrapper**
  - **Example:**
```
IF (EZESYS IS "MVSCICS")
my-vsam-fnc();            /* Peform VSAM function */
END;
```

EZEUSRID

- **Contains the user ID that is currently logged on**
  - **Example: AUDIT-LOG.USERID = EZEUSRID;**

WebSphere software IBM

EZE Words (continued)

EZECOMIT

- function that calls services to save recoverable file, database, and message queue updates since the last commit
- The scan position is lost and update locks are released for any files or databases affected by the EZECOMIT
- An exception to this occurs when using Declare Cursor With Hold
  - Example: EZECOMIT();

EZEROLLB

- function that calls system services to back out recoverable file, database, and message queue updates since the last commit point
- rollback occurs automatically if the program ends with an unexpected error
  - Example: EZEROLLB();

WebSphere software IBM

String Handling

EZESBLKT	Changes null terminator and any subsequent characters in a string to blanks
EZESCCWS	Concatenates one string to another, with a separator string between them
EZESCMPR	Compares one substring to another
EZESCNCT	Concatenates one string to another
EZESCOPY	Copies one substring to another
EZESFIND	Finds the first occurrence of a string within a string
EZESNULT	Changes trailing blanks to nulls in a string
EZESSET	Sets each character in a substring to the same character value
EZESTLEN	Returns length of an item less trailing blanks and nulls
EZESTOKN	Finds next token in string and copies it to an item

WebSphere software IBM

Math Routines - General

EZEABS	Absolute value
EZECEIL	Smallest integer not less than the numericDataItem
EZEEXP	Exponential value (*e* raised to power of numericDataItem)
EZEFLOOR	Largest integer not greater than numericDataItem
EZEFREXP	Split numeric data item into normalized fraction in range of 1/2 to 1 and a power of 2
EZELDEXP	Product of numericDataItem multiplied by 2 to the power of integer
EZELOG	Natural logarithm
EZELOG10	Base 10 logarithm
EZEMAX	Maximum
EZEMIN	Minimum
EZEMODF	Split into integral and fractional parts
EZENCMPR	Numeric comparison
EZEPOW	Raise to power
EZEPRCSN	Maximum precision in decimal digits
EZEROUND	Round to integer power of 10
EZESQRT	Square root

WebSphere software

IBM

Math Routines - Floating Point

EZEFLADD	Floating point add
EZEFLDIV	Floating point division
EZEFLMO	Floating point remainder of division
EZEFLMUL	Floating point multiplication
EZEFLSET	Convert to floating point
EZEFLSUB	Floating point subtraction

WebSphere software

IBM

Math Routines - Trigonometric Functions

EZEACOS	Arccosine
EZEASIN	Arcsine
EZEATAN	Arctangent
EZEATAN2	Theta component of the polar coordinate corresponding to the rectangular coordinate
EZECOS	Cosine
EZECOSH	Hyperbolic cosine
EZESIN	Sine
EZESINH	Hyperbolic sine
EZETAN	Tangent
EZETANH	Hyperbolic tangent

WebSphere software

IBM

Arithmetic expressions

Unary operators

+	The operand is used without changing its sign
-	The value of the operand is negated

Binary arithmetic operators

+	Operands are added
-	The second operand is subtracted from the first operand
*	Operands are multiplied
/	The first operand is divided by the second operand
//	The result is the remainder of dividing the first operand by the second operand

Examples:

```
PERCENT-CHANGE = (NEW-VALUE - OLD-VALUE) * 100 / OLD-VALUE;
OP1 = OP2 + OP3 * OP4;
OP1 = OP2 * (OP3 + OP4);
OP1 = -OP2 + OP3;
OP1 = OP2 + OP3[INDEX];
```

WebSphere software

IBM

Control Parts

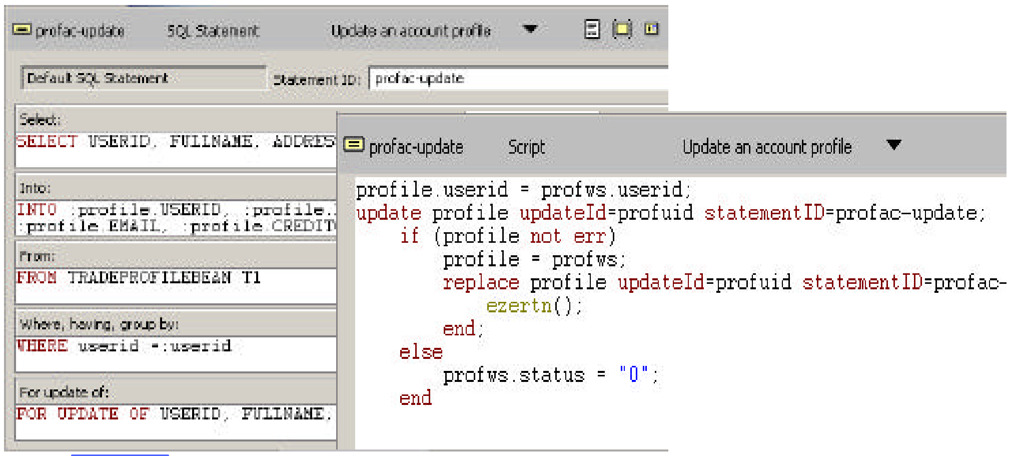
- Build descriptor
  - ►controls the generation process
  - ►used for generation
- Linkage options
  - ►describes how to implement program calls or access remote files
  - ►used for test, generation, and execution
- Resource associations
  - ►links EGL record to a file or message queue to be accessed
  - ►used for test, generation, and execution
- Bind control
  - ►for z/OS, DB2 bind control parameters
  - ►specified at generation time and used in executable preparation
- Link edit
  - ►for z/OS, describes how to form a load module from two or more programs
  - ►specified at generation time and used in executable preparation

WebSphere software

IBM

Update Function - SQL Statement

- Updating a row is a two step process
  - ►First, perform an UPDATE
  - ►An UPDATE selects and locks the rows involved, thus determining data availability and preventing other users from updating the same data



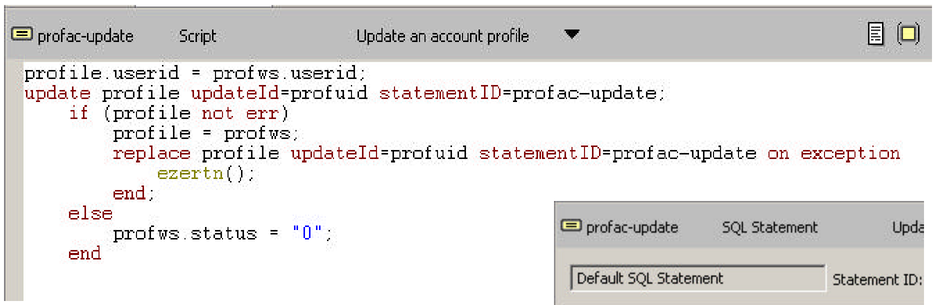
```
SELECT USERID, FULLNAME, ADDRES
INTO :profile.USERID, :profile.
:profile.EMAIL, :profile.CREDIT
FROM TRADEPROFILEBEAN T1
WHERE userid =:userid
FOR UPDATE OF USERID, FULLNAME,
```

```
profile.userid = profws.userid;
update profile updateId=profuid statementID=profac-update;
    if (profile not err)
        profile = profws;
        replace profile updateId=profuid statementID=profac-
            ezertn();
        end;
    else
        profws.status = "0";
    end
```

WebSphere software

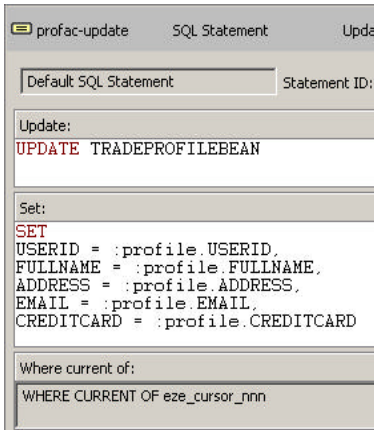
IBM

Replace Function - SQL



profac-update   Script   Update an account profile

```
profile.userid = profws.userid;
update profile updateId=profuid statementID=profac-update;
    if (profile not err)
        profile = profws;
        replace profile updateId=profuid statementID=profac-update on exception
            ezertn();
        end;
    else
        profws.status = "0";
    end
```



profac-update   SQL Statement   Upda

Default SQL Statement   Statement ID:

Update:

```
UPDATE TRADEPROFILEBEAN
```

Set:

```
SET
USERID = :profile.USERID,
FULLNAME = :profile.FULLNAME,
ADDRESS = :profile.ADDRESS,
EMAIL = :profile.EMAIL,
CREDITCARD = :profile.CREDITCARD
```

Where current of:

```
WHERE CURRENT OF eze_cursor_nnn
```

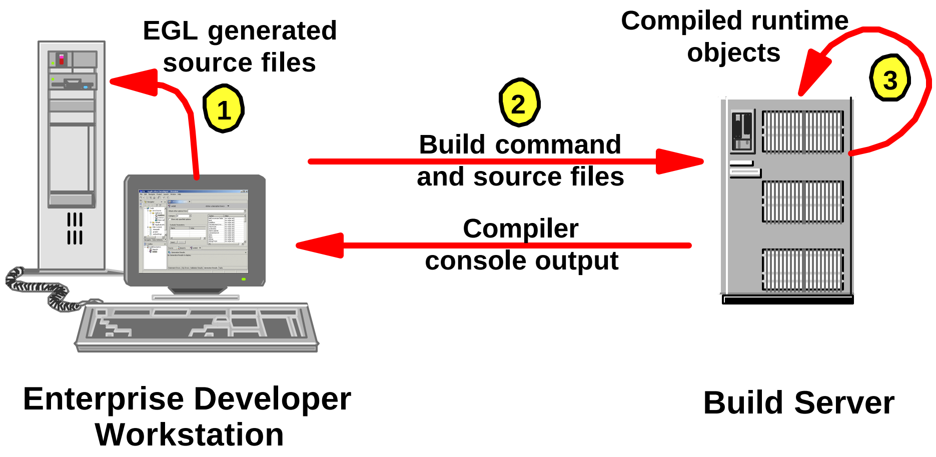
- The second step is REPLACE
  - Replace performs the actual update of the row(s) selected by the UPDATE function
  - New values must be moved into the PROFILE record for any data items that are to be changed
  - CURRENT DATE or CURRENT TIMESTAMP may be used to indicate when a row was last updated

WebSphere software   IBM

WebSphere software   IBM

Build

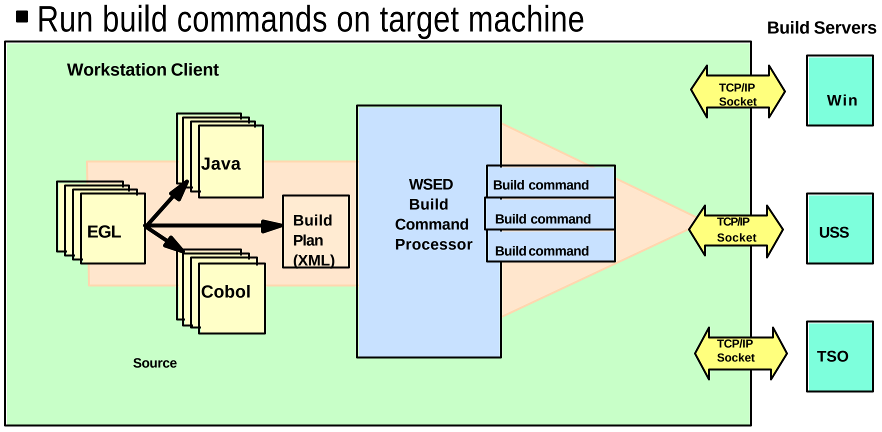
Generating source code from EGL


EGL generated source files
1
Compiled runtime objects
3
2
Build command and source files
Compiler console output
Enterprise Developer Workstation
Build Server

WebSphere software

Enterprise Developer Build Process

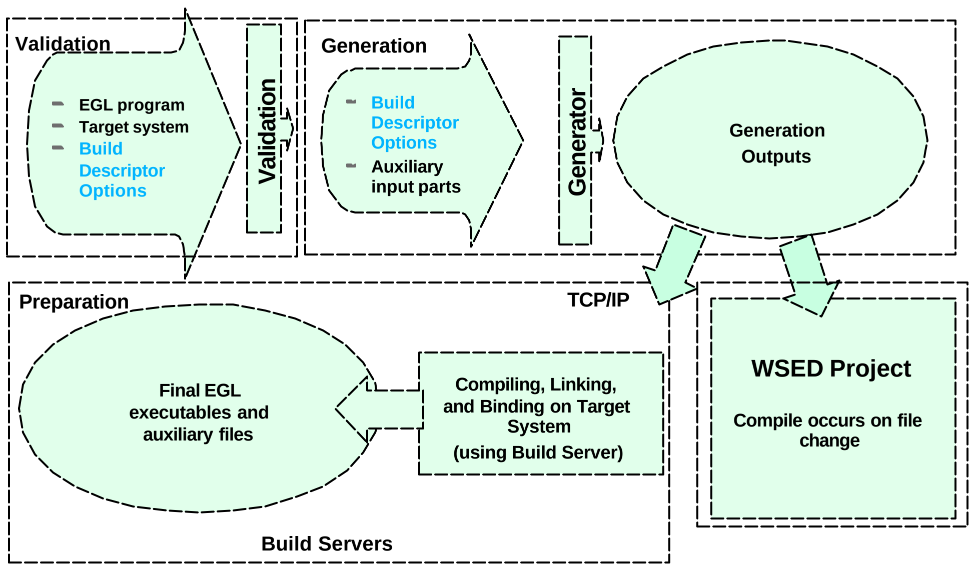
- Automated build based on build plans (XML)
- Automatic transfer to target machine
- Run build commands on target machine



Benefit: Developer spends less time in the build process

WebSphere software

EGL Build Process

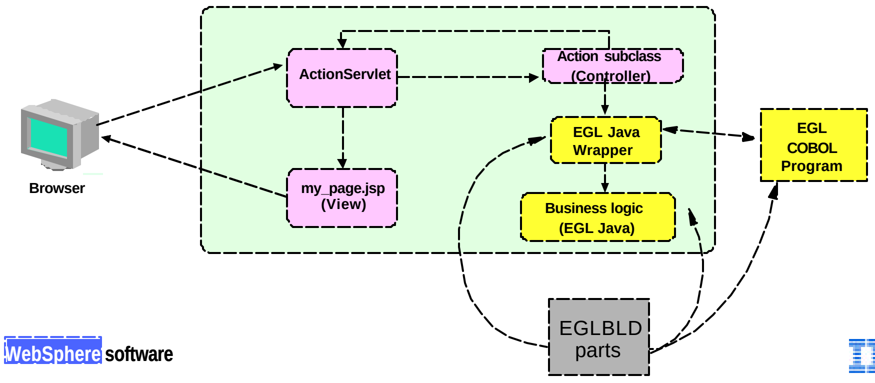


WebSphere software

IBM

EGL Generation

- Create 3GL source code
  - Java or COBOL from EGL program specification
- Create Java wrappers
  - Java Wrapper makes access to generated EGL program easier
  - EJB Session Beans can be generated as well
- EGL Build Descriptor parts are required for each

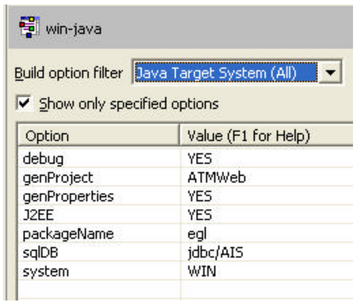


WebSphere software

IBM

Build Descriptor for Java Generation

- Generate EGL program
  - ►as Java code
  - ►for WebSphere Test Environment



win-java

Build option filter: Java Target System (All)

☑ Show only specified options

Option	Value (F1 for Help)
debug	YES
genProject	ATMWeb
genProperties	YES
J2EE	YES
packageName	egl
sqlDB	jdbc/AIS
system	WIN

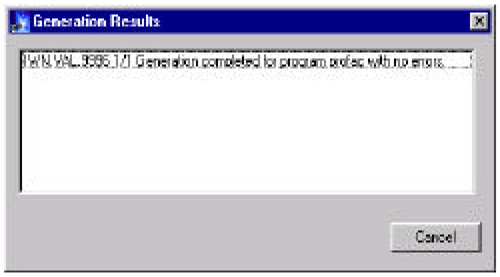
- debug
  - ►if yes, runtime debugging enabled
- genProject
  - ►project to store generated outputs
- genProperties
  - ►create properties variables
  - ►placed in deployment descriptor
- J2EE
  - ►execute in J2EE runtime
- packageName
  - ►folder(s) to store Java source and classes
- sqlDB
  - ►DB name for genProperties
- system
  - ►Target generation platform
  - ►Win = Java source for Windows

WebSphere software IBM

Generation Results

- Generation Results window will appear
  - ►With Validation Messages
  - ►Successful generation

- Validation messages are cross field validation and need to be corrected for generation to complete successfully.



Generation Results

```
IWN.VAL.9996.1/1 Generation completed for program prefec with no errors.
```

WebSphere software IBM

Generation Outputs for Java Generation

Source Part	Output Type	Generated Name	Example
Program	Server Program	partname.cbl partname.java	LOGAC.CBL LOGAC.java
Records/Structures in Java Server Programs	Java Code	**Eze**partname.java	Ezelogws.java Ezeregistry.java
Records/Structures used parameters in functions	Java Code	**Eze$param**functionname parametername.java	Eze$paramFuncRec.java
Functions used in Java program	Methods in server program	**$func**functionname	$funclogin
Program	Debug Control XML	partname_**debug.xml**	LOGAC_debug.xml
Program	Build Plan XML (for remote build)	partname**BuildPlan**.xml	LOGACBuildPlan.xml
Program	Text file for runtime properties	partname-env.txt	LOGAC-env.txt

Note: '-' will be changed to 'x002D' in java names (no quotes)

Generation Outputs for Java Wrappers

Source Part	Output Type	Generated Name	Example
Program	Java Bean	partname**Wrapper**.java	LOGACWrapper.java
Records used as parameters in programs	Java Bean	partname.java or typedefname.java	logws.java

Note: '-' will be changed to 'x002D' in java names (no quotes)

WebSphere software

Test

Test Scenarios

- Testing an EGL program alone
  - e.g.called EGL server program
    - or batch EGL program
  - runs in JVM, no J2EE web container required
    - More test turn arounds
      - environment starts faster, runs faster
  - Source level debugging with EGL debugger

Benefit: Faster iterative development and test

- Testing a complete web application
  - e.g. web client calls EGL server program
  - runs in J2EE container
    - e.g. WebSpere Test Enviroment
  - Source level debugging for
    - JSP, Java, EGL

Benefit: End-to-end test and debug from the Workbench

Testing an EGL program alone

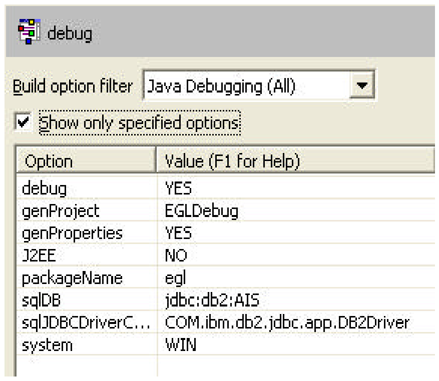
- One time setup
  - Create a debug Build Descriptor
  - Set breakpoint(s) in the program
    - at least one at the beginning of the program script to initialize variables
  - Define a Launch configuration
- Repeat for each test
  - Generate the program's EGL into Java
    - everytime the program or one of its parts has changed
  - Start the Launch configuration
  - Initialize variables
    - e.g. parameters for called EGL program
    - alternative: create a test client EGL program
  - Debug program

WebSphere software

IBM

Build Descriptor for non-J2EE Test

- key options
  - debug = YES
  - J2EE = NO


debug

Build option filter: Java Debugging (All)

☑ Show only specified options

Option	Value (F1 for Help)
debug	YES
genProject	EGLDebug
genProperties	YES
J2EE	NO
packageName	egl
sqlDB	jdbc:db2:AIS
sqlJDBCDriverC...	COM.ibm.db2.jdbc.app.DB2Driver
system	WIN

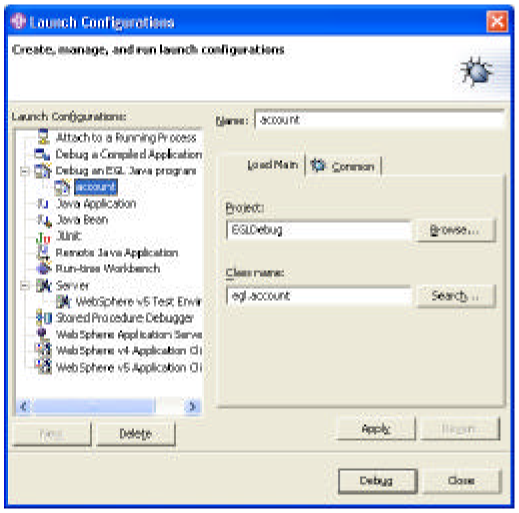
- debug
  - = YES, creates debug mapper
- genProject
  - project to store generated outputs
- genProperties
  - create properties variables
  - placed in *program*.properties file
- J2EE
  - =NO, create Java for non-J2EE
- packageName
  - folder(s) to store generated Java
- sqlDB
  - DB name for genProperties
- sqlJDBCDriver
  - driver for genProperties
- system
  - Target generation platform
  - Win = Java source for Windows

WebSphere software

IBM

Create a launch configuration

- Defines a context in which to run your code
- Define once and reuse


Launch Configurations
Create, manage, and run launch configurations
Launch Configurations:
Attach to a Running Process
Debug a Compiled Application
Debug an EGL Java program
account
Java Application
Java Bean
JUnit
Remote Java Application
Run-time Workbench
Server
WebSphere v5 Test Envi
Stored Procedure Debugger
WebSphere Application Serve
WebSphere v4 Application Cl
WebSphere v5 Application Cl
Load Main
Common
Browse...
Search...
New
Delete
Apply
Revert
Debug
Close

Name: account

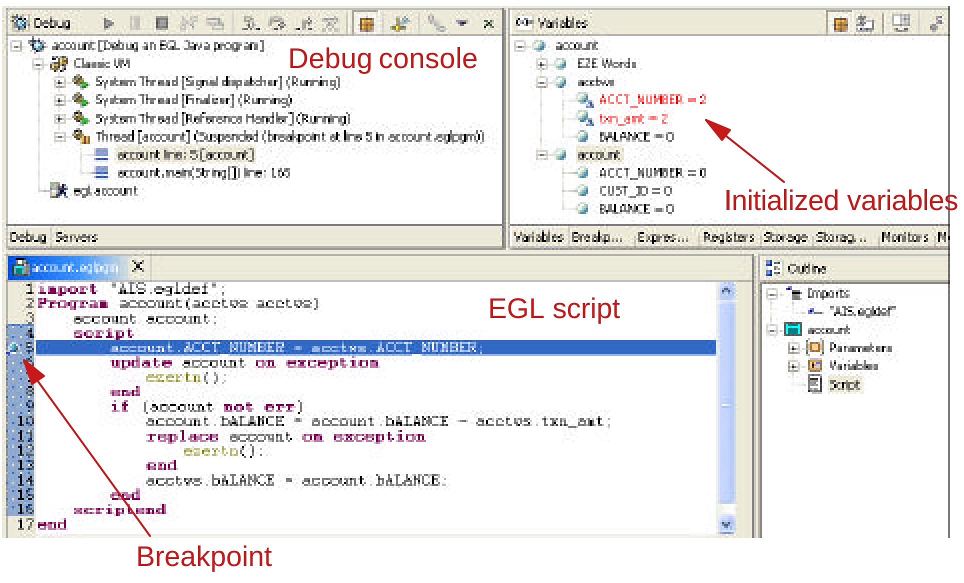
Project: EGLDebug

Class name: egl.account

WebSphere software

IBM

EGL Debugger



```
account [Debug an EGL Java program]
  Classic VM
    System Thread [Signal dispatcher] (Running)
    System Thread [Finalizer] (Running)
    System Thread [Reference Handler] (Running)
    Thread [account] (Suspended (breakpoint at line 5 in account.eglpgm))
      account line: 5 [account]
      account.main(String[]) line: 165
  egl.account
```

```
account
  EZE Words
  acctvs
    ACCT_NUMBER = 2
    txn_amt = 2
    BALANCE = 0
  account
    ACCT_NUMBER = 0
    CUST_ID = 0
    BALANCE = 0
```

```
import "AIS.egldef";
Program account(acctvs acctvs)
    account account;
    script
        account.ACCT_NUMBER = acctvs.ACCT_NUMBER;
        update account on exception
            exertn();
        end
        if (account not err)
            account.bALANCE = account.bALANCE - acctvs.txn_amt;
            replace account on exception
                exertn();
            end
            acctvs.bALANCE = account.bALANCE;
        end
    scriptend
end
```

WebSphere software

IBM

Testing a complete web application

- e.g. Using WebSphere Test Environment
- One time setup
  - ►Create Build Descriptors for server and wrapper generation
  - ►Set breakpoint(s) in the program
    - –at least one at the beginning of the program script to initialize variables
  - ►Create Server and Server Configuration
    - –Define Data Source in Server Configuration
- Repeat for each test
  - ►Generate the program's EGL into Java
    - –everytime the program or one of its parts has changed
    - –use server and wrapper Build Descriptors
  - ►Start Web project on Server
    - –right click on web project name and select
      - "Run on Server"
      - "Debug on Server"

WebSphere software

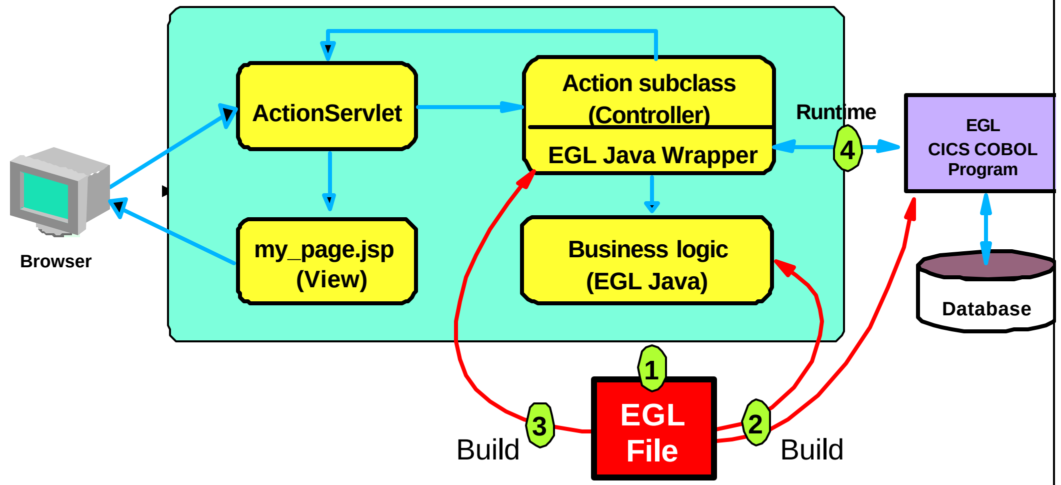
IBM

Moving to Websphere

WebSphere software

IBM

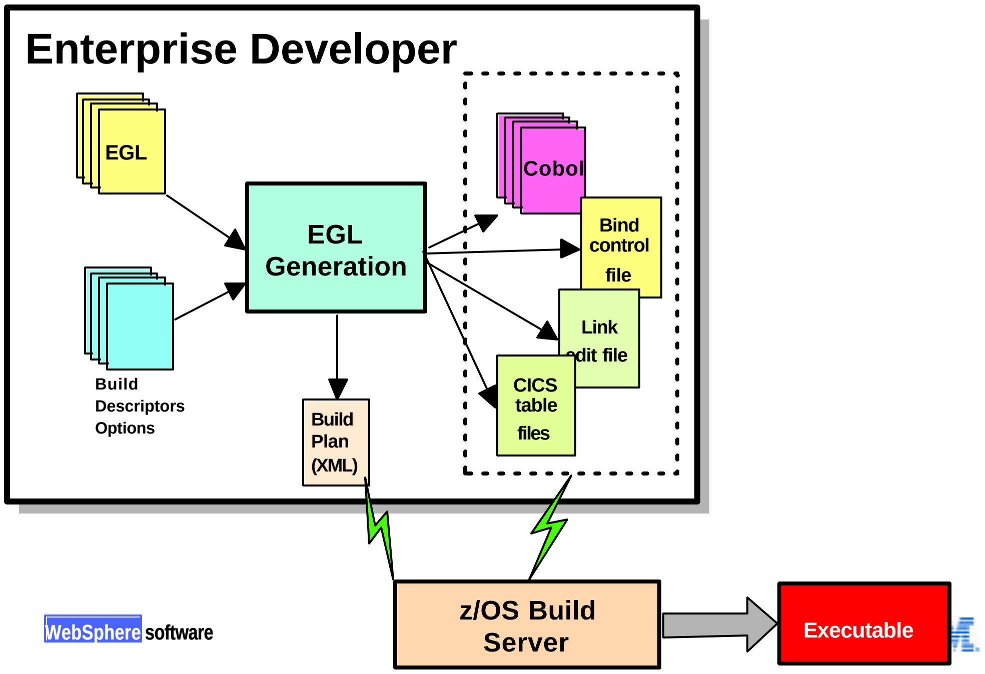
Creating and generating EGL programs



WebSphere software

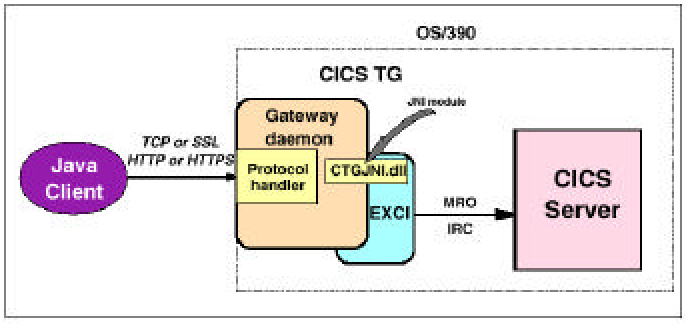
IBM

COBOL generation and deployment architecture



WebSphere software

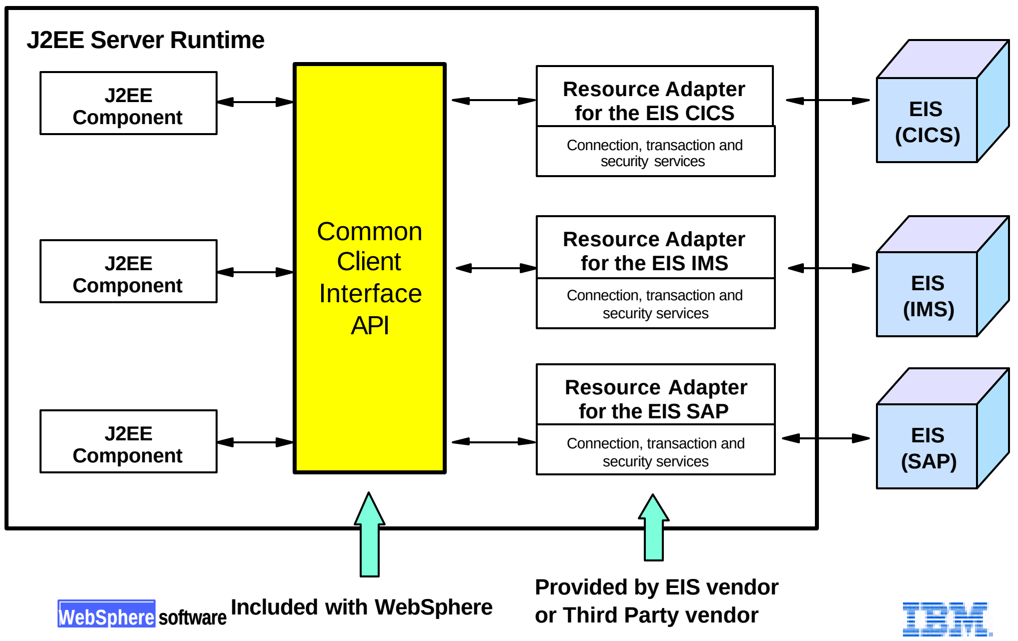
CICS Transaction Gateway



WebSphere software

IBM

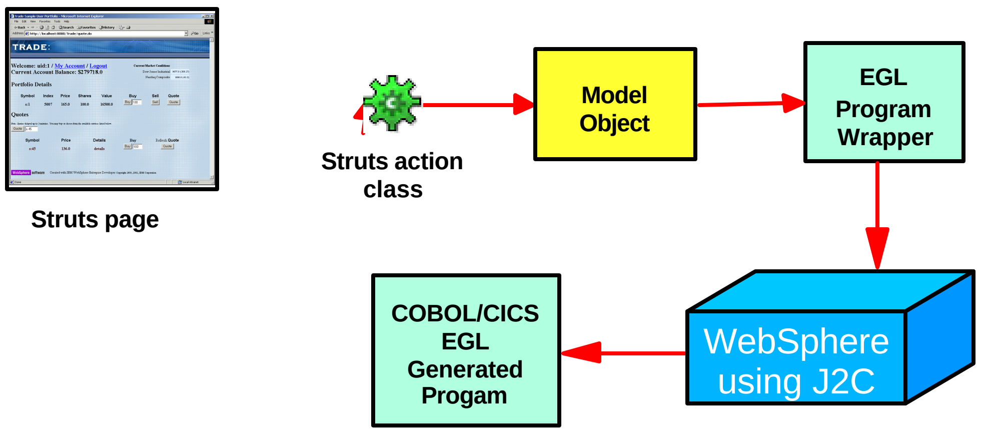
J2EE connector architecture (J2C or J2CA)


J2EE Server Runtime
J2EE Component
J2EE Component
J2EE Component
Common Client Interface API
Resource Adapter for the EIS CICS
Connection, transaction and security services
Resource Adapter for the EIS IMS
Connection, transaction and security services
Resource Adapter for the EIS SAP
Connection, transaction and security services
EIS (CICS)
EIS (IMS)
EIS (SAP)
Included with WebSphere
Provided by EIS vendor or Third Party vendor

WebSphere software

IBM

Using the Java program wrapper to COBOL



WebSphere software

IBM

e business software

WebSphere Studio Enterprise Developer 5.0 EA

POT - Lab Introduction

Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

Lab #1 - EGL Lab

- Use EGL to build a server program
  - this is a back end of the ATM Withdrawal web application
  - the server reduces the balance for a specified account by a withdrawal amount
  - the server returns the new account balance to the caller

WebSphere software

IBM

Lab #1 - Tips

- Make sure you use a different workspace from the one you used for WebSphere Studio Application Developer
- Working with Tables
  - EGL frequently makes use of tables in its wizards and editors.
    - e.g. EGL Build Descriptors in the EGL Parts Editor
  - Tabbing out of a field in a table will ensure that the value is not "lost".
  - Tabbing from cell to cell in a table is often faster then clicking on the cell to edit (see next comment).
  - Editing a cell in the table requires 2 "slow" clicks - the first to select the cell, the second select the contents of the cell for editing. If the cell has a drop down button, then a third click will show the drop down list.

Lab #2 - Struts Tools

- Design and build the front end of the ATM Withdrawal web application using Struts tools
  - a JSP accepts an account # and withdrawal amount from the user
  - calls an EGL server from a Struts action
  - a JSP displays the new account balance.

Lab #2 - Tips

- **Using Page designer**
  - If Jsp editor behaves unusually, close and reopen. This is particularly true, when a Jsp is first created.
- **Tables**
  - the tips in Lab #1 apply in this lab as well

Lab #3 - z/OS IDE

- **Part I - Setting up DEMOMVS system**
  - DEMOMVS userid obtained
  - Create partitioned datasets (PDS) to be used in Part II
  - Primer for navigating around TSO/ISPF
- **Part II - Using z/OS IDE to work with host artifacts**
  - Defining and connecting to host systems
  - Working with host artifacts thru the MVS project

Lab #3 - Tips

- Due to network latency, datasets you've added to your MVS project may "disappear". To resolve this, do a connect again to the system from the z/OS Systems view.

Lab #4 - XML Enablement

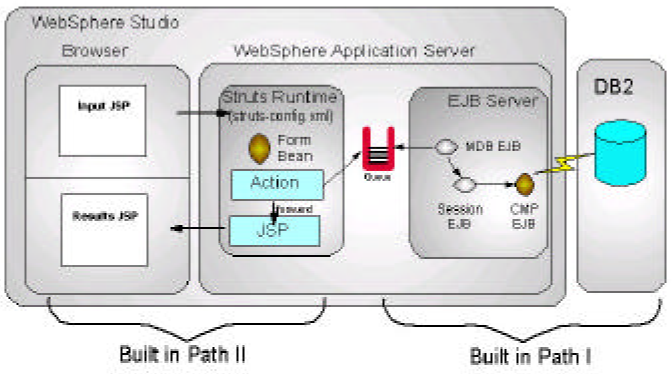
- Use the XML enablement wizard to create
  - ►Inbound converter program
  - ►Outbound converter program
  - ►Template driver program

Lab #4 - Tips

- **Make sure you're working from a SIMPLE project.**
  - ►MVS projects cannot be the source/target of the XML enablement wizard.

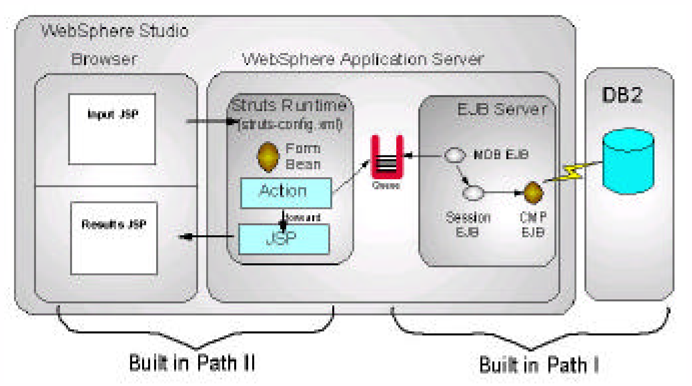
Lab #5 - Optional -Working with J2EE and STRUTS (Part II)

- **Could apply for any version of WSAD**
- **Requires some J2EE and Java Skill.**
- **Optional..**
  - ►This Lab could have small bugs it will be updated in near future.



Lab #6 - Optional -Working with J2EE and EJB (Part I)

- Could apply for WSAD
- Its used in Lab 5
- Requires some J2EE and Java Skill.
- Optional..
  - This Lab could have small bugs it will be updated in near future.



WebSphere software

Struts-based Web Applications w/Studio Enterprise Developer

WebSphere Studio Enterprise Developer V5.0

- Struts Tools
  - Set of Wizards, editors, and validation support
  - for the design and construction of Struts-based J2EE web applications
- Enterprise Generation Language (EGL)
  - Simple, high level programming specifications
  - for creating full-function COBOL and Java applications
- z/OS Application Development Tools
  - Interactive, workstation-based development
  - for mainframe COBOL, PL/I, ASM applications
- XML Enablement Enhancements for z/OS applications
  - Set of wizards to create XML transformation code
  - and web services for XML-enabled z/OS applications

Objectives and Agenda

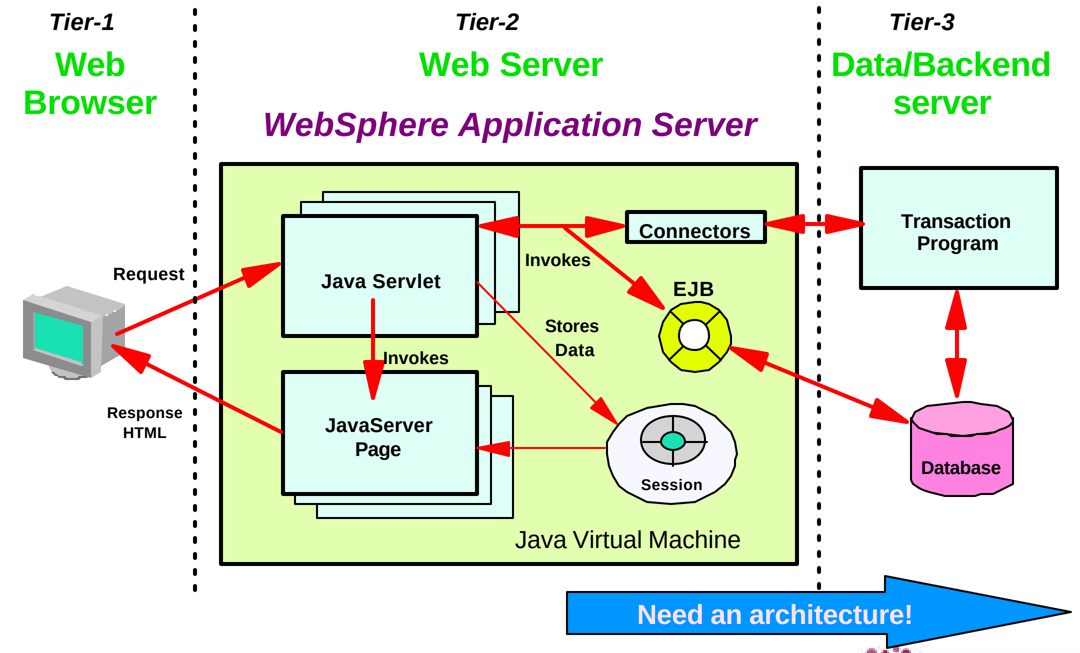
In this session we learn

- **Struts Overview**
  - **Model-View-Controller 2**
  - **What is Struts?**
  - **Struts application and components**
- **Struts Example**
  - **Small example**
- **Struts in Application Developer**
  - **Implementing Struts in Application Developer**
- **Struts in Enterprise Developer**
  - **Struts support in Enterprise Developer**
  - **Wizards and graphical design tool**
  - **Enterprise Generation Language**

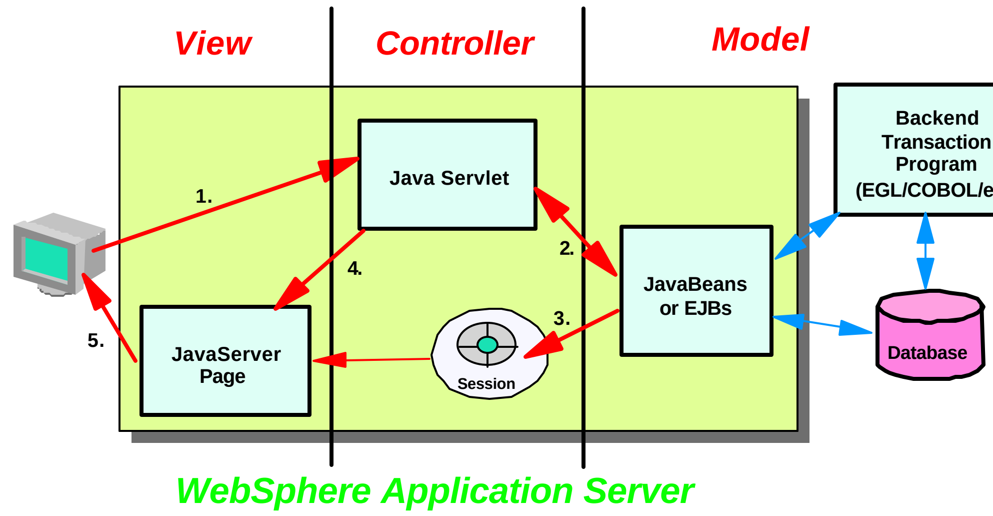
Struts Overview

WebSphere Technical Exchange

Anatomy of a Thin Client Web Application



Model-View-Controller Architecture 2



- Separates component responsibilities
- Exploits strengths of each component ("best practice")
- Promotes reuse

Implementation of the MVC Architecture

Controller "glue" code rewritten many times

- Receiving parameters from HTML form
- Validating form fields/setting error messages
- Control flow/navigation logic
- Saving state in session

Hand coding of JSPs repetitive and time consuming

- Creating dynamic elements (form fields, etc)
- Common JavaScript (e.g. setting focus)

Model logic

- Usually requires Java skills
- Population of beans for JSP elements repetitive

==> Something needed to help with these issues !

What is Struts?

A framework for building well-structured JSP and servlet based Web applications

- Supports/encourages MVC
- Uses concept of "action" classes

Includes facilities to simplify:

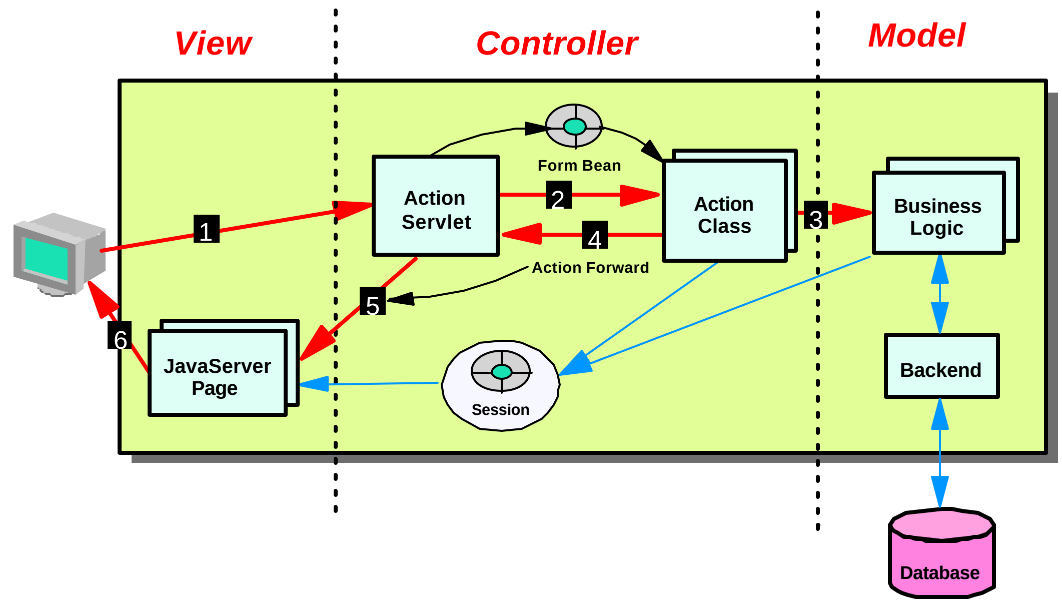
- Form input handling and validation
- Error handling and reporting
- Control flow
- JSP tag libraries to simplify JSP development

Open-source Apache Jakarta project

- **http://jakarta.apache.org/struts/**
- Struts Version 1.0 was released 6/01
- Current 1.0.2 (and 1.1 beta)

A set of cooperating classes, servlets, and JSP tags that make up a reusable MVC design

Struts Application Flow



Struts Components

ActionServlet

- Generic, provided by Struts
- Fills HTML form data into form bean and calls action class
- Calls JSPs or actions based on return from action class

Form bean (subclass of Struts ActionForm)

- Simple JavaBean with form data from HTML
- Performs validation of input data

Action class (subclass of Struts Action)

- Controller, uses form bean and invokes business logic
- Returns an ActionForward to the ActionServlet

ActionForward

- Symbolic name of next action (JSP or action class)
- Used by ActionServlet to invoke next action

Struts binaries distributed in struts.jar

Struts configuration file drives the ActionServlet

Struts Components ...

Struts configuration file (struts-config.xml)

- Contains mapping of actions (in HTML form) to actions classes
- Contains mappings of ActionForwards to JSPs or actions
- Used by the ActionServlet

Struts tag libraries

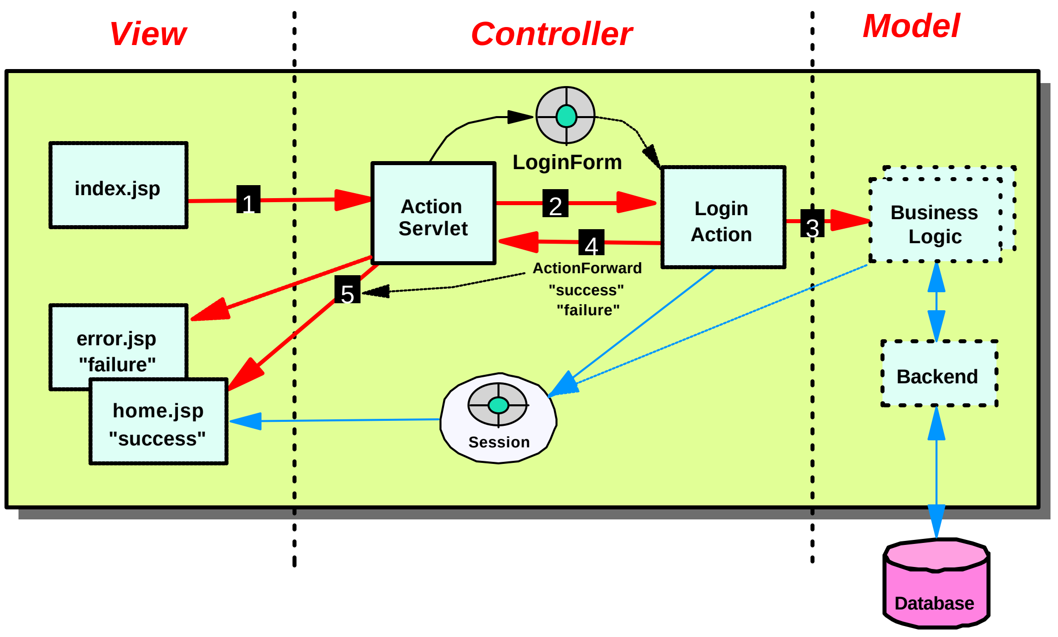
- Ease coding of JSPs through symbolic text variables
  `struts-bean.tld, struts-html.tld, struts-logic.tld, struts-template.tld`
- Texts defined in ApplicationResources.properties
  - **Headings, labels, buttons, error messages, ...**

Error handling

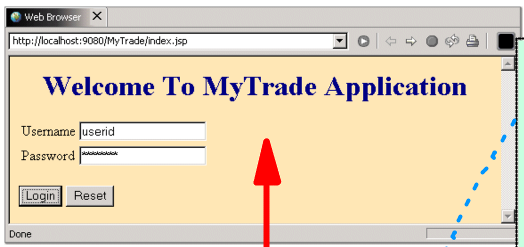
- ActionErrors and ActionError provide easy reporting of error messages in JSPs

Struts Example

Struts Example



Welcome Page Using Struts Tag Libraries


Web Browser
http://localhost:9080/MyTrade/index.jsp
Welcome To MyTrade Application
Username userid
Password
Login Reset
Done

ApplicationResources.properties

```
index.title=Welcome to MyTrade Application
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a user ID.
error.login.failed=Invalid user ID/password entered.
error.login.exception=Exception occurred in action.
# Optional header and footer for <errors/> tag.
errors.header=<ul>
errors.footer=</ul>
```

index.jsp

```
<%@ taglib uri="/WEB-INF/struts-html.tld" prefix="html" %>
......
<h1 align="center"><bean:message key="index.title"/></h1>
<html:form action="/loginAction">

<table>
  <tr><td><bean:message key="global.field.username"/></td>
      <td><html:text property="username" size="20" maxlength="30"/></td> </tr>
  <tr><td><bean:message key="global.field.password"/></td>
      <td><html:password property="password" size="20" maxlength="30"/></td></tr>
</table>
<html:submit><bean:message key="welcome.button.login"/></html:submit>
<input type="reset">
</html:form>
```

Struts Action Servlet

Configured in web.xml (Web deployment descriptor)

- Name:
  `action`
- Class:
  `org.apache.struts.action.ActionServlet`
- Initialization parameters:
  ```
  config: WEB-INF/struts-config.xml
  application: <package>.ApplicationResources
  debug: 2
  detail: 2
  validate: true
  ```
- URL mapping:
  `*.do`
  - **<html:form action="/loginAction"> ==> /loginAction.do**

Initialization:
- **Read struts-config**

Action:
- **Fill form bean from input JSP**
- **Call action class**
- **Process return (ActionForward)**
- **Call output JSP (or other action)**

Redbooks © 2002 IBM Corporation WebSphere Technical Exchange

Struts Configuration File

struts-config.xml (in WEB-INF)

- Form beans
  - **symbolic name ==> class**
- Action mappings for each path
  - **input JSP ==> form bean ==> action class**
  - **forward actions ==> output JSP**

```
<?xml version="1.0" encoding="UTF-8"?>
<struts-config>
    <form-beans>
        <form-bean name="loginForm" type="strutscommon.LoginForm">
        </form-bean>
    </form-beans>
    <action-mappings>
        <action name="loginForm" path="/loginAction" type="strutsaction.LoginAction"
                input="/index.jsp">
            <forward name="success" path="/home.jsp"></forward>
            <forward name="failure" path="/error.jsp"></forward>
        </action>
    </action-mappings>
</struts-config>
```

Redbooks © 2002 IBM Corporation WebSphere Technical Exchange

Form Bean

- Form bean extends ActionForm
- Properties for fields in input JSP
- Reset method to reset field values
- Validate method to check input fields

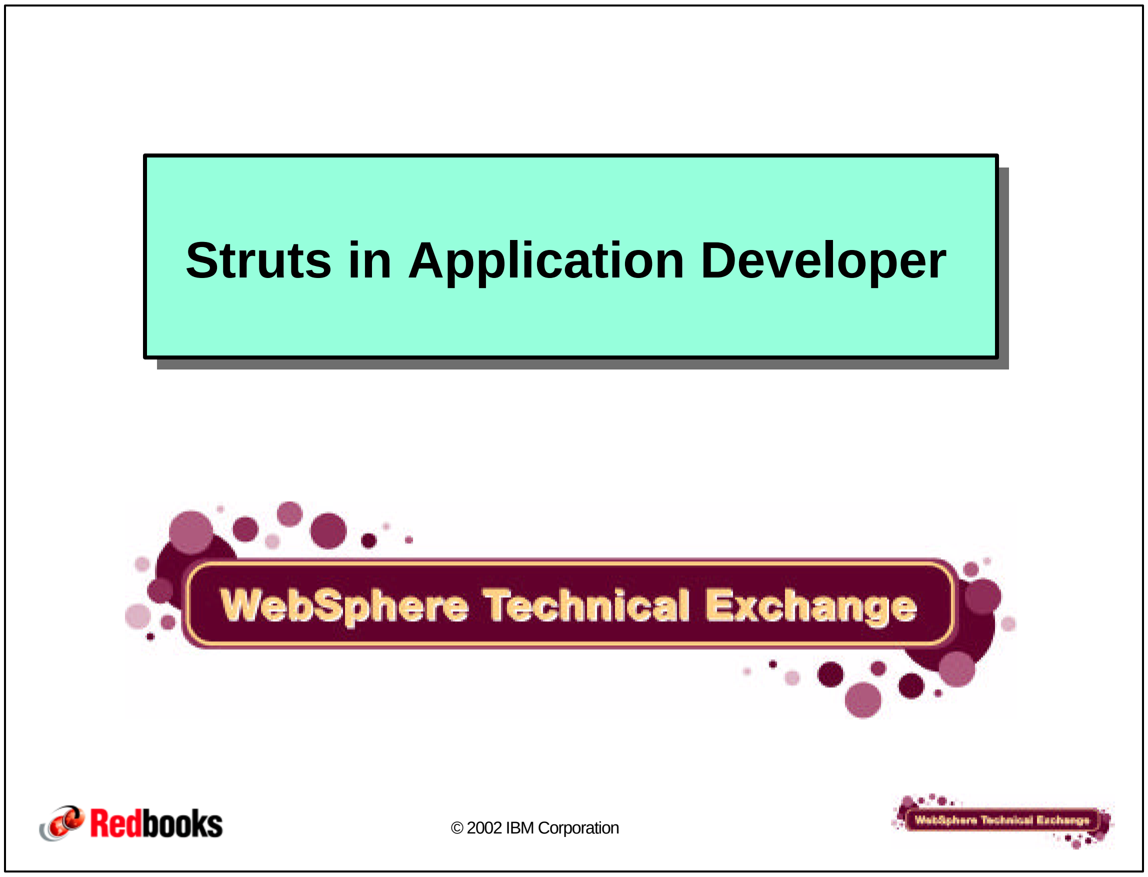
```
public class LoginForm extends ActionForm {

    private java.lang.String username = null;
    private java.lang.String password = null;
    // getter and setter methods (not shown)

    public void reset(ActionMapping mapping, HttpServletRequest request) {
        username = null; password = null;
    }
    public ActionErrors validate(ActionMapping mapping, HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();
        if (username.trim().equals(""))
            errors.add("login", new ActionError("error.login.nouserid"));
        return errors;
    }
}
```

Action Class

```
public class LoginAction extends Action {

    public ActionForward perform (ActionMapping mapping, ActionForm form,
        HttpServletRequest request,HttpServletResponse response) throws ... {
        ActionErrors errors = ActionErrors();
        ActionForward forward = ActionForward();
        LoginForm loginForm = (LoginForm) form;
        try {
            String userID = loginForm.getUsername();
            if (!userID.equals("userid")) {
                errors.add("login", new ActionError("error.login.failed"));
            }
        } catch (Exception e) {
            errors.add("login", new ActionError("error.login.exception"));
        }
        if ( !errors.empty() ) {
            saveErrors(request, errors);
            forward = mapping.findForward("failure");
        } else {
            forward = mapping.findForward("success");
        }
        return (forward);
    }
}
```

Business Logic

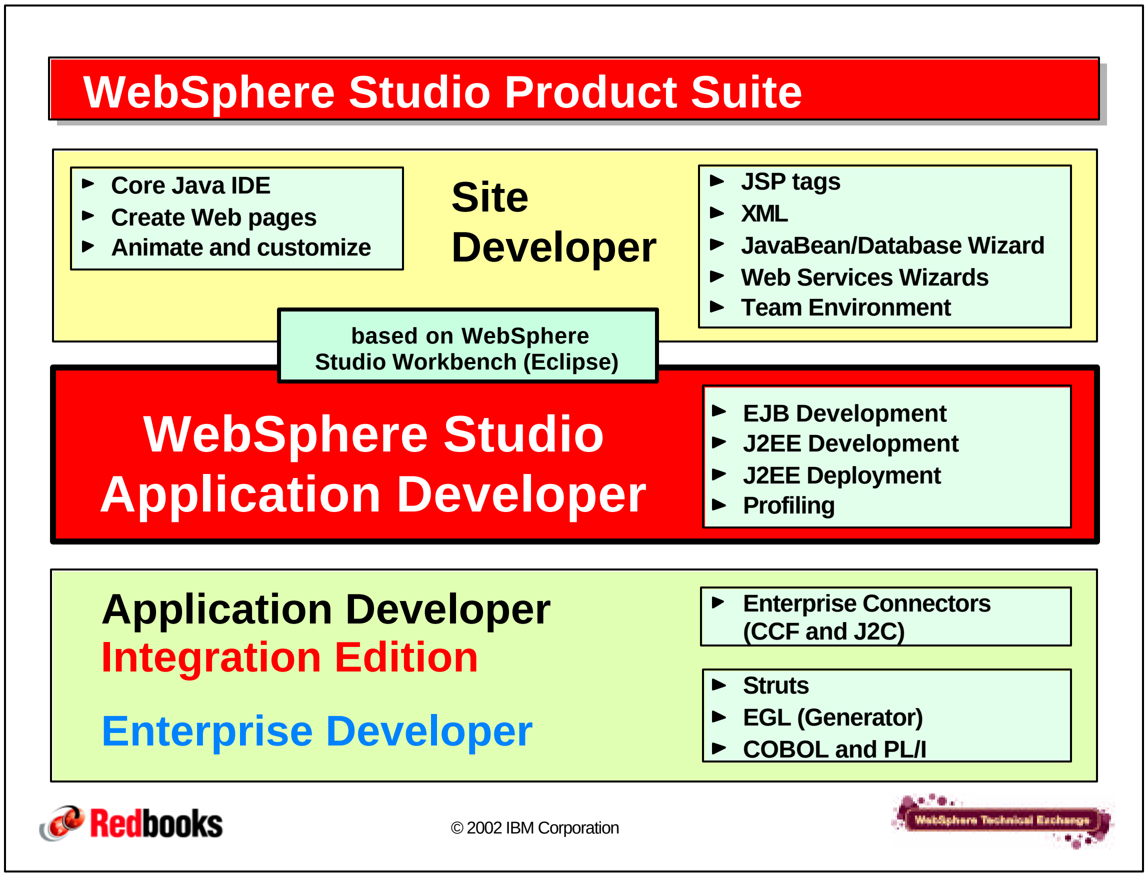
Action Forward



Struts in Application Developer

WebSphere Technical Exchange

Redbooks © 2002 IBM Corporation



WebSphere Studio Product Suite

Site Developer

- Core Java IDE
- Create Web pages
- Animate and customize

- JSP tags
- XML
- JavaBean/Database Wizard
- Web Services Wizards
- Team Environment

based on WebSphere Studio Workbench (Eclipse)

WebSphere Studio Application Developer

- EJB Development
- J2EE Development
- J2EE Deployment
- Profiling

Application Developer Integration Edition

- Enterprise Connectors (CCF and J2C)

Enterprise Developer

- Struts
- EGL (Generator)
- COBOL and PL/I

Redbooks © 2002 IBM Corporation

Configuring a Web Application for Struts

Create Web project (and EAR project)

Tailor Web project:

- Import **struts.jar** to **WEB-INF\lib**
- Import JSP tag library files (**struts-xxx.tld**) to **WEB-INF**
- Add JSP tag libraries in **web.xml**: *References -> JSP Tag Libraries*
  - **Point to imported tag libraries in WEB-INF**
- Add **ActionServlet** in **web.xml**: *Servlets*
  - **Class: org.apache.struts.action.ActionServlet**
  - **Name: action**
  - **URL mappings: *.do**
  - **Initialization parameters: config, application, debug, detail, validate**
  - **Load on startup: 2**
- Create **struts-config.xml** skeleton file in **WEB-INF**
  - **config parameter of ActionServlet points to this file**
- Create **ApplicationResources.properties** file to a **Java package**
  - **application parameter of ActionServlet points to this file**

Manual setup of a Web project with Struts support

Use a skeleton config file

Redbooks © 2002 IBM Corporation WebSphere Technical Exchange

Creating Struts Actions

To add a Struts action to the Web application:

- Create input page (HTML or JSP)
  - **Use JSP tag libraries**
  - **Add texts to ApplicationResources file**
  - **Set an action in the form**
- Create output JSPs
  - **Use <html:errors/> to display error messages.**
- Define the form bean (JavaBean) in a Java package
  - **Properties from input page**
  - **Validate method**
  - **Place error messages into ApplicationResources file**
- Create action class and code the logic and error messages
  - **Place error messages into ApplicationResources file**
- Edit **struts-config.xml** file
  - **Add form bean to <form-beans> section**
  - **Add action to <action-mappings> section with action path and forward names**

Use skeleton models for form beans and action classes

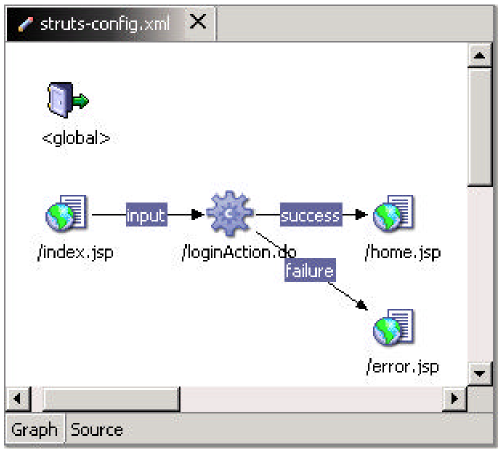
Copy paste existing actions in XML file

Redbooks © 2002 IBM Corporation WebSphere Technical Exchange

Using a Struts Configuration Editor

Search the Internet for Struts Configuration Editor

- ❑ Stand-alone editors and **eclipse plugins** available
- ❑ For example:
  `http://www.improve-technologies.com/alpha/struts-config-editor/`
  - **Associated with Struts config files after install of plugin**
  - **Graphical view**
  - **Source view**
  - **Edit JSPs**
  - **Edit action programs**
  - **Updates only in source view**
    - **does not create any Java code**

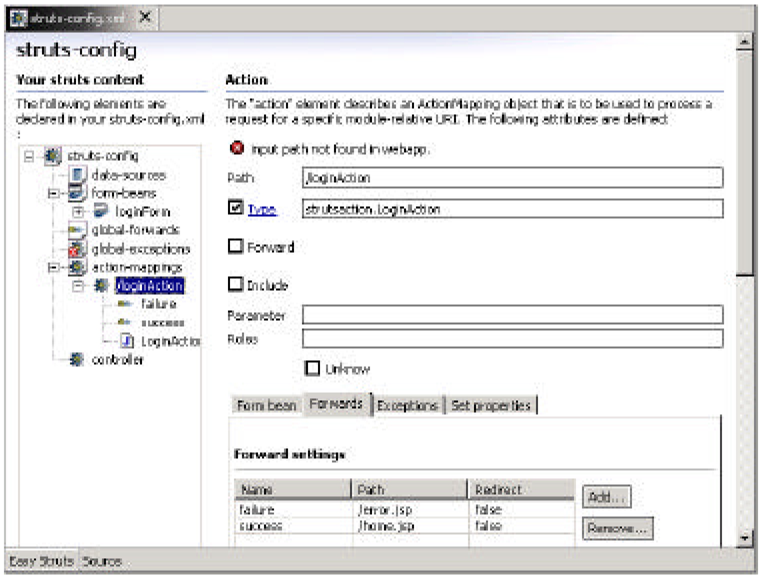


Can also use the standard XML Editor of Workbench

Using a Struts Configuration Editor ...

Easy Struts: `http://easystruts.sourceforge.net`

- ❑ Eclipse plugin
- ❑ Associated with Struts config files after install
- ❑ No graphical view
- ❑ Dialogs to create new form beans and actions in config file
  - **does not create Java code**


struts-config.xml

struts-config

Your struts content

The following elements are declared in your struts-config.xml :

- struts-config
  - data-sources
  - form-beans
    - loginForm
  - global-forwards
  - global-exceptions
  - action-mappings
    - loginAction
      - failure
      - success
      - LoginActio
  - controller

Action

The "action" element describes an ActionMapping object that is to be used to process a request for a specific module-relative URI. The following attributes are defined:

input path not found in webapp.

Path: /loginAction

[✓] Type: strutsaction.LoginAction

☐ Forward

☐ Include

Parameter:

Roles:

☐ Unknow

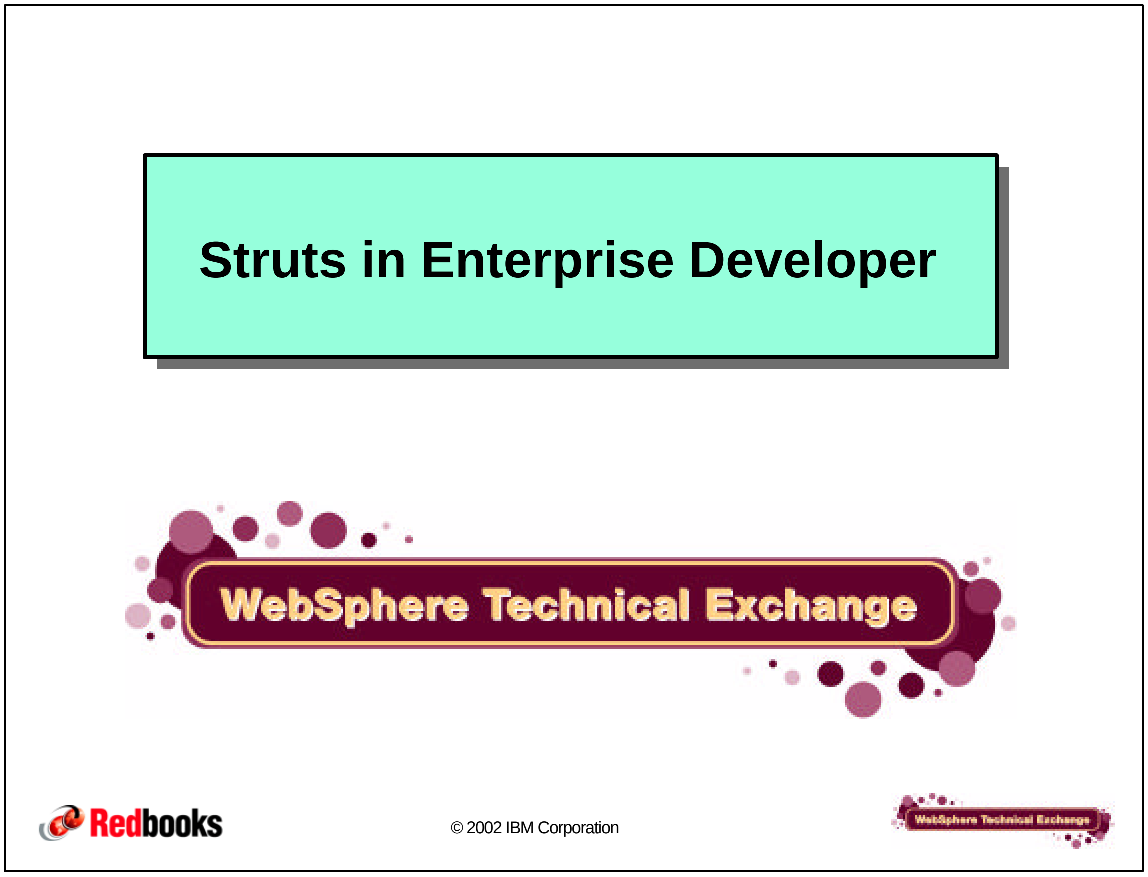
Form bean | Forwards | Exceptions | Set properties

Forward settings

Name	Path	Redirect
failure	/error.jsp	false
success	/home.jsp	false

Add... Remove...

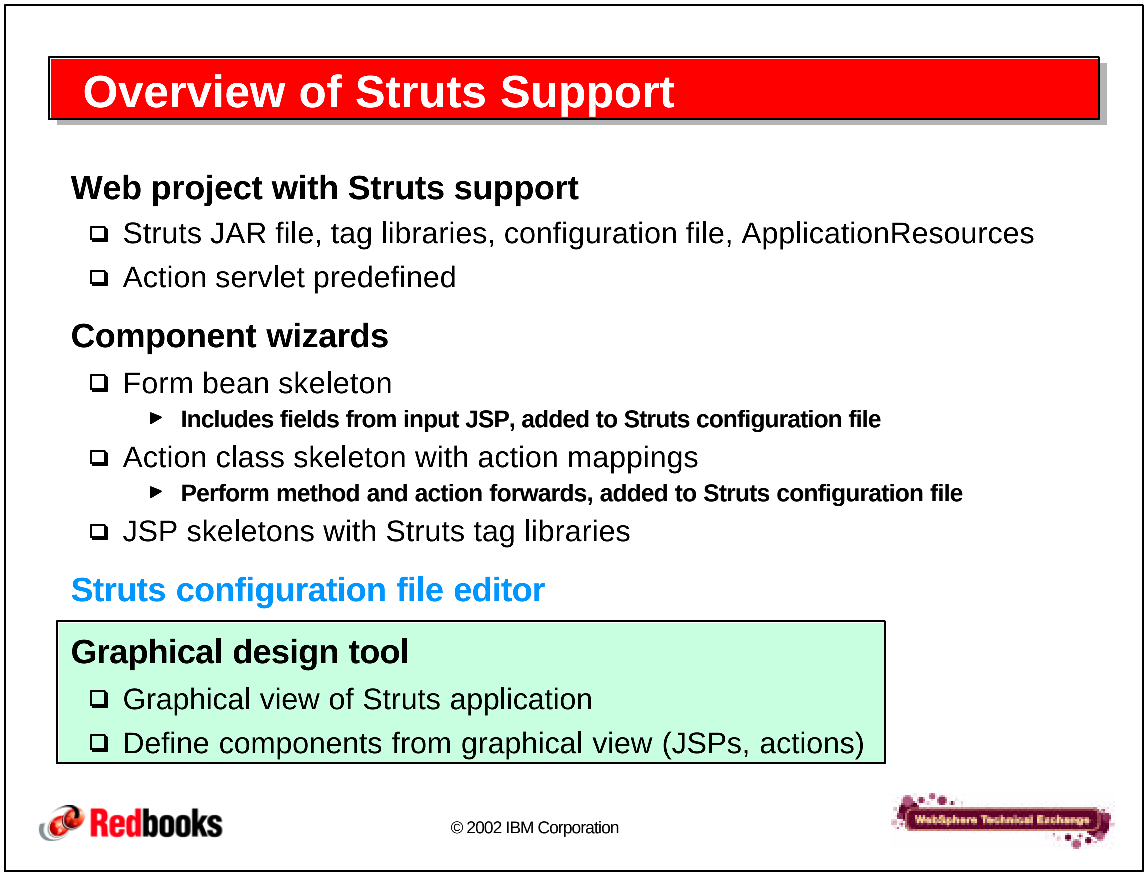
Easy Struts | Source



Struts in Enterprise Developer

WebSphere Technical Exchange

Redbooks © 2002 IBM Corporation



Overview of Struts Support

Web project with Struts support

- Struts JAR file, tag libraries, configuration file, ApplicationResources
- Action servlet predefined

Component wizards

- Form bean skeleton
  - **Includes fields from input JSP, added to Struts configuration file**
- Action class skeleton with action mappings
  - **Perform method and action forwards, added to Struts configuration file**
- JSP skeletons with Struts tag libraries

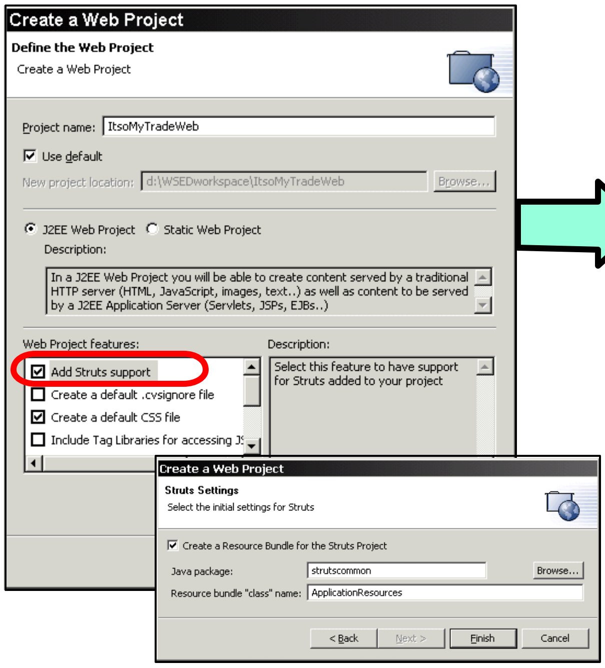
Struts configuration file editor

Graphical design tool

- Graphical view of Struts application
- Define components from graphical view (JSPs, actions)

Redbooks © 2002 IBM Corporation

Web Project with Struts Support



Create a Web Project

Define the Web Project
Create a Web Project

Project name: ItsoMyTradeWeb
☑ Use default
New project location: d:\WSEDworkspace\ItsoMyTradeWeb

[✓ J2EE Web Project] Static Web Project
Description:
In a J2EE Web Project you will be able to create content served by a traditional HTTP server (HTML, JavaScript, images, text..) as well as content to be served by a J2EE Application Server (Servlets, JSPs, EJBs..)

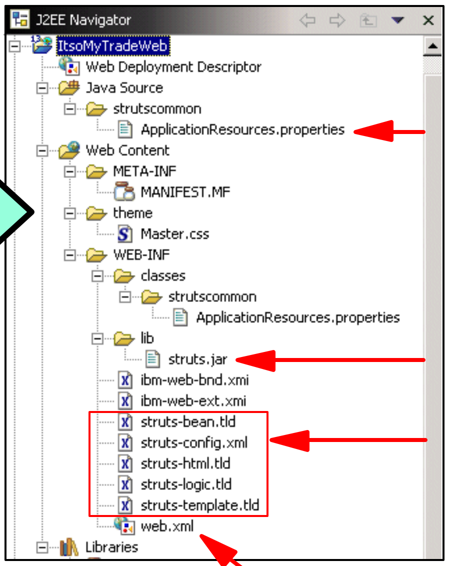
Web Project features:
☑ Add Struts support
☐ Create a default .cvsignore file
☑ Create a default CSS file
☐ Include Tag Libraries for accessing JS

Description:
Select this feature to have support for Struts added to your project

Create a Web Project

Struts Settings
Select the initial settings for Struts

☑ Create a Resource Bundle for the Struts Project
Java package: strutscommon
Resource bundle "class" name: ApplicationResources

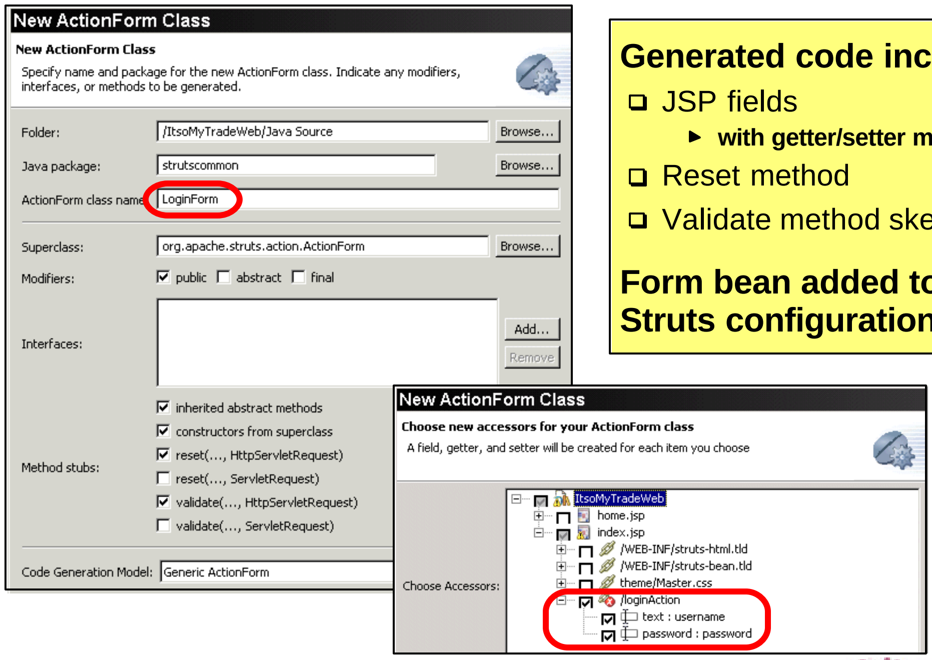


J2EE Navigator

```
ItsoMyTradeWeb
  Web Deployment Descriptor
  Java Source
    strutscommon
      ApplicationResources.properties
  Web Content
    META-INF
      MANIFEST.MF
    theme
      Master.css
    WEB-INF
      classes
        strutscommon
          ApplicationResources.properties
      lib
        struts.jar
      ibm-web-bnd.xmi
      ibm-web-ext.xmi
      struts-bean.tld
      struts-config.xml
      struts-html.tld
      struts-logic.tld
      struts-template.tld
      web.xml
  Libraries
```

ActionServlet

Form Bean Wizard



New ActionForm Class

New ActionForm Class
Specify name and package for the new ActionForm class. Indicate any modifiers, interfaces, or methods to be generated.

Folder: /ItsoMyTradeWeb/Java Source
Java package: strutscommon
ActionForm class name: LoginForm
Superclass: org.apache.struts.action.ActionForm
Modifiers: ☑ public ☐ abstract ☐ final
Interfaces:
Method stubs:
☑ inherited abstract methods
☑ constructors from superclass
☑ reset(..., HttpServletRequest)
☐ reset(..., ServletRequest)
☑ validate(..., HttpServletRequest)
☐ validate(..., ServletRequest)
Code Generation Model: Generic ActionForm

New ActionForm Class

Choose new accessors for your ActionForm class
A field, getter, and setter will be created for each item you choose

Choose Accessors:

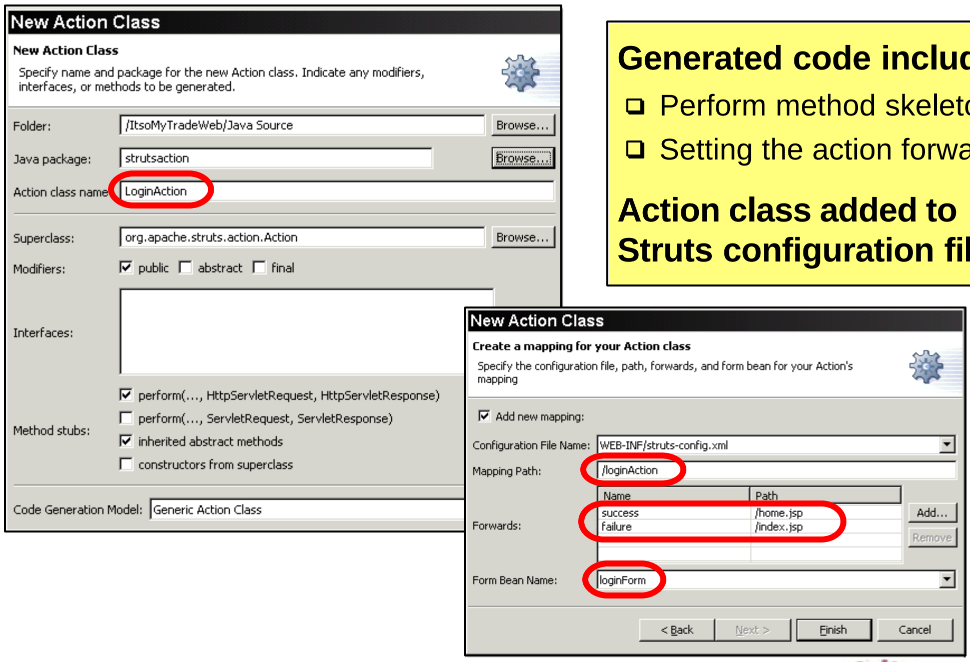
```
ItsoMyTradeWeb
  home.jsp
  index.jsp
    /WEB-INF/struts-html.tld
    /WEB-INF/struts-bean.tld
    theme/Master.css
    ☑ /loginAction
      ☑ text : username
      ☑ password : password
```

Generated code includes:

- JSP fields
  - **with getter/setter methods**
- Reset method
- Validate method skeleton

Form bean added to Struts configuration file

Action Class Wizard



New Action Class

New Action Class

Specify name and package for the new Action class. Indicate any modifiers, interfaces, or methods to be generated.

Folder: /ItsoMyTradeWeb/Java Source

Java package: strutsaction

Action class name: LoginAction

Superclass: org.apache.struts.action.Action

Modifiers: [✓ public] abstract final

Interfaces:

Method stubs:
- [✓ perform(..., HttpServletRequest, HttpServletResponse)]
- perform(..., ServletRequest, ServletResponse)
- [✓ inherited abstract methods]
- constructors from superclass

Code Generation Model: Generic Action Class

New Action Class

Create a mapping for your Action class

Specify the configuration file, path, forwards, and form bean for your Action's mapping

[✓ Add new mapping:]

Configuration File Name: WEB-INF/struts-config.xml

Mapping Path: /loginAction

Forwards:

Name	Path
success	/home.jsp
failure	/index.jsp

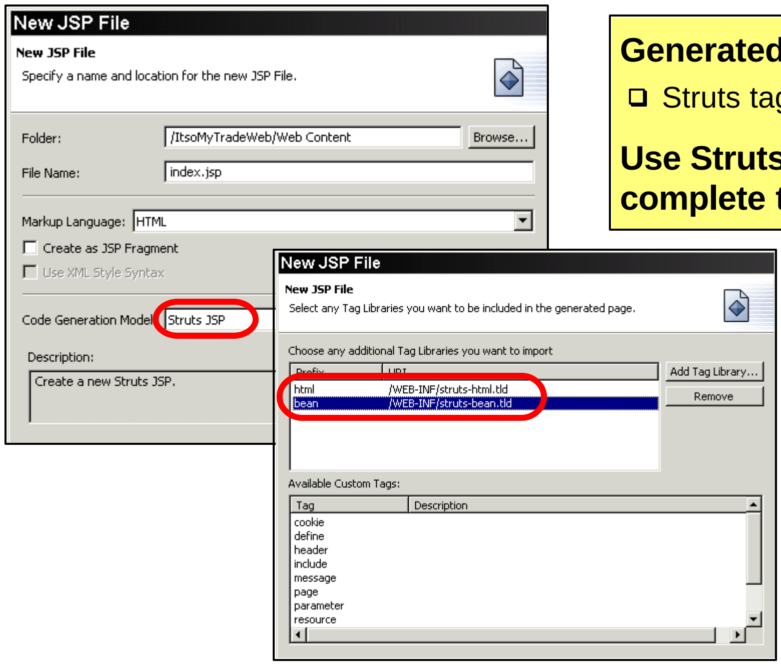
Form Bean Name: loginForm

Generated code includes:
- Perform method skeleton
- Setting the action forward

Action class added to Struts configuration file

© 2002 IBM Corporation

JSP Skeletons



New JSP File

New JSP File

Specify a name and location for the new JSP File.

Folder: /ItsoMyTradeWeb/Web Content

File Name: index.jsp

Markup Language: HTML

Create as JSP Fragment

Use XML Style Syntax

Code Generation Model: Struts JSP

Description:

Create a new Struts JSP.

New JSP File

New JSP File

Select any Tag Libraries you want to be included in the generated page.

Choose any additional Tag Libraries you want to import

Prefix	URI
html	/WEB-INF/struts-html.tld
bean	/WEB-INF/struts-bean.tld

Available Custom Tags:

Tag	Description
cookie	
define	
header	
include	
message	
page	
parameter	
resource	

Generated code includes:
- Struts tag libraries

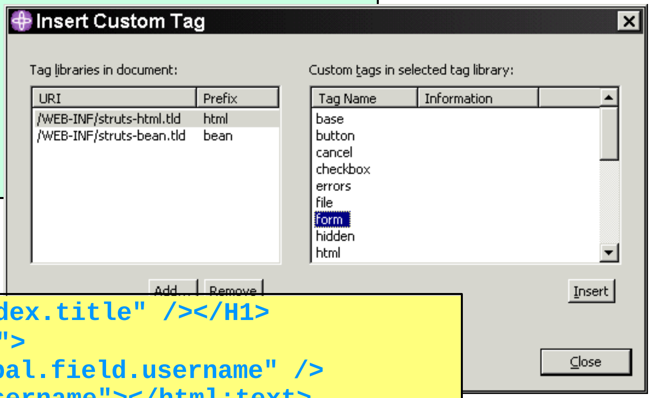
Use Struts custom tags to complete the JSP

© 2002 IBM Corporation

JSP Editor: Page Designer

Page Designer supports custom tag libraries:

- ***JSP -> Insert Custom***
  - **html -> form**
  - **html -> text, password**
  - **html -> submit**
  - **bean -> message**
  - **......**

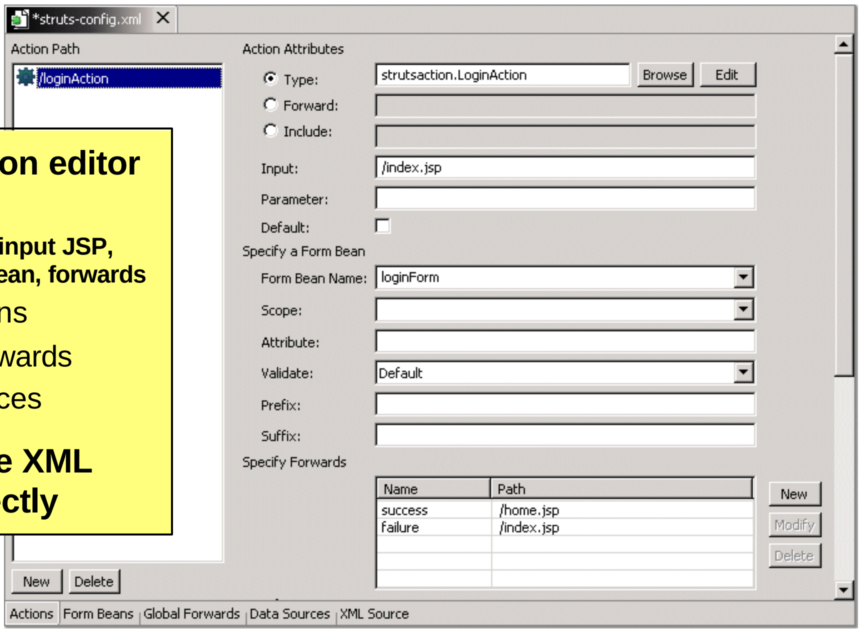

Insert Custom Tag
Tag libraries in document:
Custom tags in selected tag library:
Add...
Remove
Insert
Close

URI	Prefix
/WEB-INF/struts-html.tld	html
/WEB-INF/struts-bean.tld	bean

Tag Name	Information
base	
button	
cancel	
checkbox	
errors	
file	
form	
hidden	
html	

```
<H1><bean:message key="index.title" /></H1>
<html:form action="/login">
   <bean:message key="global.field.username" />
   <html:text property="username"></html:text>
   <bean:message key="global.field.password" />
   <html:password property="password"></html:password>
   <html:submit>
      <bean:message key="welcome.button.login" />
   </html:submit>
</html:form>
```

Struts Configuration File Editor



/loginAction

Type: [✓] strutsaction.LoginAction
Forward:
Include:
Input: /index.jsp
Parameter:
Default: ☐
Specify a Form Bean
Form Bean Name: loginForm
Scope:
Attribute:
Validate: Default
Prefix:
Suffix:
Specify Forwards

Name	Path
success	/home.jsp
failure	/index.jsp

Configuration editor

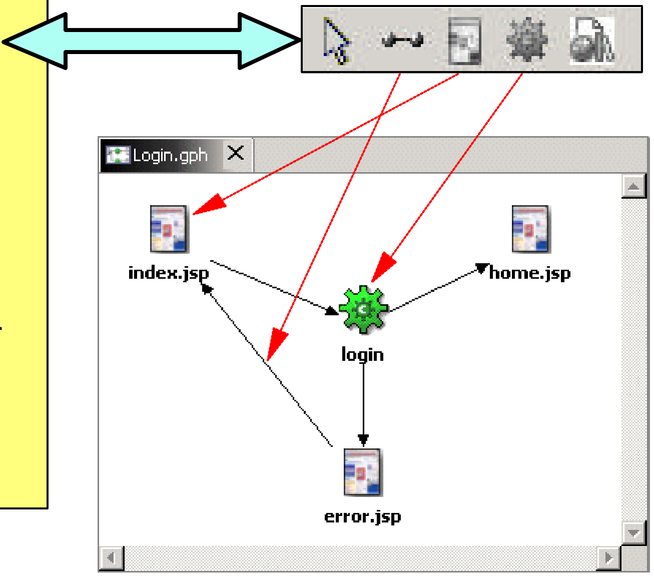
- Actions
  - **Class, input JSP, form bean, forwards**
- Form beans
- Global forwards
- Data sources

Can edit the XML source directly

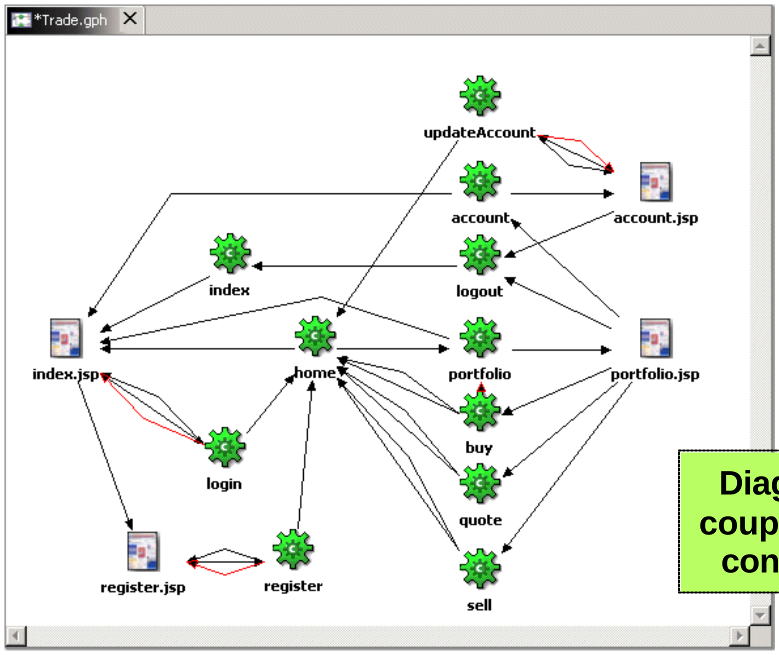
Struts Graphical Design Tool

Web Diagram Editor

- Icons to add components and connections
- Double-click on new components
  - **Wizard to define new component (JSP, action)**
- Double-click on realized components
  - **JSP editor**
  - **Struts configuration editor**
- Each node has path and description
- Connections have action forward names



Struts Graphical Design Tool ...



Diagrams can be tailored

- Layout of components
- Connections

Diagram editor is coupled with Struts configuration file

User Coding

JSPs

- Layout using Struts custom tags

Form bean

- validate logic

```
if (username.trim().equals(""))
errors.add("login",new ActionError("error.login.nouser"));
```

Action class

- perform logic

```
try {
   String userID = loginForm.getUsername();
   if (!userID.equals("userid")) {
      errors.add("login", new ActionError("error.xxx"));
   }
} catch (Exception e) {
   errors.add("login", new ActionError("error.yyy"));
}
```

Real business logic in JavaBeans (EJBs) that are called from the action class

Enterprise Generation Language

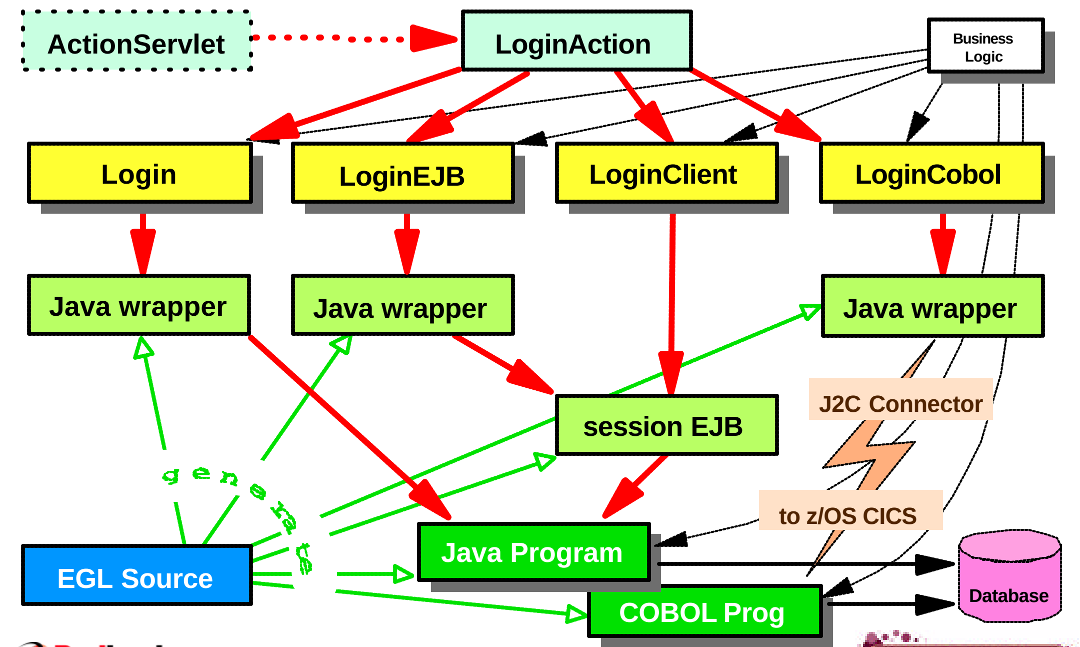
Enterprise Generation Language (EGL)

- Future replacement/migration path for VisualAge Generator
  - **Environment independent language**
- High-level programming specification
  - **Generates Java for Windows and z/OS UNIX**
  - **Generates COBOL for z/OS CICS transactions**
- EGL parts
  - **Programs and functions (4GL)**
  - **Items, structures, records (including SQL)**
  - **Control parts (build descriptors, linkage options)**
- Iterative development and test
  - **built-in EGL debugger**
- Code generation with distributed build processors

Can generate Struts actions business logic using EGL

- Call through Java wrapper or session EJB (generated from EGL)

Struts and EGL Alternatives



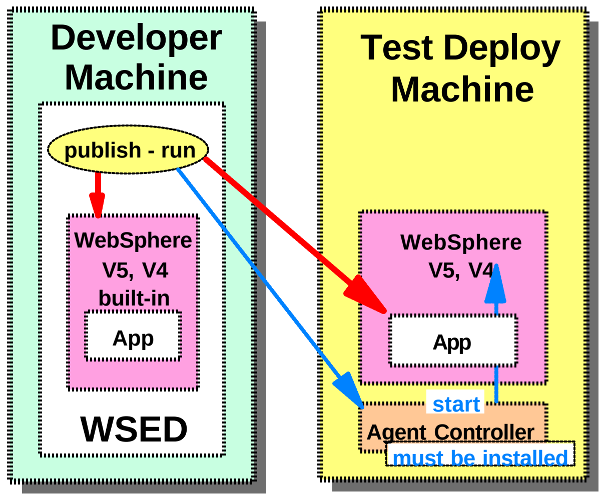
Testing of Struts Applications

Enterprise Developer includes WebSphere Application Server

- Version 4 server for J2EE 1.2
- Version **5** server for **J2EE 1.3** or 1.2

Process:

- Define a server
- Configure the server with **data sources**
- Assign projects to server
- Start server
  - **normal or debug mode**
- Run project
- Universal test client for EJBs and Web services



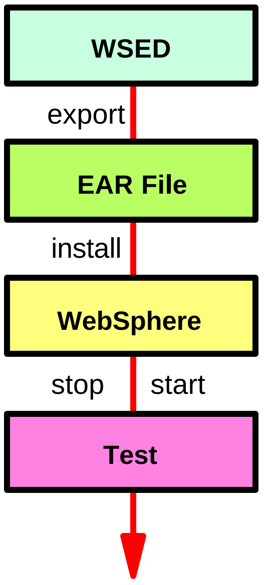
Deployment to WebSphere

Configure WebSphere with Admin Console
- JDBC drivers and Data sources

Export application as EAR file
- Contains Web and EJB modules
- **Web module contains Struts support**
- Optionally tailor EAR with AAT

Install EAR file with Admin Console
- Configure JNDI names
- Configure EJB references
- Do not redeploy EJBs (code is generated)
- Configure J2C connector

Stop/start server and test

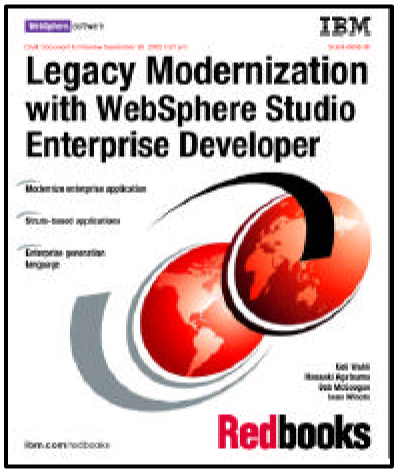


Summary

Struts helps to implement model-view-controller architecture
- One servlet
- View implemented using JSPs
- Action classes to link to business logic

Struts in Application Developer
- Manual work
- Some eclipse plug-ins available

Struts support in Enterprise Developer
- Web projects with Struts support built-in
- Wizards
- Graphical editor
- Enterprise Generation Language


IBM
Legacy Modernization with WebSphere Studio Enterprise Developer
Redbooks

Redbook: SG24-6806 Legacy Modernization with WebSphere Studio Enterprise Developer

N O T E S

THE END

Redbooks

International Technical Support Organization

ibm.com/redbooks

WebSphere software

IBM

e business software

WebSphere Studio Enterprise Developer - Enterprise Generation Language (EGL) and Struts Technical Overview

Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

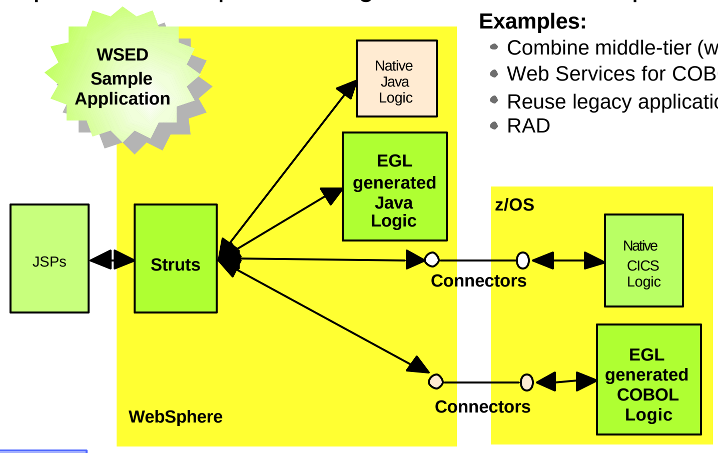
IBM Software Group

WebSphere Studio Enterprise Developer V5.0

- Struts Tools
  - Set of Wizards, editors, and validation support
  - for the design and construction of Struts-based J2EE web applications
- Enterprise Generation Language (EGL)
  - Simple, high level programming specifications
  - for creating full-function COBOL and Java applications
- z/OS Application Development Tools
  - Interactive, workstation-based development
  - for mainframe COBOL, PL/I, ASM applications
- XML Enablement Enhancements for z/OS applications
  - Set of wizards to create XML transformation code
  - and web services for XML-enabled z/OS applications

Extending the Enterprise to the Web

- Comprehensive end-to-end development environment
  - ►Build large-scale, dynamic web applications and services
  - ►that leverage heterogeneous technologies and skill sets
- Speeds developers through the entire development process

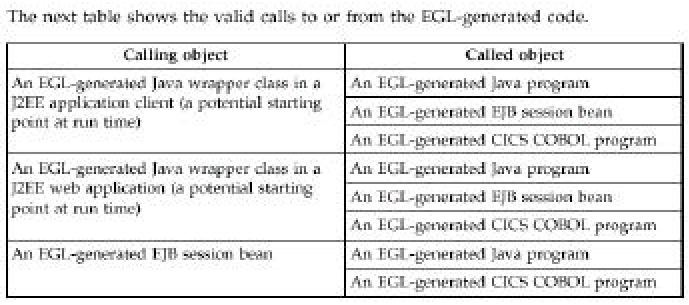


Examples:
- Combine middle-tier (web) and host logic
- Web Services for COBOL applications
- Reuse legacy applications
- RAD

WebSphere software

IBM

Extending the Enterprise to the Web



The next table shows the valid calls to or from the EGL-generated code.

Calling object	Called object
An EGL-generated Java wrapper class in a J2EE application client (a potential starting point at run time)	An EGL-generated Java program
	An EGL-generated EJB session bean
	An EGL-generated CICS COBOL program
An EGL-generated Java wrapper class in a J2EE web application (a potential starting point at run time)	An EGL-generated Java program
	An EGL-generated EJB session bean
	An EGL-generated CICS COBOL program
An EGL-generated EJB session bean	An EGL-generated Java program
	An EGL-generated CICS COBOL program

WebSphere software

IBM

Key Benefits of Enterprise Developer

- Struts Tools
  - ►Rapid design and quicker understanding of complex web applications
  - ►Faster development with less errors of well-structured web applications
- Enterprise Generation Language
  - ►Rapid development
  - ►Cross platform applications (CICS, WebSphere Application Server)
  - ►Using existing programmers with traditional business skills
- z/OS Application Development
  - ►Extends benefits of WebSphere Studio Workbench features/tools to COBOL, PL/I, ASM programmers
  - ►and the z/OS applications they create and maintain
- XML Enablement Enhancements for z/OS applications
  - ►Faster development of XML-enabled z/OS applications and of supporting Web Services

Struts Tools

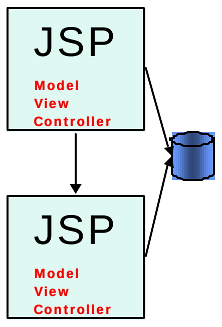
Struts Tools

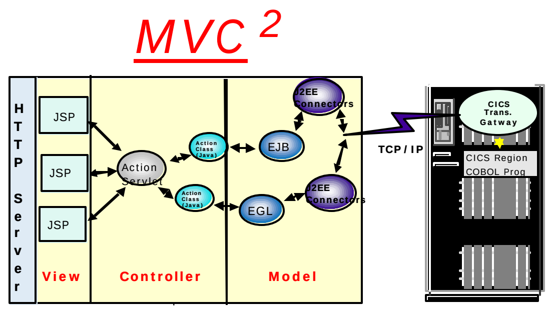
- Rapid design and construction of J2EE web applications
  - Promotes well-structured web applications
  - Enables development in less time with fewer errors
  - Connects to business logic of choice
    - EJBs and Java classes
    - COBOL, PL/I, EGL programs
- Wizards and editors
  - Setup J2EE web project with Struts support
  - Create Struts components
  - Struts configuration file editor
  - Web diagram editor
    - Visual design and assembly of web applications
- Build Support
  - Validates changes against existing resources and identifies errors

WebSphere software IBM

Why Model-View-Controller 2?

From monolithic ⇨ To well-structured

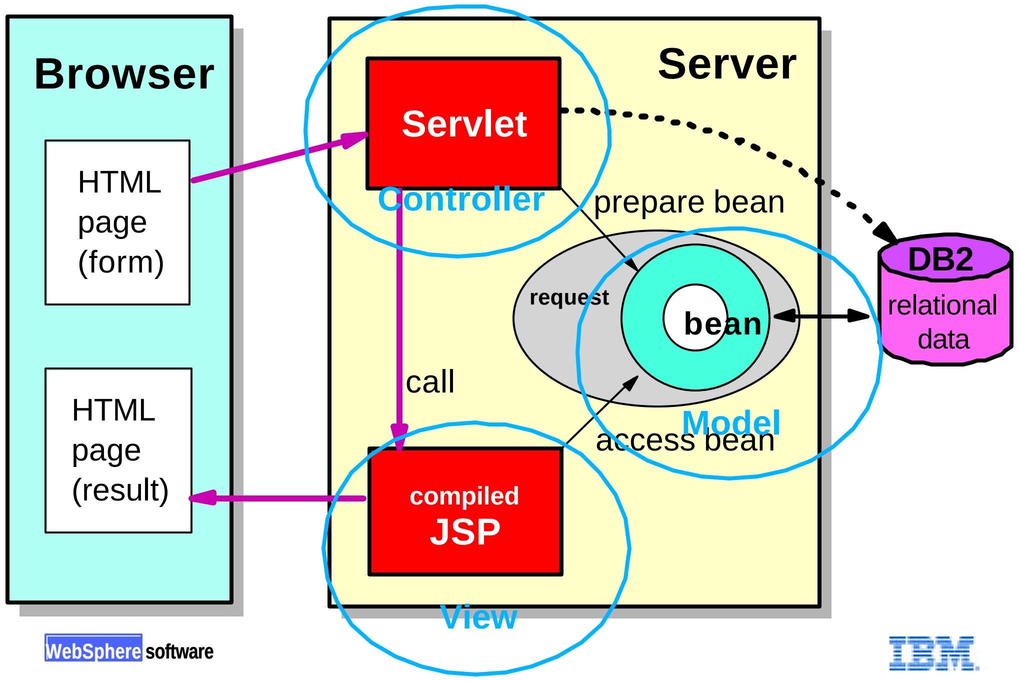




MVC value:
- applications are more adaptable to change
- e-business apps are more maintainable
- Reduces technical expertise required
- Includes all developer roles in process

WebSphere software IBM

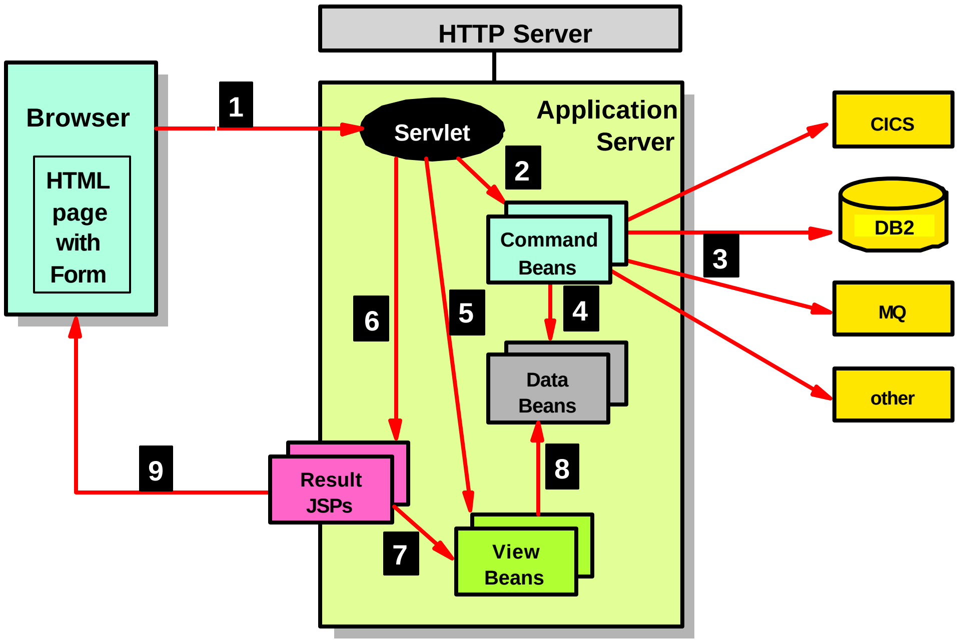
Model-View-Controller



WebSphere software

IBM

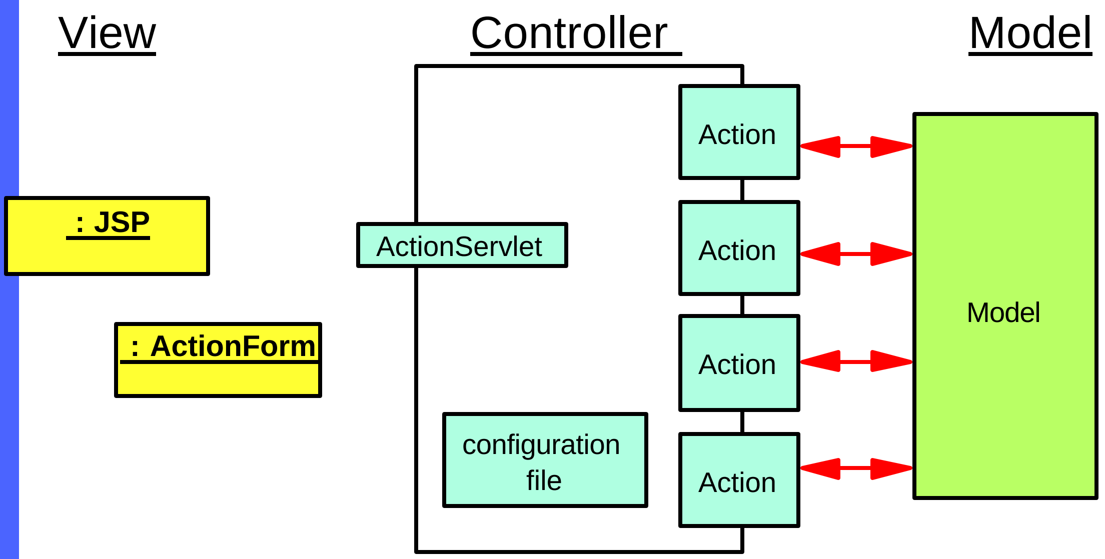
Model-View-Controller



WebSphere software

IBM

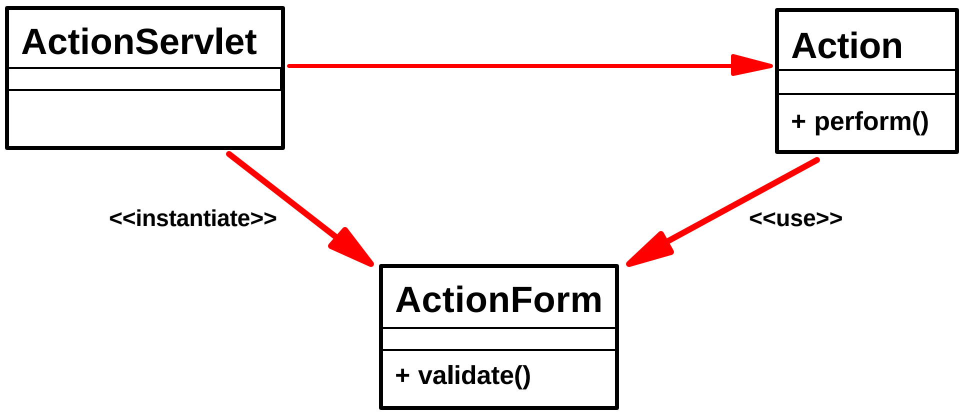
Struts Components



WebSphere software

IBM

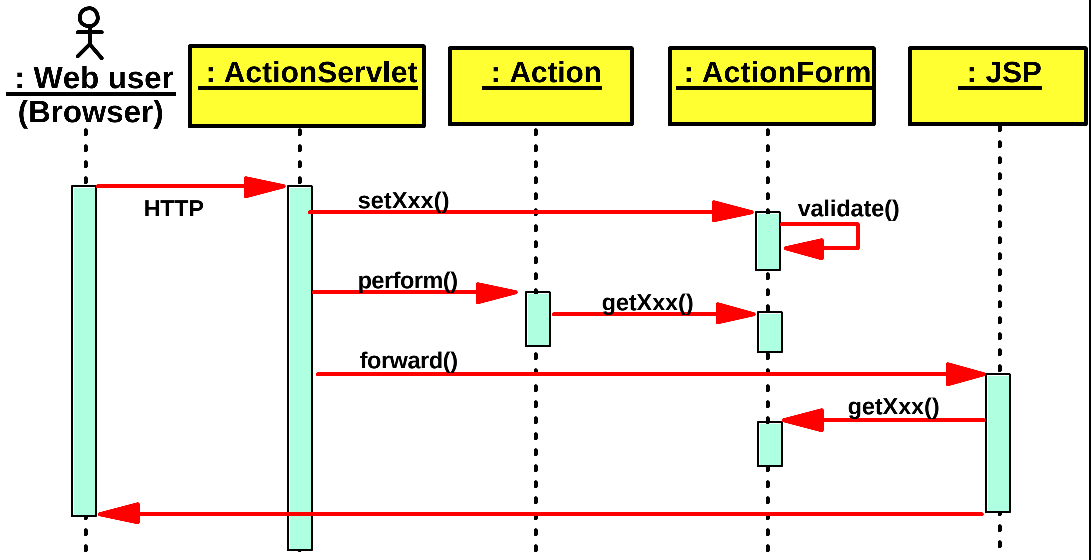
Struts Action Form Handling



WebSphere software

IBM

Struts Request Sequence



WebSphere software

IBM

Struts Configuration

Configurations

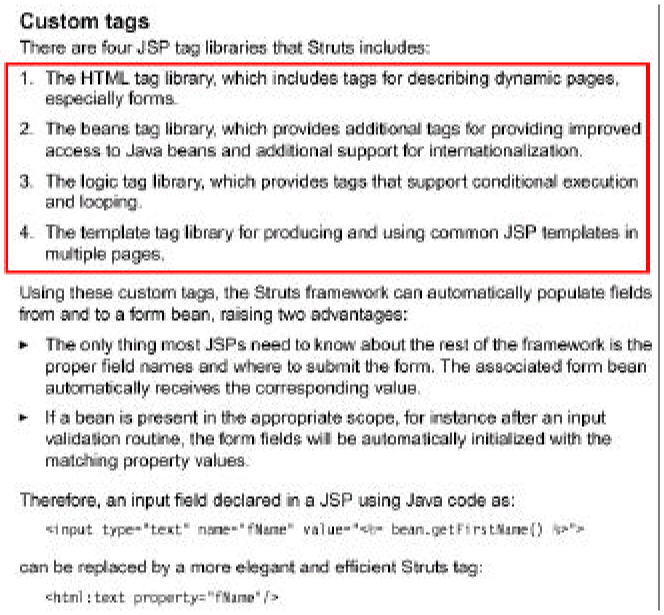
Struts includes a servlet that implements the primary function of mapping a request URI to an action class. Therefore, your primary responsibilities related to the controller are:

- Write an action class for each logical request that may be received (extend `org.apache.action.Action`).
- Configure an action mapping (in XML) for each logical request that may be submitted. The XML configuration file is usually named *struts-config.xml*.
- Update the Web application deployment descriptor file (in XML) for your application to include the necessary Struts components.
- Add the appropriate Struts components to your application.

WebSphere software

IBM

Struts Custom tags



Custom tags

There are four JSP tag libraries that Struts includes:

1. The HTML tag library, which includes tags for describing dynamic pages, especially forms.
2. The beans tag library, which provides additional tags for providing improved access to Java beans and additional support for internationalization.
3. The logic tag library, which provides tags that support conditional execution and looping.
4. The template tag library for producing and using common JSP templates in multiple pages.

Using these custom tags, the Struts framework can automatically populate fields from and to a form bean, raising two advantages:

- The only thing most JSPs need to know about the rest of the framework is the proper field names and where to submit the form. The associated form bean automatically receives the corresponding value.
- If a bean is present in the appropriate scope, for instance after an input validation routine, the form fields will be automatically initialized with the matching property values.

Therefore, an input field declared in a JSP using Java code as:

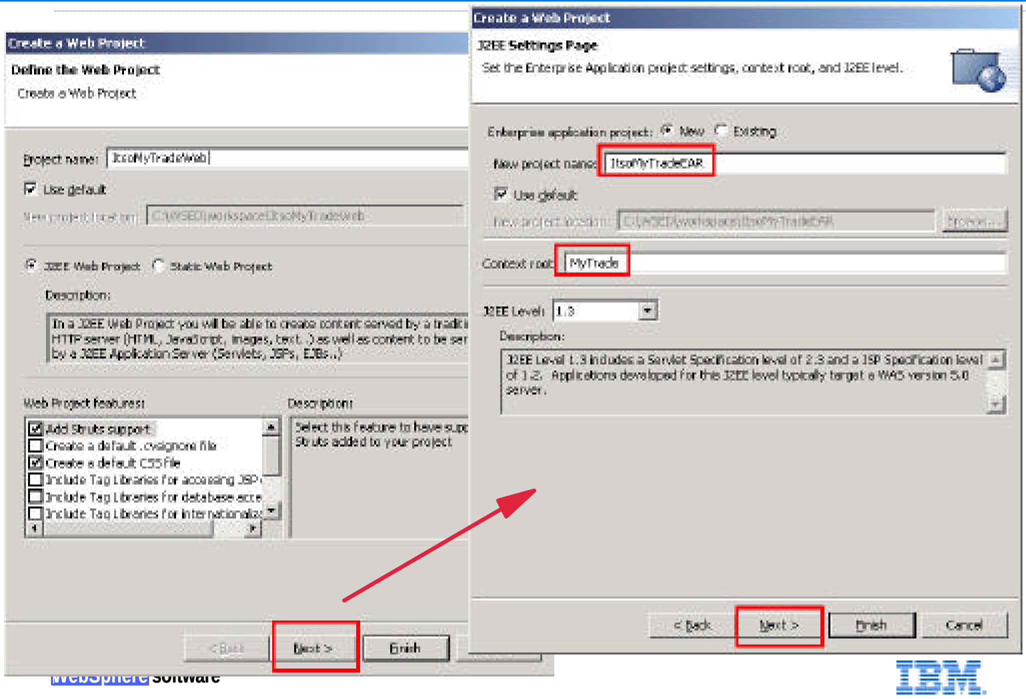
```
<input type="text" name="fName" value="<%= bean.getFirstName() %>">
```

can be replaced by a more elegant and efficient Struts tag:

```
<html:text property="fName"/>
```

WebSphere software

Example: Creating Struts Application



Create a Web Project

Define the Web Project

Create a Web Project

Project name: ItsoMyTradeWeb

[✓] Use default

New project location: C:\WSED\workspace\ItsoMyTradeWeb

[✓ J2EE Web Project] Static Web Project

Description:

In a J2EE Web Project you will be able to create content served by a tradi… HTTP server (HTML, JavaScript, images, text…) as well as content to be ser… by a J2EE Application Server (Servlets, JSPs, EJBs..)

Web Project features:

- [✓] Add Struts support
- [ ] Create a default .cvsignore file
- [✓] Create a default CSS file
- [ ] Include Tag Libraries for accessing JSP…
- [ ] Include Tag Libraries for database acce…
- [ ] Include Tag Libraries for internationaliz…

Description:

Select this feature to have supp… Struts added to your project

Create a Web Project

J2EE Settings Page

Set the Enterprise Application project settings, context root, and J2EE level.

Enterprise application project: [✓ New] Existing

New project name: ItsoMyTradeEAR

[✓] Use default

New project location: C:\WSED\workspace\ItsoMyTradeEAR

Context root: MyTrade

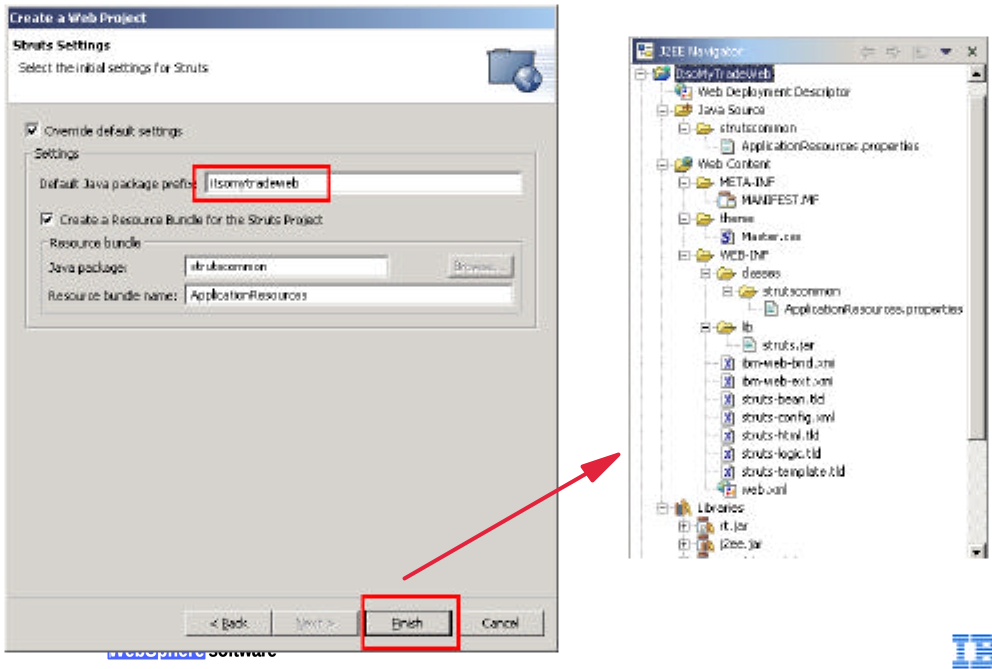
J2EE Level: 1.3

Description:

J2EE Level 1.3 includes a Servlet Specification level of 2.3 and a JSP Specification level of 1.2. Applications developed for this J2EE level typically target a WAS version 5.0 server.

WebSphere software

Example: Creating Struts Application...



Create a Web Project

Struts Settings

Select the initial settings for Struts

☑ Override default settings

Settings

Default Java package prefix: itsomytradeweb

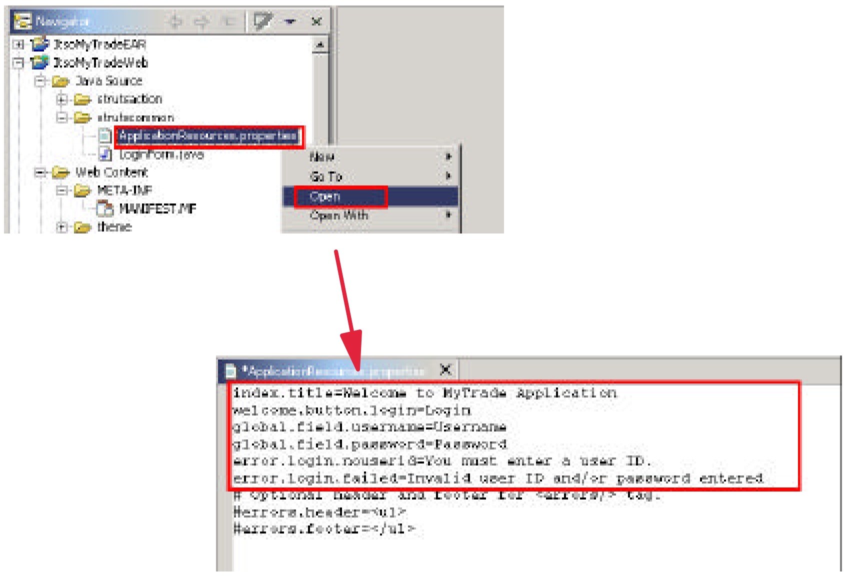
☑ Create a Resource Bundle for the Struts Project

Resource bundle

Java package: strutscommon

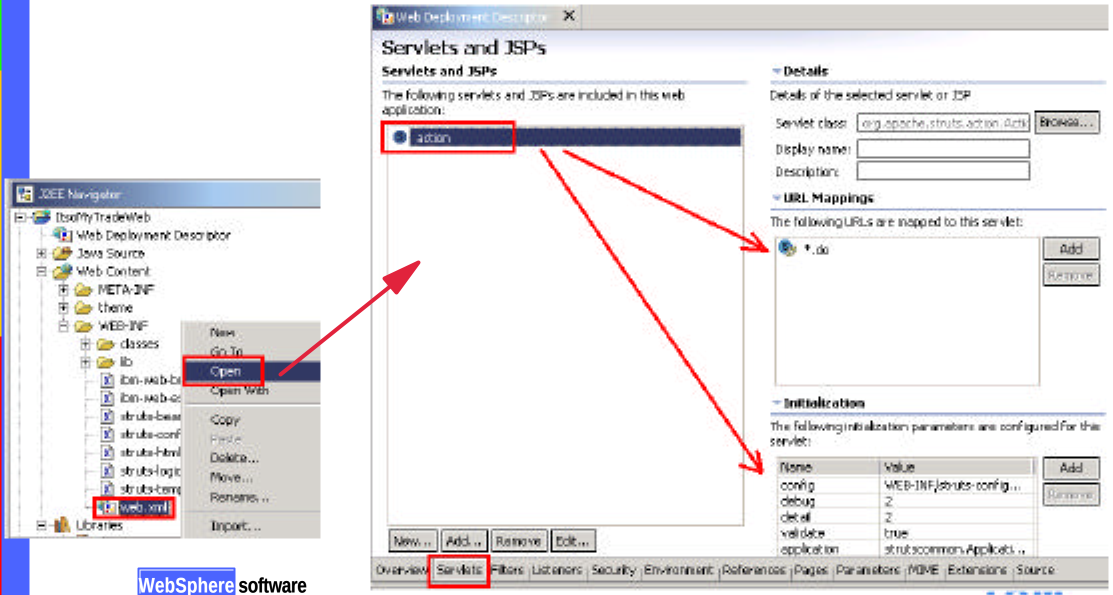
Resource bundle name: ApplicationResources

Example: Editing resources properties file...



```
index.title=Welcome to MyTrade Application
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a user ID.
error.login.failed=Invalid user ID and/or password entered
# Optional header and footer for <errors/> tag.
#errors.header=<ul>
#errors.footer=</ul>
```

Example: Web deployment descriptor...



Servlets and JSPs

Servlets and JSPs

The following servlets and JSPs are included in this web application:

action

Details

Details of the selected servlet or JSP

Servlet class: org.apache.struts.action.Acti

Display name:

Description:

URL Mappings

The following URLs are mapped to this servlet:

*.do

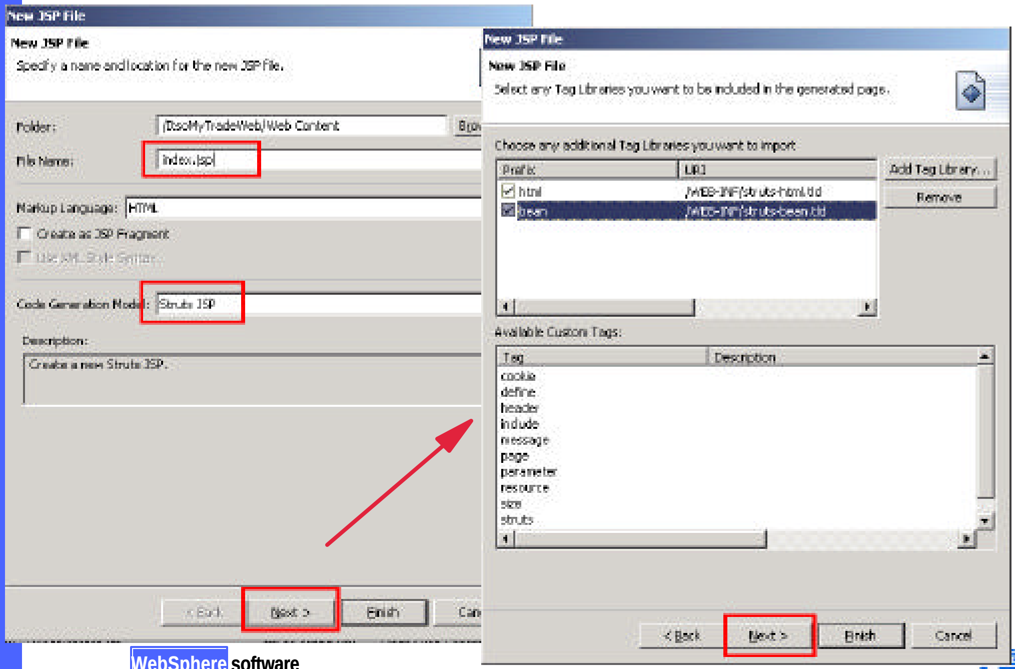
Initialization

The following initialization parameters are configured for this servlet:

Name	Value
config	WEB-INF/struts-config...
debug	2
detail	2
validate	true
application	strutscommon.Applicati...

WebSphere software

Example: Adding JSPs to Appl (index.jsp)...



New JSP File

New JSP File

Specify a name and location for the new JSP file.

Folder: /ItsoMyTradeWeb/Web Content

File Name: index.jsp

Markup Language: HTML

Create as JSP Fragment

Use XML Style Syntax

Code Generation Model: Struts JSP

Description:

Create a new Struts JSP.

New JSP File

New JSP File

Select any Tag Libraries you want to be included in the generated page.

Choose any additional Tag Libraries you want to import

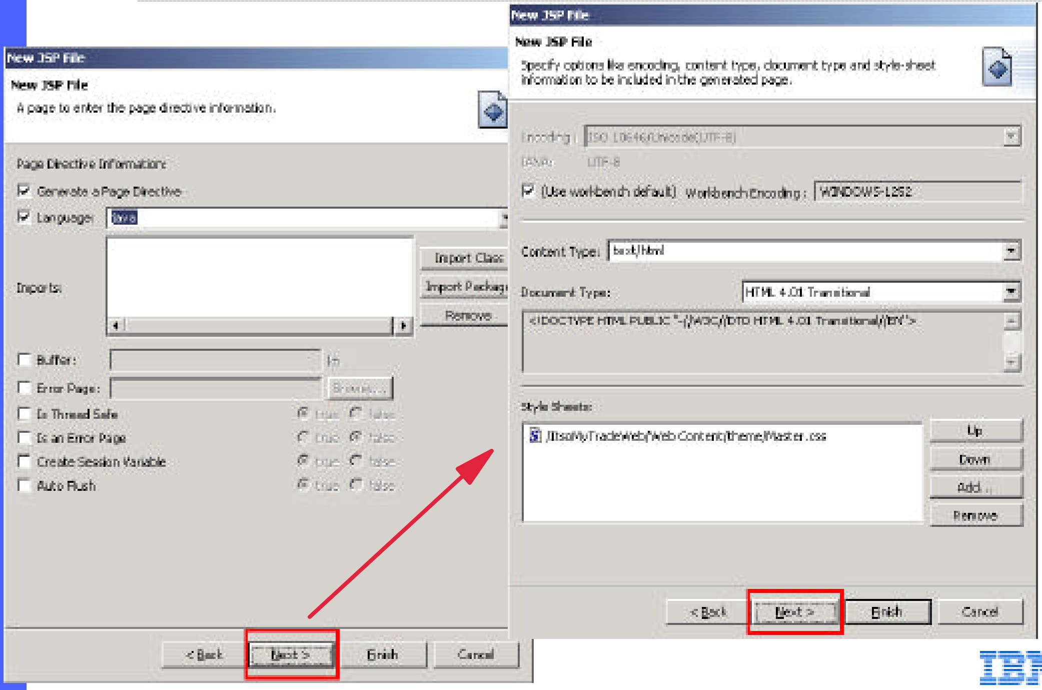
Prefix	URI
✓ html	/WEB-INF/struts-html.tld
✓ bean	/WEB-INF/struts-bean.tld

Available Custom Tags:

Tag	Description
cookie	
define	
header	
include	
message	
page	
parameter	
resource	
size	
struts	

WebSphere software

Example: Adding JSPs to Appl (index.jsp)...


New JSP File

A page to enter the page directive information.

Page Directive Information:

[✓] Generate a Page Directive

[✓] Language: java

Imports:

[ ] Buffer: kb

[ ] Error Page:

[ ] Is Thread Safe [✓ true] false

[ ] Is an Error Page true [✓ false]

[ ] Create Session Variable [✓ true] false

[ ] Auto Flush [✓ true] false

New JSP File

Specify options like encoding, content type, document type and style-sheet information to be included in the generated page.

Encoding: ISO 10646/Unicode(UTF-8)

IANA: UTF-8

[✓] (Use workbench default) Workbench Encoding: WINDOWS-1252

Content Type: text/html

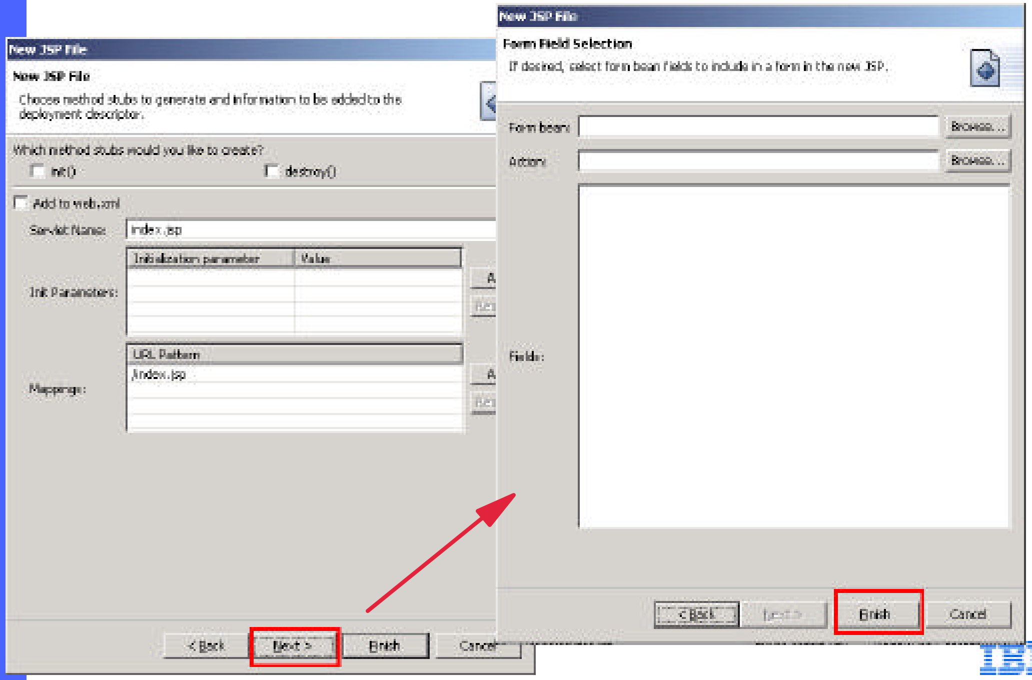
Document Type: HTML 4.01 Transitional

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

Style Sheets:

/ItsoMyTradeWeb/Web Content/theme/Master.css

Example: Adding JSPs to Appl (index.jsp)...


New JSP File

Choose method stubs to generate and information to be added to the deployment descriptor.

Which method stubs would you like to create?

[ ] init() [ ] destroy()

[ ] Add to web.xml

Servlet Name: index.jsp

Init Parameters:

Initialization parameter	Value

Mappings:

URL Pattern
/index.jsp

Form Field Selection

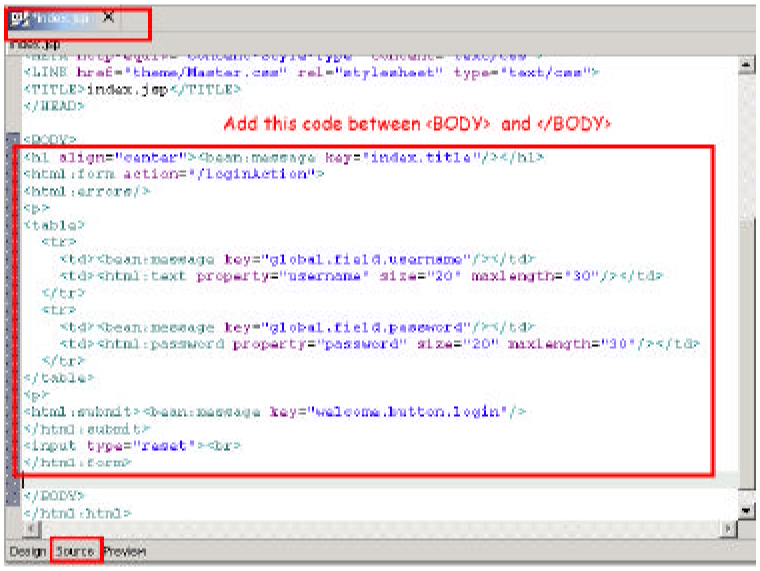
If desired, select form bean fields to include in a form in the new JSP.

Form bean:

Action:

Fields:

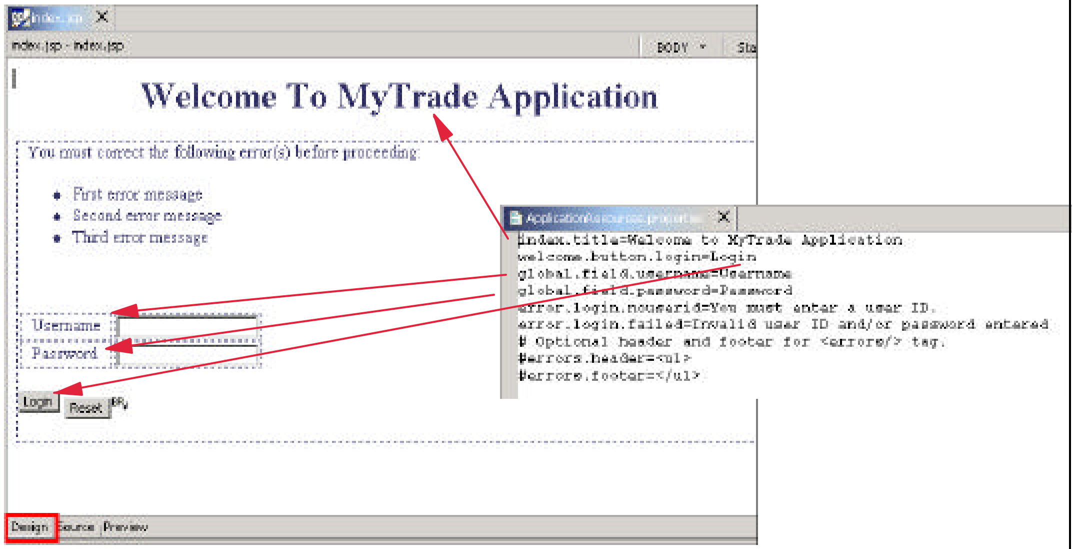
Example: Adding JSPs to Appl (index.jsp)...



```
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>index.jsp</TITLE>
</HEAD>
<BODY>
<h1 align="center"><bean:message key="index.title"/></h1>
<html:form action="/loginAction">
<html:errors/>
<p>
<table>
  <tr>
    <td><bean:message key="global.field.username"/></td>
    <td><html:text property="username" size="20" maxlength="30"/></td>
  </tr>
  <tr>
    <td><bean:message key="global.field.password"/></td>
    <td><html:password property="password" size="20" maxlength="30"/></td>
  </tr>
</table>
<p>
<html:submit><bean:message key="welcome.button.login"/>
</html:submit>
<input type="reset"><br>
</html:form>

</BODY>
</html:html>
```

WebSphere software

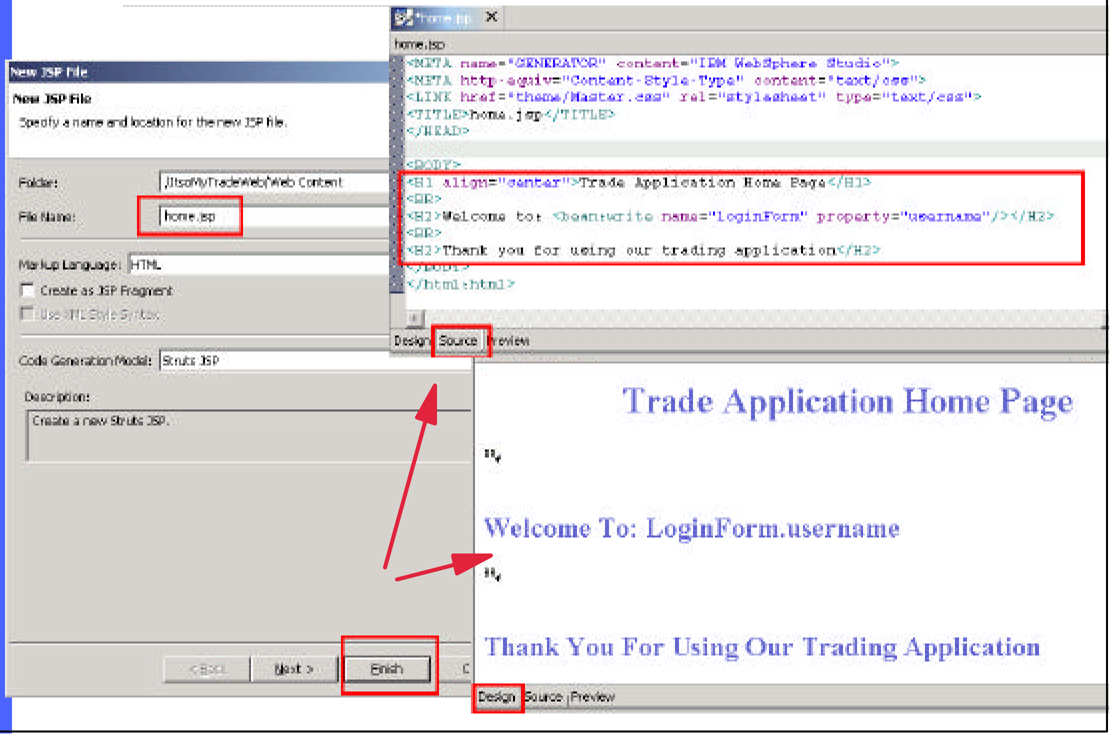
Example: Adding JSPs to Appl (index.jsp)...



```
index.title=Welcome to MyTrade Application
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a user ID.
error.login.failed=Invalid user ID and/or password entered
# Optional header and footer for <errors/> tag.
#errors.header=<ul>
#errors.footer=</ul>
```

WebSphere software

Example: Adding JSPs to Appl (home.jsp)...



New JSP File

Specify a name and location for the new JSP file.

Folder: /ItsoMyTradeWeb/Web Content

File Name: home.jsp

Markup Language: HTML

☐ Create as JSP Fragment

☐ Use XML Style Syntax

Code Generation Model: Struts JSP

Description: Create a new Struts JSP.

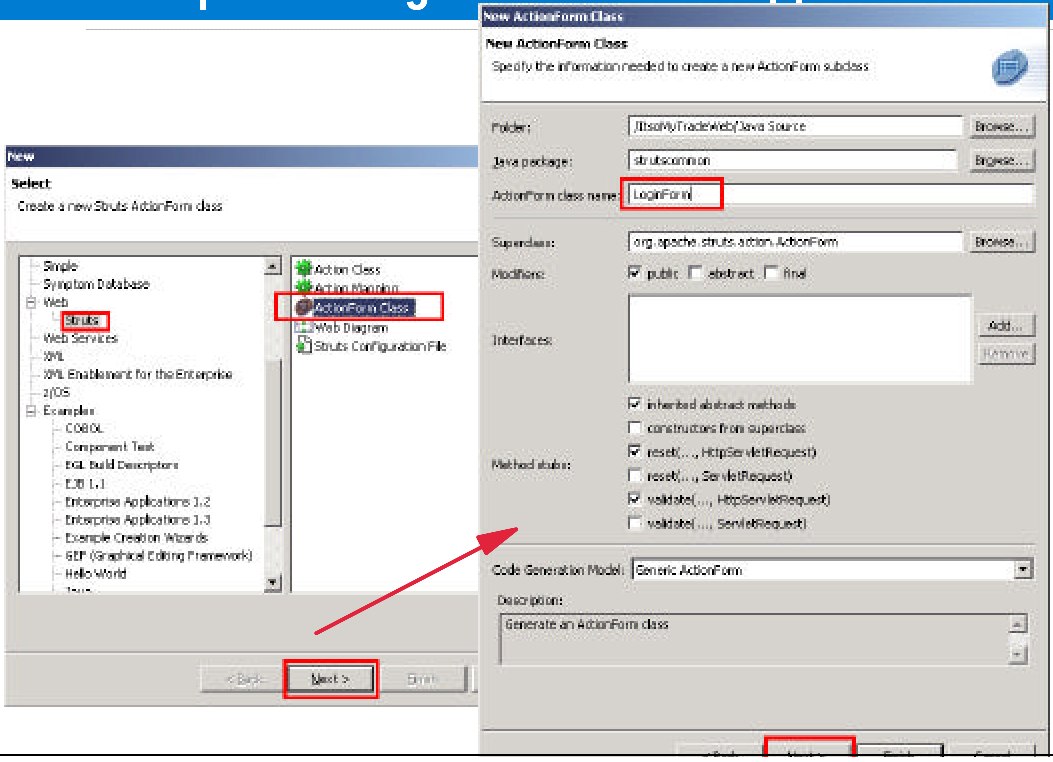
```
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet" type="text/css">
<TITLE>home.jsp</TITLE>
</HEAD>

<BODY>
<H1 align="center">Trade Application Home Page</H1>
<BR>
<H2>Welcome to: <bean:write name="loginForm" property="username"/></H2>
<BR>
<H2>Thank you for using our trading application</H2>
</BODY>
</html:html>
```

Trade Application Home Page

Welcome To: LoginForm.username

Thank You For Using Our Trading Application

Example: Adding actionForm to Appl...



New

Select

Create a new Struts ActionForm class

New ActionForm Class

Specify the information needed to create a new ActionForm subclass

Folder: /ItsoMyTradeWeb/Java Source

Java package: strutscommon

ActionForm class name: LoginForm

Superclass: org.apache.struts.action.ActionForm

Modifiers: [✓ public] ☐ abstract ☐ final

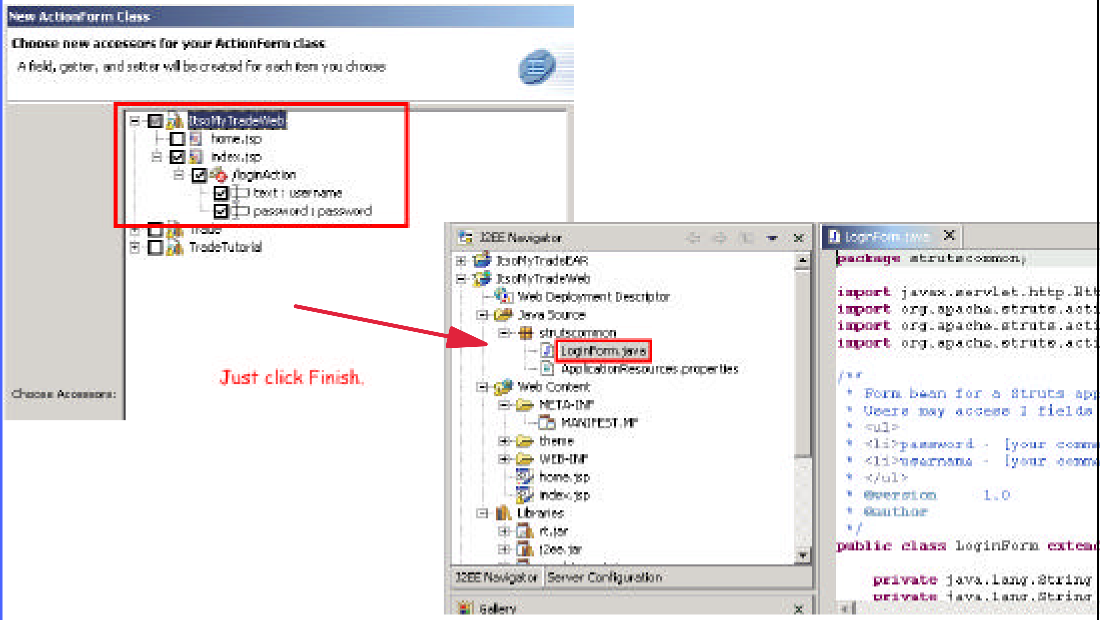
Interfaces:

Method stubs:
[✓ inherited abstract methods]
☐ constructors from superclass
[✓ reset(..., HttpServletRequest)]
☐ reset(..., ServletRequest)
[✓ validate(..., HttpServletRequest)]
☐ validate(..., ServletRequest)

Code Generation Model: Generic ActionForm

Description: Generate an ActionForm class

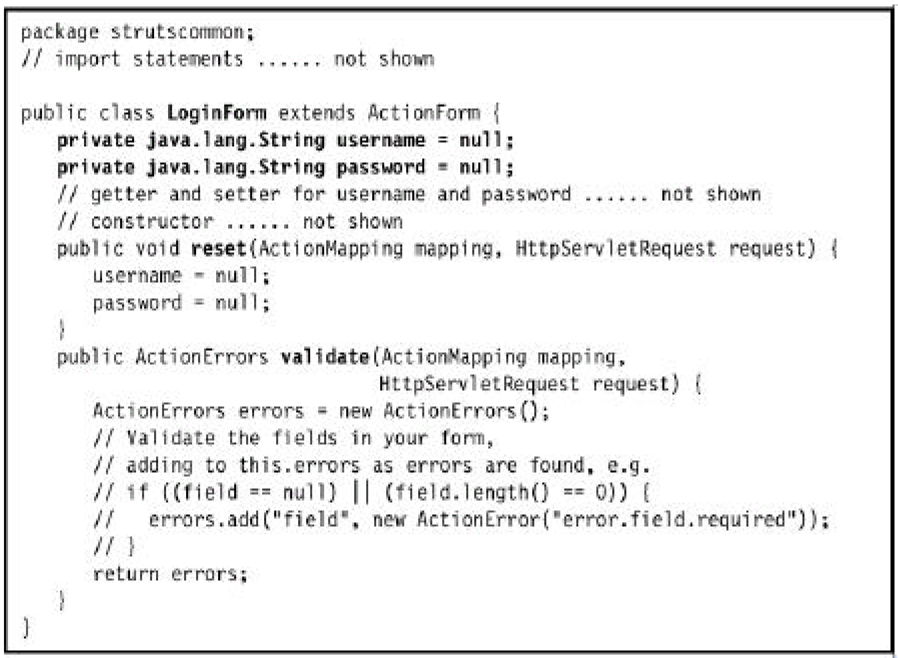
Example: Adding actionForm to Appl...



```
package strutscommon;

import javax.servlet.http.Htt
import org.apache.struts.acti
import org.apache.struts.acti
import org.apache.struts.acti

/**
 * Form bean for a Struts app
 * Users may access 2 fields
 * <ul>
 * <li>password - [your comme
 * <li>username - [your comme
 * </ul>
 * @version     1.0
 * @author
 */
public class LoginForm extend

    private java.lang.String
    private java.lang.String
```

WebSphere software

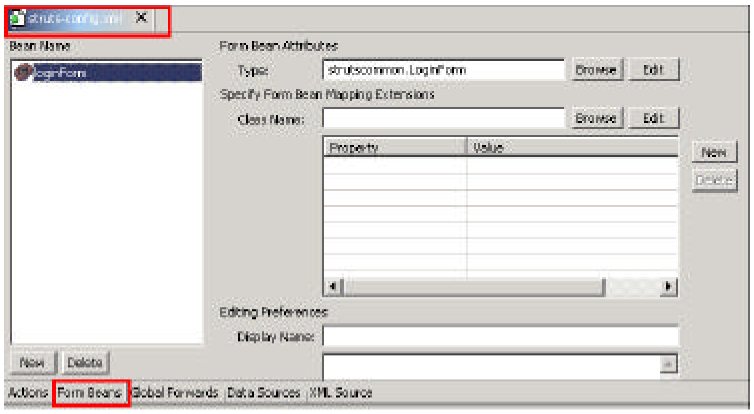
Example: Adding actionForm to Appl...



```
package strutscommon;
// import statements ...... not shown

public class LoginForm extends ActionForm {
    private java.lang.String username = null;
    private java.lang.String password = null;
    // getter and setter for username and password ...... not shown
    // constructor ...... not shown
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        username = null;
        password = null;
    }
    public ActionErrors validate(ActionMapping mapping,
                                    HttpServletRequest request) {
        ActionErrors errors = new ActionErrors();
        // Validate the fields in your form,
        // adding to this.errors as errors are found, e.g.
        // if ((field == null) || (field.length() == 0)) {
        //   errors.add("field", new ActionError("error.field.required"));
        // }
        return errors;
    }
}
```

Figure 6-15 Generated code for the LoginForm class (abbreviated)

Example: Results of adding actionForm ...


struts-config.xml

Bean Name: loginForm

Form Bean Attributes

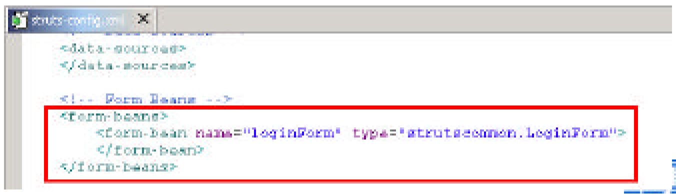
Type: strutscommon.LoginForm

Specify Form Bean Mapping Extensions

Class Name:

Editing Preferences

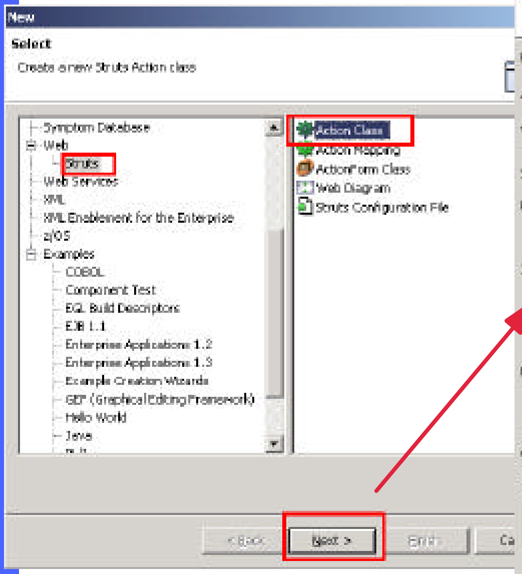
Display Name:


struts-config.xml

```
<data-sources>
</data-sources>

<!-- Form Beans -->
<form-beans>
    <form-bean name="loginForm" type="strutscommon.LoginForm">
    </form-bean>
</form-beans>
```

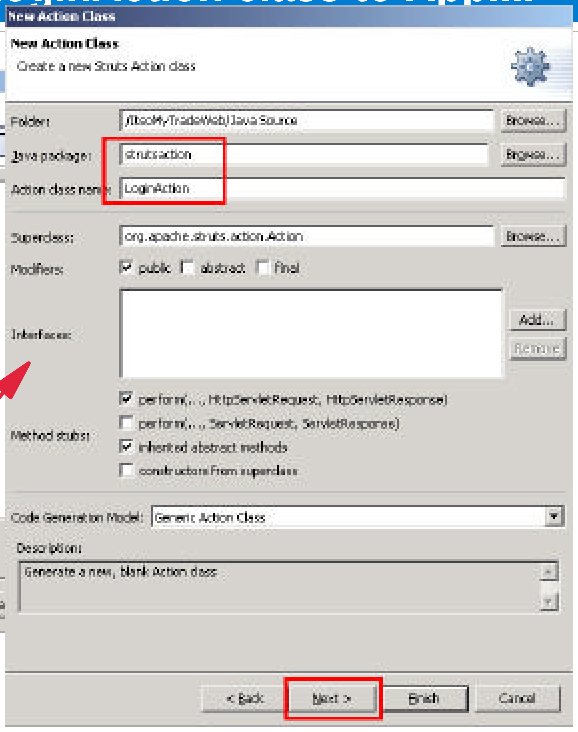
WebSphere software

Example: Adding LoginAction class to Appl...


New

Select

Create a new Struts Action class


New Action Class

New Action Class

Create a new Struts Action class

Folder: /ItsoMyTradeWeb/Java Source

Java package: strutsaction

Action class name: LoginAction

Superclass: org.apache.struts.action.Action

Modifiers: [✓ public] abstract final

Interfaces:

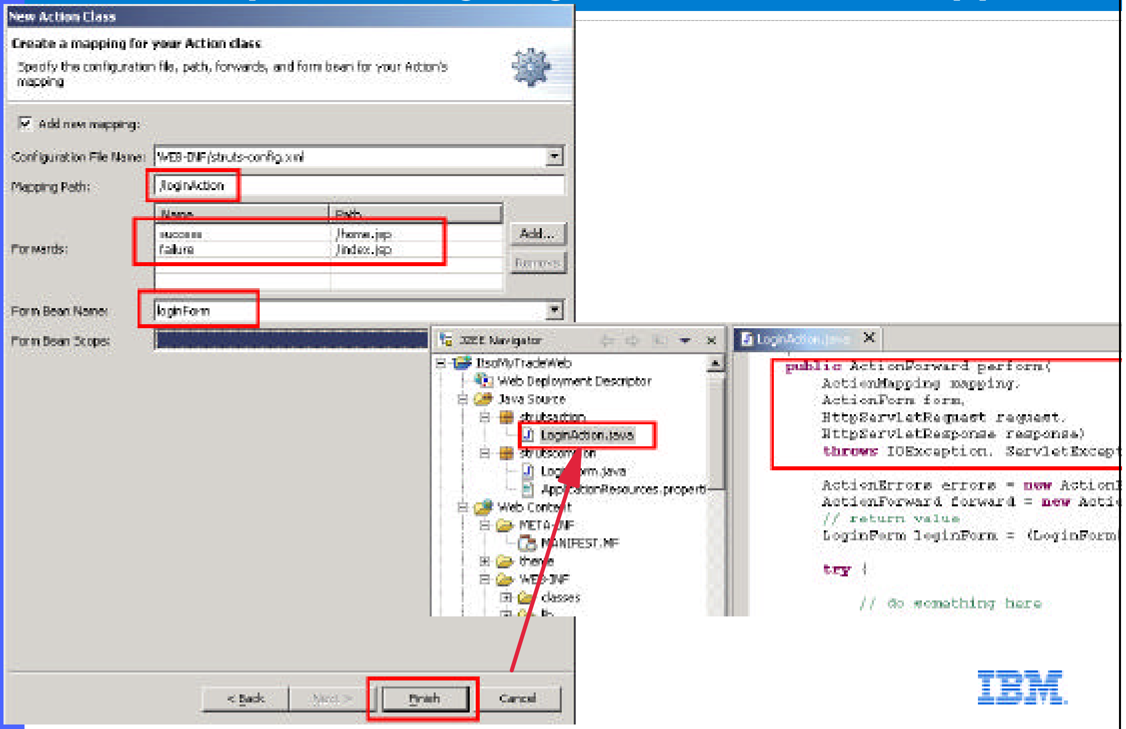
Method stubs: [✓ perform(..., HttpServletRequest, HttpServletResponse)] perform(..., ServletRequest, ServletResponse) [✓ inherited abstract methods] constructors from superclass

Code Generation Model: Generic Action Class

Description: Generate a new, blank Action class

WebSphere software

Example: Adding LoginAction class to Appl...


New Action Class

Create a mapping for your Action class

Specify the configuration file, path, forwards, and form bean for your Action's mapping

☑ Add new mapping:

Configuration File Name: WEB-INF/struts-config.xml

Mapping Path: /loginAction

Forwards:

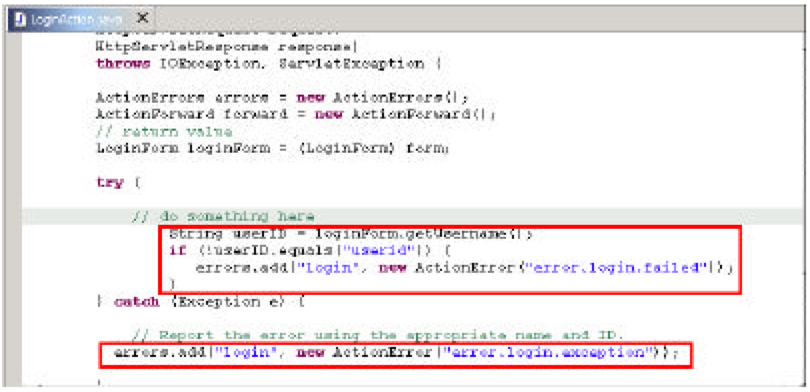
Name	Path
success	/home.jsp
failure	/index.jsp

Form Bean Name: loginForm

Form Bean Scope:

```
public ActionForward perform(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletExcept

    ActionErrors errors = new ActionE
    ActionForward forward = new Actio
    // return value
    LoginForm loginForm = (LoginForm)

    try {

        // do something here
```

Example: Customizing LoginAction class ...



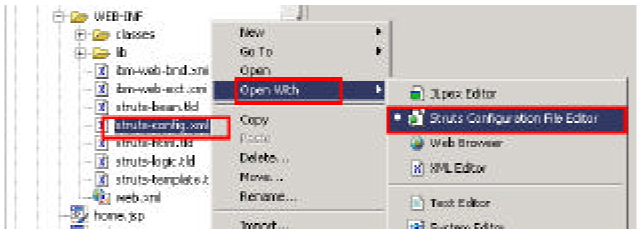
```
    HttpServletResponse response)
    throws IOException, ServletException {

    ActionErrors errors = new ActionErrors();
    ActionForward forward = new ActionForward();
    // return value
    LoginForm loginForm = (LoginForm) form;

    try {

        // do something here
        String userID = loginForm.getUsername();
        if (!userID.equals("userid")) {
            errors.add("login", new ActionError("error.login.failed"));
        }
    } catch (Exception e) {

        // Report the error using the appropriate name and ID.
        errors.add("login", new ActionError("error.login.exception"));

    }
```

Example: Adding action mappings to Appl...

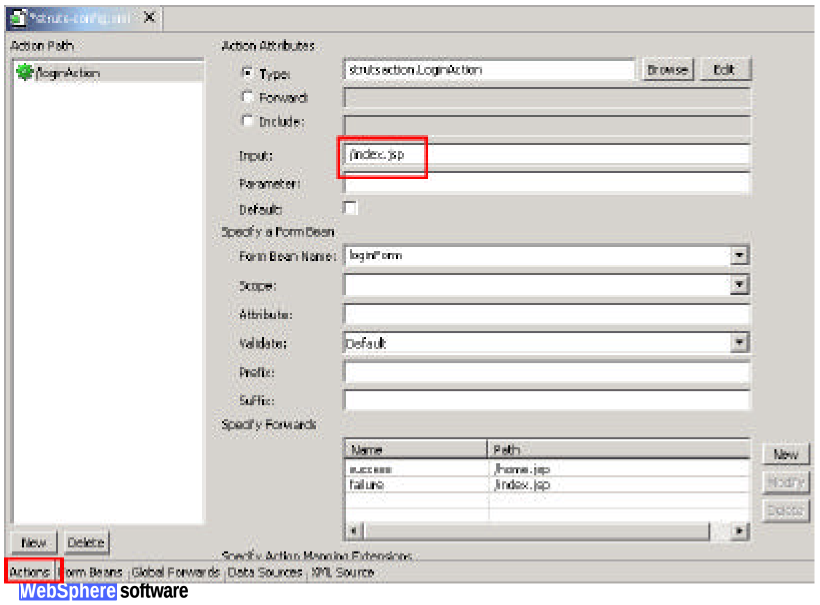
Action mappings

So far we have added JSPs, action forms, and actions to our basic Struts application. We have added action forwards and action errors to our action. We will now tie all those pieces together using action mappings to complete the login portion of our application.

Edit struts-config.xml



Example: Completing Login action ...



Action Path

/loginAction

Action Attributes

[✓ Type:] struts.action.LoginAction

Forward:

Include:

Input: /index.jsp

Parameter:

Default: ☐

Specify a Form Bean

Form Bean Name: loginForm

Scope:

Attribute:

Validate: Default

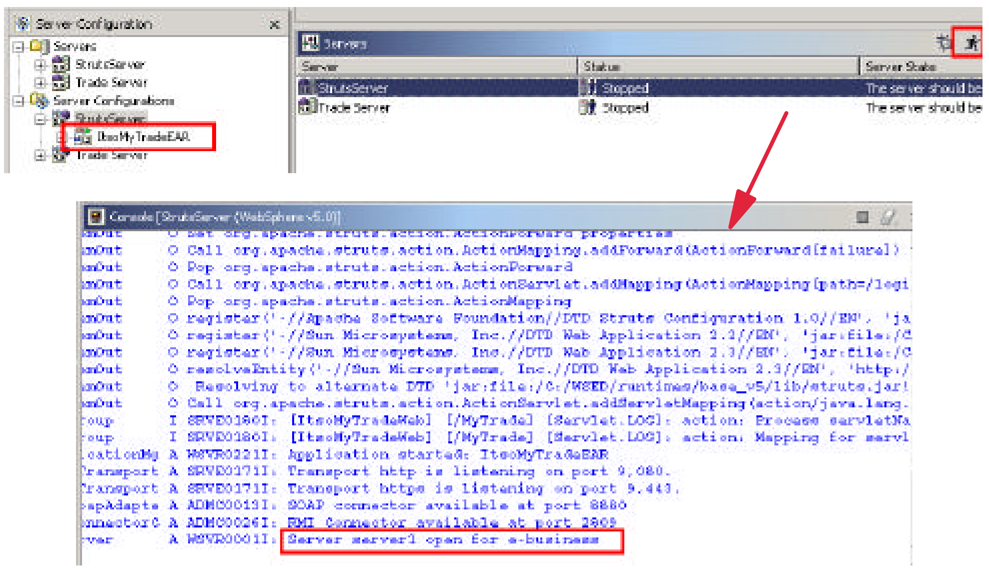
Prefix:

Suffix:

Specify Forwards

Name	Path
success	/home.jsp
failure	/index.jsp

Example: Testing the Appl...



Server Configuration

- Servers
  - StrutsServer
  - Trade Server
- Server Configurations
  - StrutsServer
    - ItsoMyTradeEAR
  - Trade Server

Servers

Server	Status	Server State
StrutsServer	Stopped	The server should be
Trade Server	Stopped	The server should be

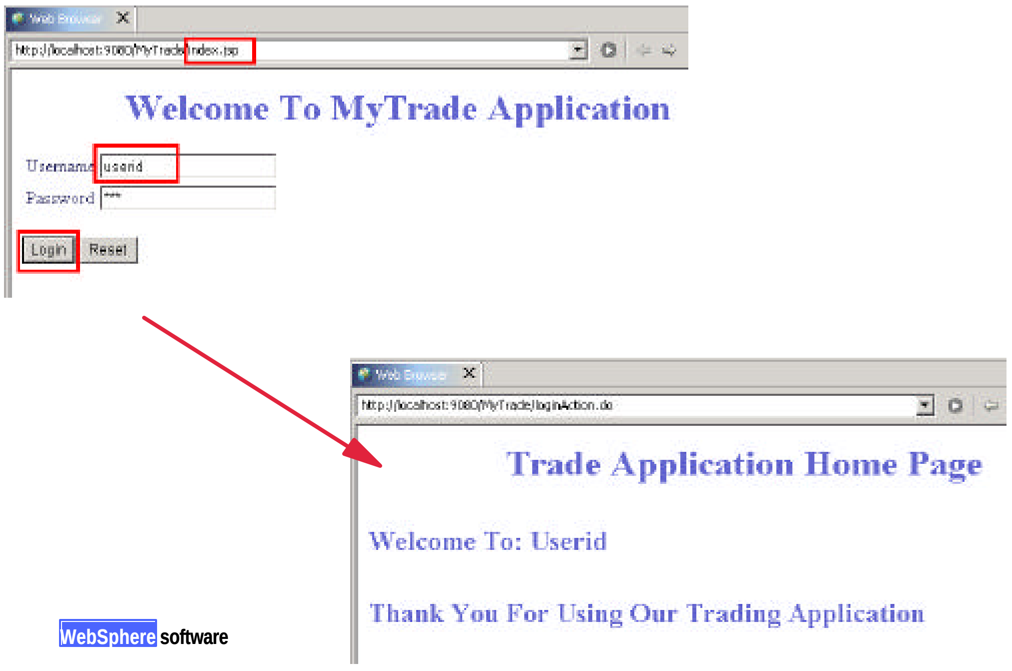
Console [StrutsServer (WebSphere v5.0)]

```
mOut       O Set org.apache.struts.action.ActionForward properties
mOut       O Call org.apache.struts.action.ActionMapping.addForward(ActionForward[failure])
mOut       O Pop org.apache.struts.action.ActionForward
mOut       O Call org.apache.struts.action.ActionServlet.addMapping(ActionMapping[path=/logi
mOut       O Pop org.apache.struts.action.ActionMapping
mOut       O register('-//Apache Software Foundation//DTD Struts Configuration 1.0//EN', 'ja
mOut       O register('-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN', 'jar:file:/C
mOut       O register('-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN', 'jar:file:/C
mOut       O resolveEntity('-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN', 'http:/
mOut       O  Resolving to alternate DTD 'jar:file:/C:/WSED/runtimes/base_v5/lib/struts.jar!
mOut       O Call org.apache.struts.action.ActionServlet.addServletMapping(action/java.lang.
oup        I SRVE0180I: [ItsoMyTradeWeb] [/MyTrade] [Servlet.LOG]: action: Process servletNa
oup        I SRVE0180I: [ItsoMyTradeWeb] [/MyTrade] [Servlet.LOG]: action: Mapping for servl
cationMg A WSVR0221I: Application started: ItsoMyTradeEAR
ransport A SRVE0171I: Transport http is listening on port 9,080.
ransport A SRVE0171I: Transport https is listening on port 9,443.
apAdapte A ADMC0013I: SOAP connector available at port 8880
nnectorC A ADMC0026I: RMI Connector available at port 2809
ver        A WSVR0001I: Server server1 open for e-business
```

WebSphere software

IBM

Example: Testing the Appl...


Web Browser
http://localhost:9080/MyTrade/index.jsp

Welcome To MyTrade Application

Username	userid
Password	***

Login Reset

Web Browser
http://localhost:9080/MyTrade/loginAction.do

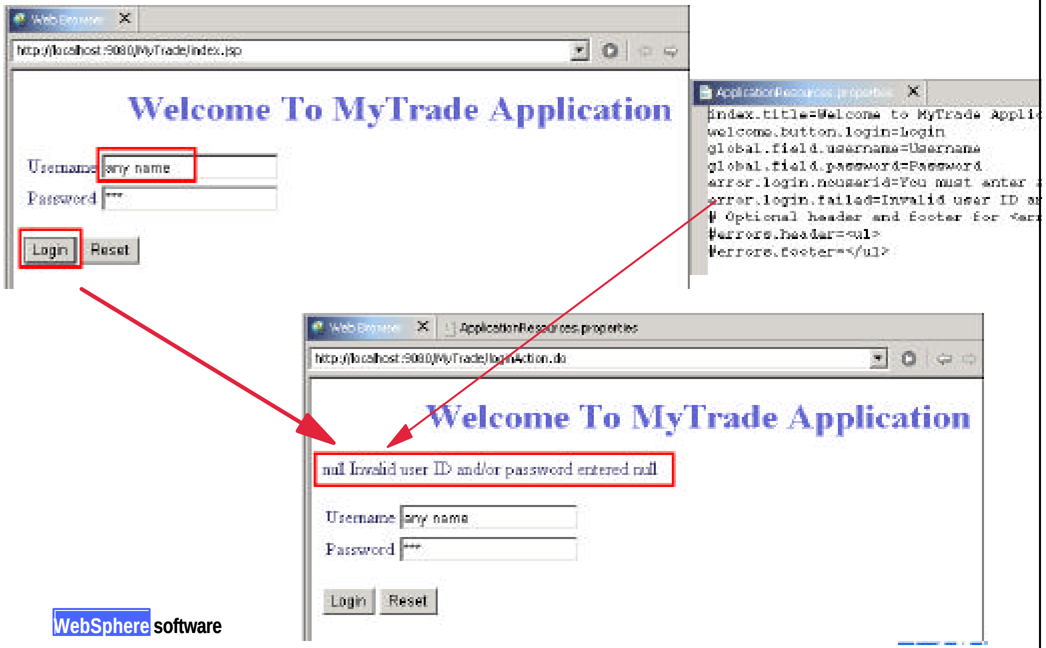
Trade Application Home Page

Welcome To: Userid

Thank You For Using Our Trading Application

WebSphere software

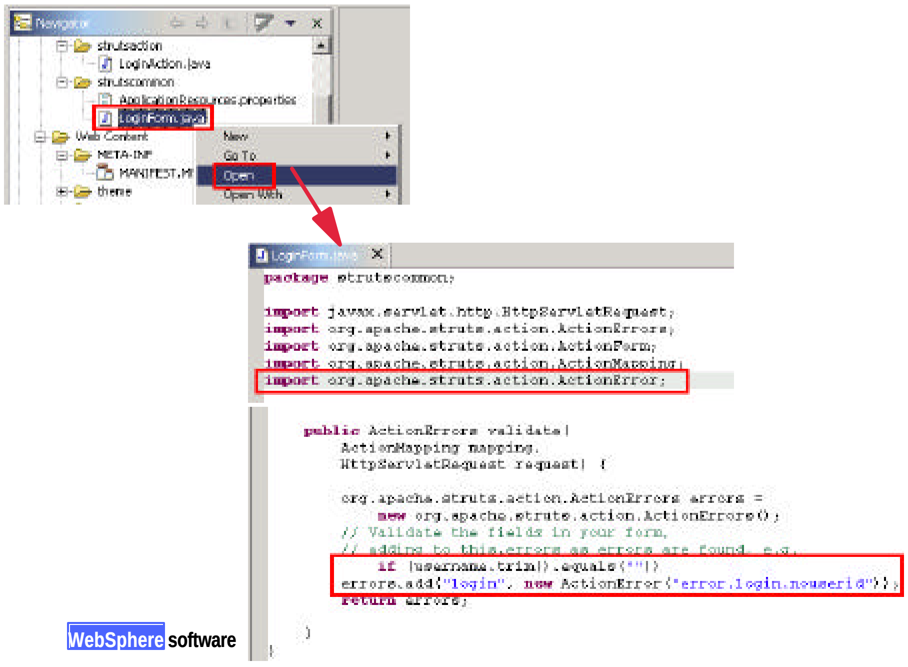
Example: Testing the Appl...


Web Browser
http://localhost:9080/MyTrade/index.jsp
Welcome To MyTrade Application
Username
any name
Password
Login
Reset
ApplicationResources.properties
http://localhost:9080/MyTrade/loginAction.do
null Invalid user ID and/or password entered null

```
index.title=Welcome to MyTrade Applic
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a
error.login.failed=Invalid user ID an
# Optional header and footer for <err
#errors.header=<ul>
#errors.footer=</ul>
```

WebSphere software

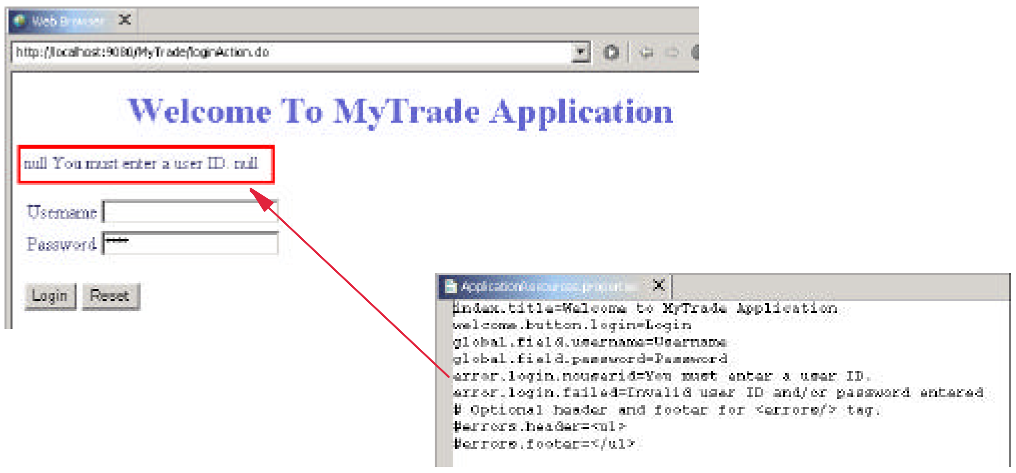
Example: Implementing simple validation...



```
package strutscommon;

import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import org.apache.struts.action.ActionError;


    public ActionErrors validate(
        ActionMapping mapping,
        HttpServletRequest request) {

        org.apache.struts.action.ActionErrors errors =
            new org.apache.struts.action.ActionErrors();
        // Validate the fields in your form,
        // adding to this.errors as errors are found, e.g.
            if (username.trim().equals(""))
        errors.add("login", new ActionError("error.login.nouserid"));
        return errors;

    }
}
```

WebSphere software

Example: Test again.. user ID empty..


Web Browser
http://localhost:9080/MyTrade/loginAction.do
Welcome To MyTrade Application
Username
Password
Login
Reset
ApplicationResources.properties

null You must enter a user ID. null

```
index.title=Welcome to MyTrade Application
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a user ID.
error.login.failed=Invalid user ID and/or password entered
# Optional header and footer for <errors/> tag.
#errors.header=<ul>
#errors.footer=</ul>
```

WebSphere software

IBM

Example2 : Using appl diagram editor

Our sample Web application has a simple flow:

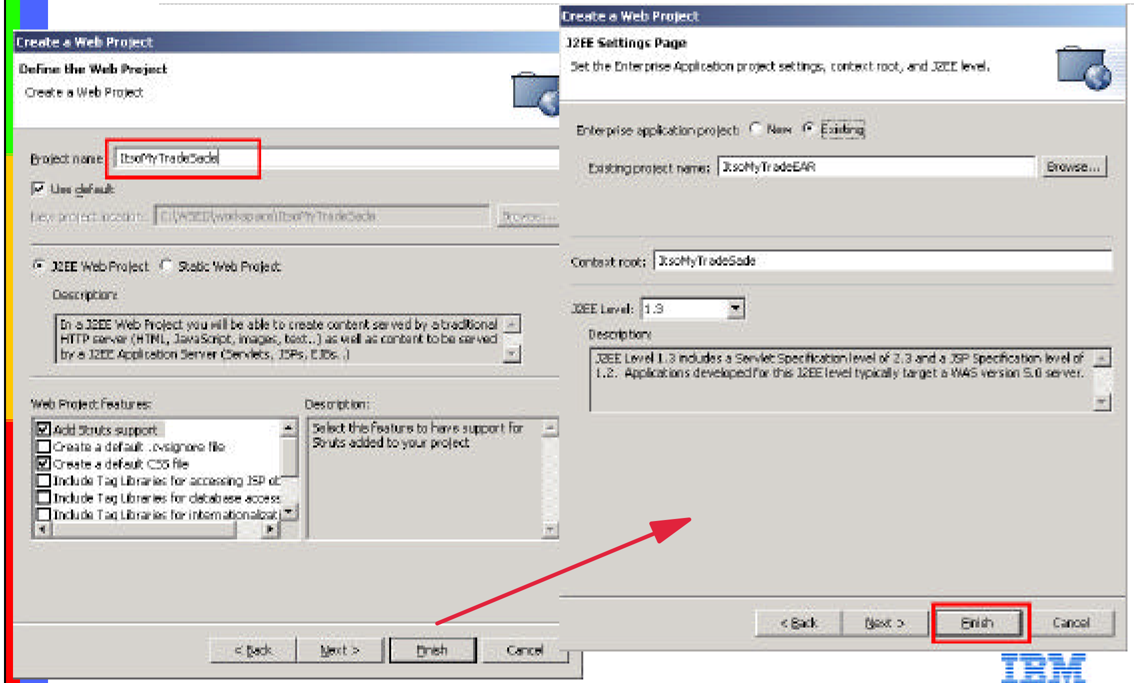
- A welcome page is displayed initially
- A user can enter a user ID and a password on the welcome page and click *Submit*
- The server invokes a Struts action class to verify the user ID and password
- If the authentication is successful, the user can proceed to the home page of the application
- If the authentication fails, an error page is shown and the user can go back to the welcome page

To implement this Web application we use the Struts application diagram editor, from which we can implement the JSPs and the action.

WebSphere software

IBM

Example2 : Create a new Web project



Define the Web Project
Create a Web Project

Project name: ItsoMyTradeSade
[✓] Use default
New project location: C:\WSED\workspace\ItsoMyTradeSade

[✓ J2EE Web Project] Static Web Project

Description:
In a J2EE Web Project you will be able to create content served by a traditional HTTP server (HTML, JavaScript, images, text...) as well as content to be served by a J2EE Application Server (Servlets, JSPs, EJBs..)

Web Project Features:
- [✓] Add Struts support
- [ ] Create a default .cvsignore file
- [✓] Create a default CSS file
- [ ] Include Tag Libraries for accessing JSP ob
- [ ] Include Tag Libraries for database access
- [ ] Include Tag Libraries for internationalizat

Description:
Select this feature to have support for Struts added to your project

J2EE Settings Page
Set the Enterprise Application project settings, context root, and J2EE level.

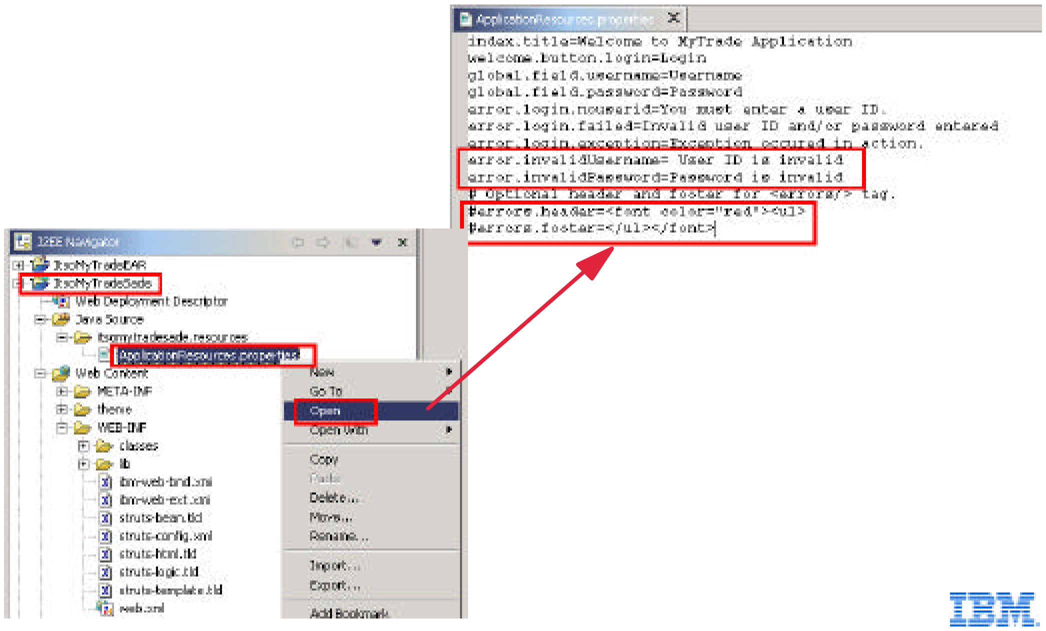
Enterprise application project: New [✓ Existing]
Existing project name: ItsoMyTradeEAR

Context root: ItsoMyTradeSade

J2EE Level: 1.3

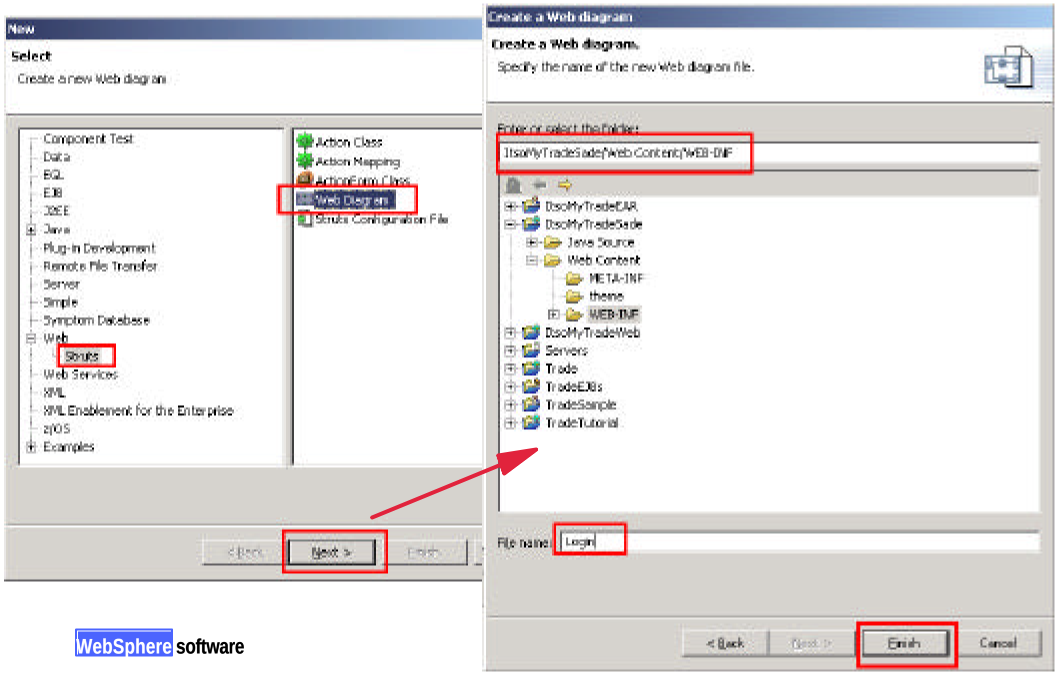
Description:
J2EE Level 1.3 includes a Servlet Specification level of 2.3 and a JSP Specification level of 1.2. Applications developed for this J2EE level typically target a WAS version 5.0 server.

Example2 : Change the appl resources..



```
index.title=Welcome to MyTrade Application
welcome.button.login=Login
global.field.username=Username
global.field.password=Password
error.login.nouserid=You must enter a user ID.
error.login.failed=Invalid user ID and/or password entered
error.login.exception=Exception occured in action.
error.invalidUsername= User ID is invalid
error.invalidPassword=Password is invalid
# Optional header and footer for <errors/> tag.
#errors.header=<font color="red"><ul>
#errors.footer=</ul></font>
```

Example2 : Create the Struts appl diagram file


New

Select

Create a new Web diagram

Create a Web diagram

Create a Web diagram.

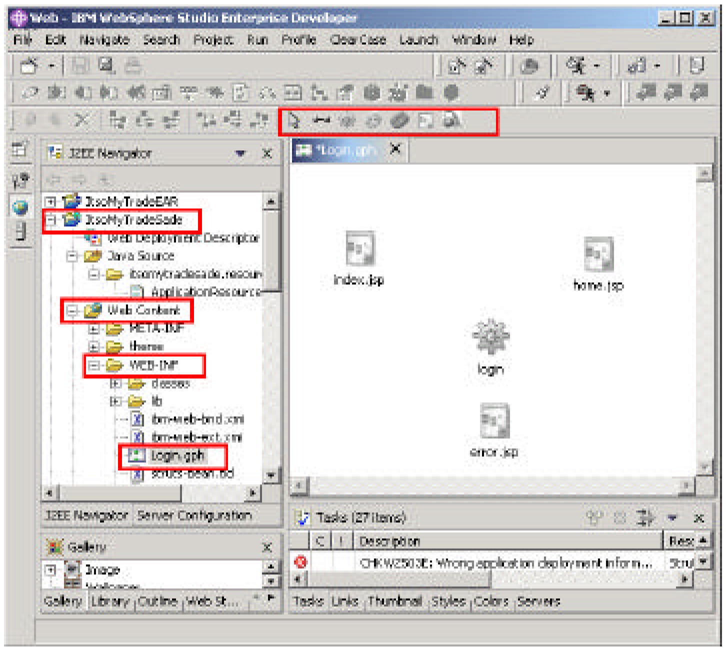
Specify the name of the new Web diagram file.

Enter or select the folder: ItsoMyTradeSade/Web Content/WEB-INF

File name: Login

WebSphere software

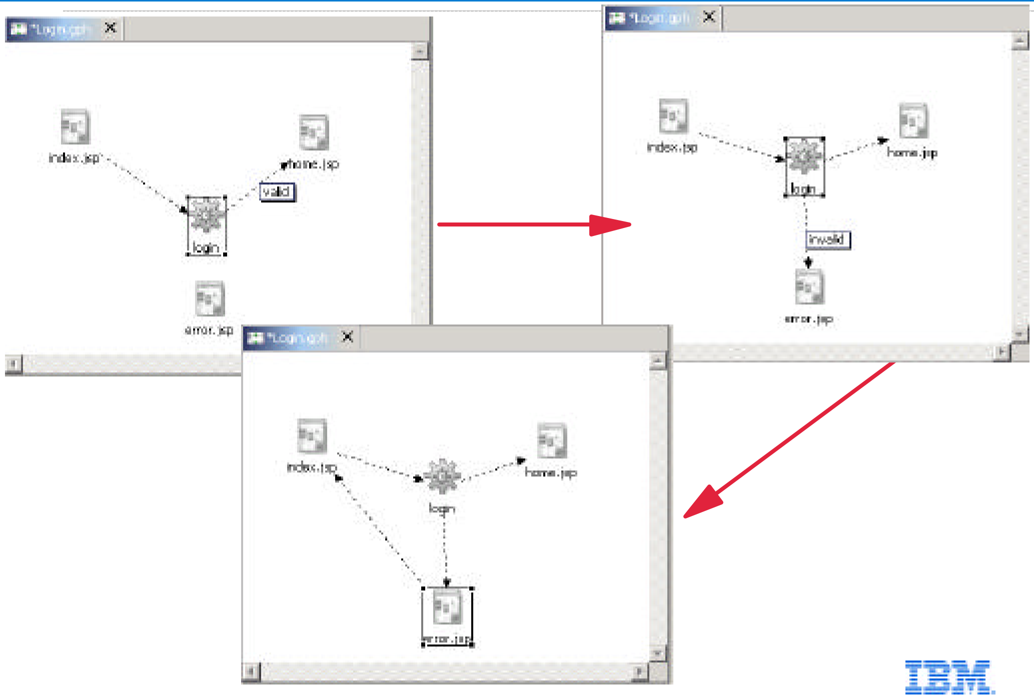
Example2 : Using appl diagram editor...



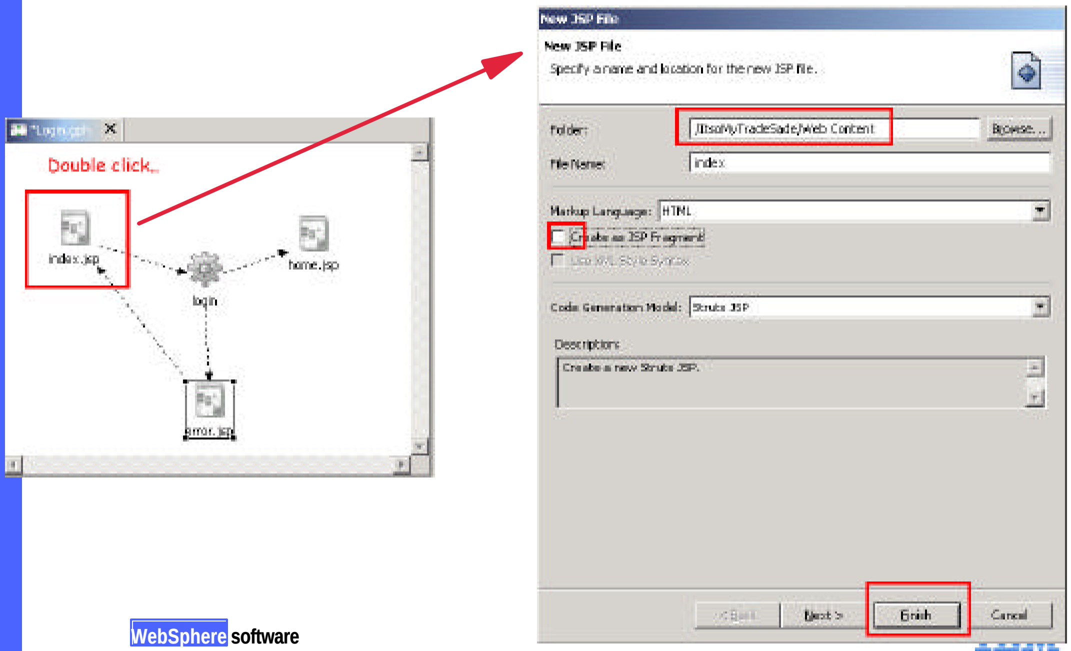
CHKW2503E: Wrong application deployment inform...

IBM

Example2 : Using appl diagram editor...



Example2 : Implementing index JSP page



New JSP File

New JSP File

Specify a name and location for the new JSP file.

Folder: /ItsoMyTradeSade/Web Content

File Name: index

Markup Language: HTML

[ ] Create as JSP Fragment

[ ] Use XML Style Syntax

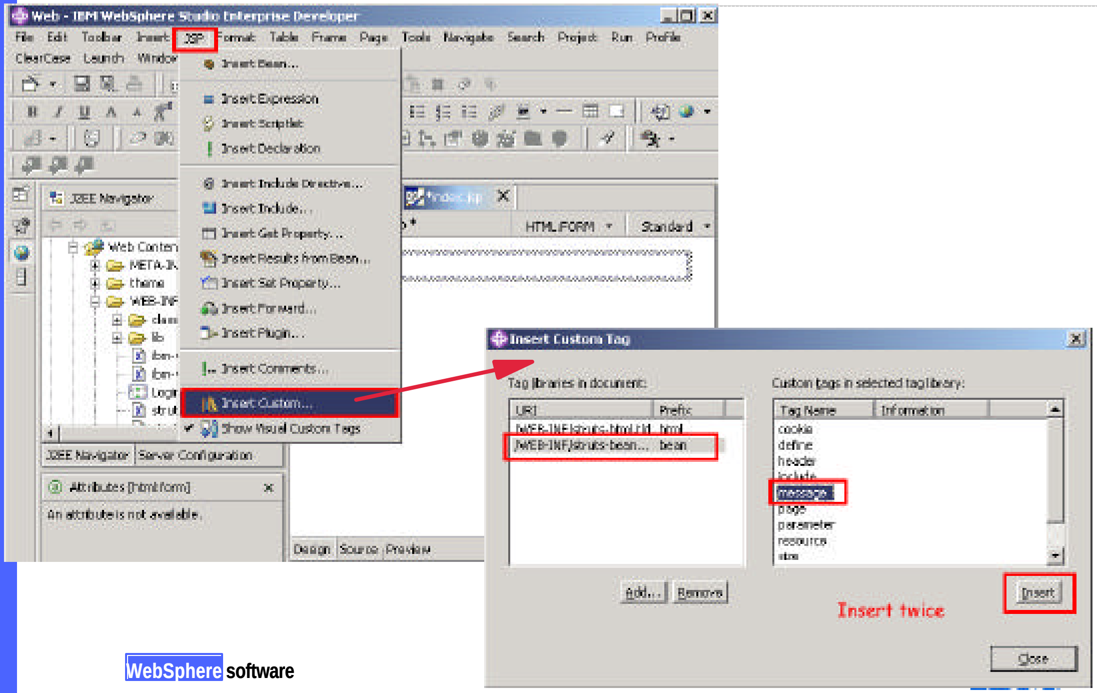
Code Generation Model: Struts JSP

Description:

Create a new Struts JSP.

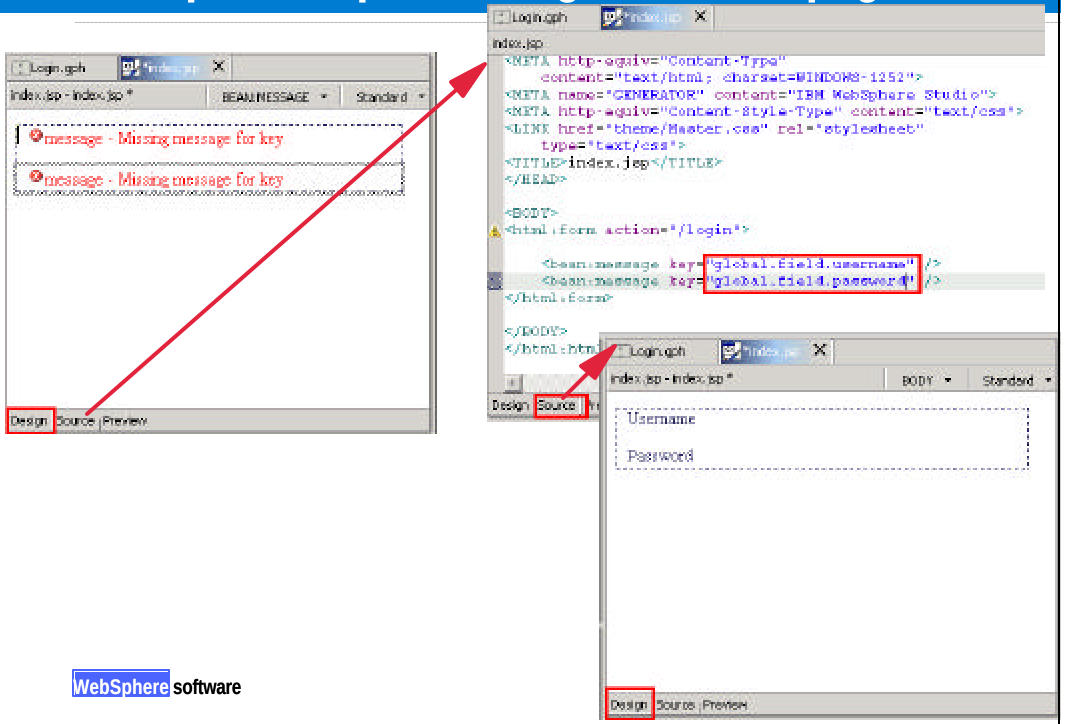
WebSphere software

Example2 : Implementing index JSP page...



WebSphere software

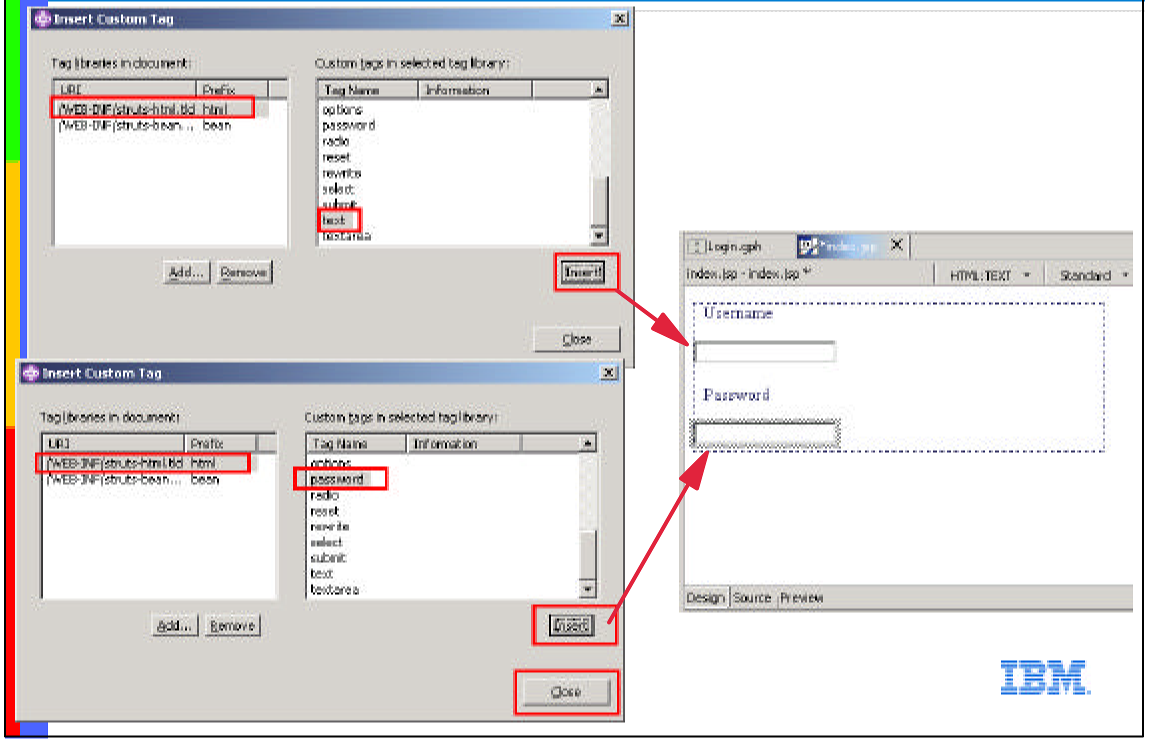
Example2 : Implementing index JSP page...



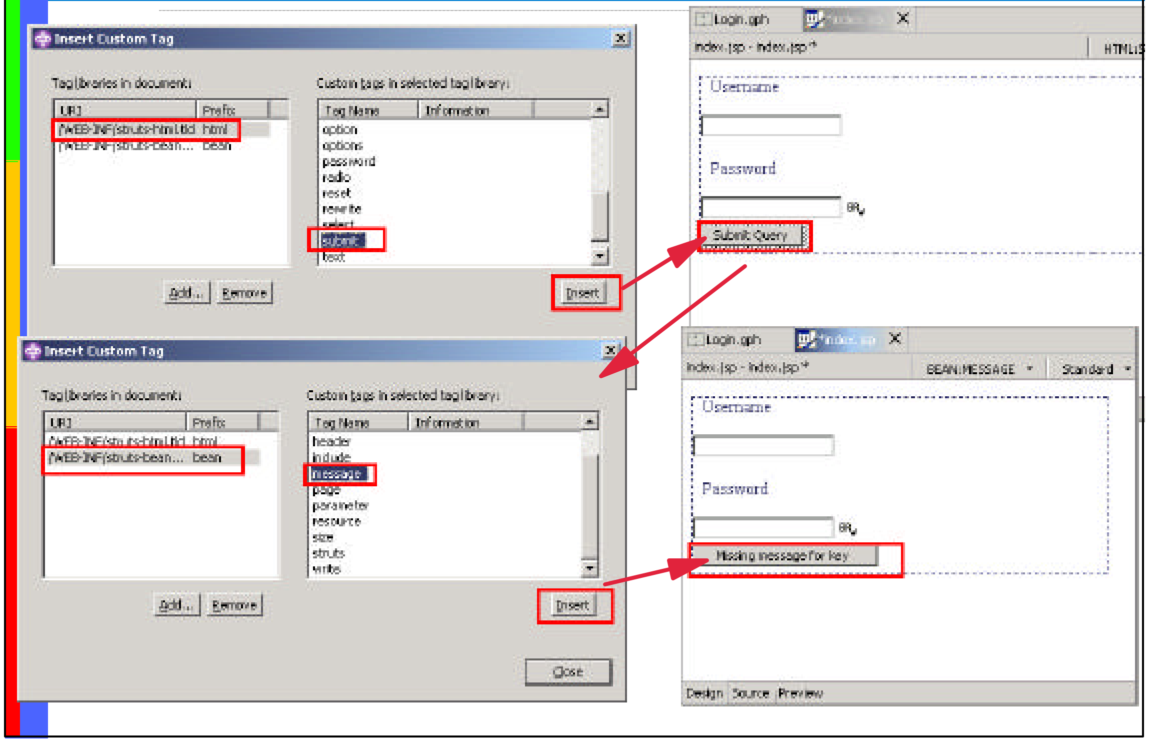
```
<META http-equiv="Content-Type"
    content="text/html; charset=WINDOWS-1252">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet"
    type="text/css">
<TITLE>index.jsp</TITLE>
</HEAD>

<BODY>
<html:form action="/login">

    <bean:message key="global.field.username" />
    <bean:message key="global.field.password" />
</html:form>

</BODY>
</html:html>
```

WebSphere software

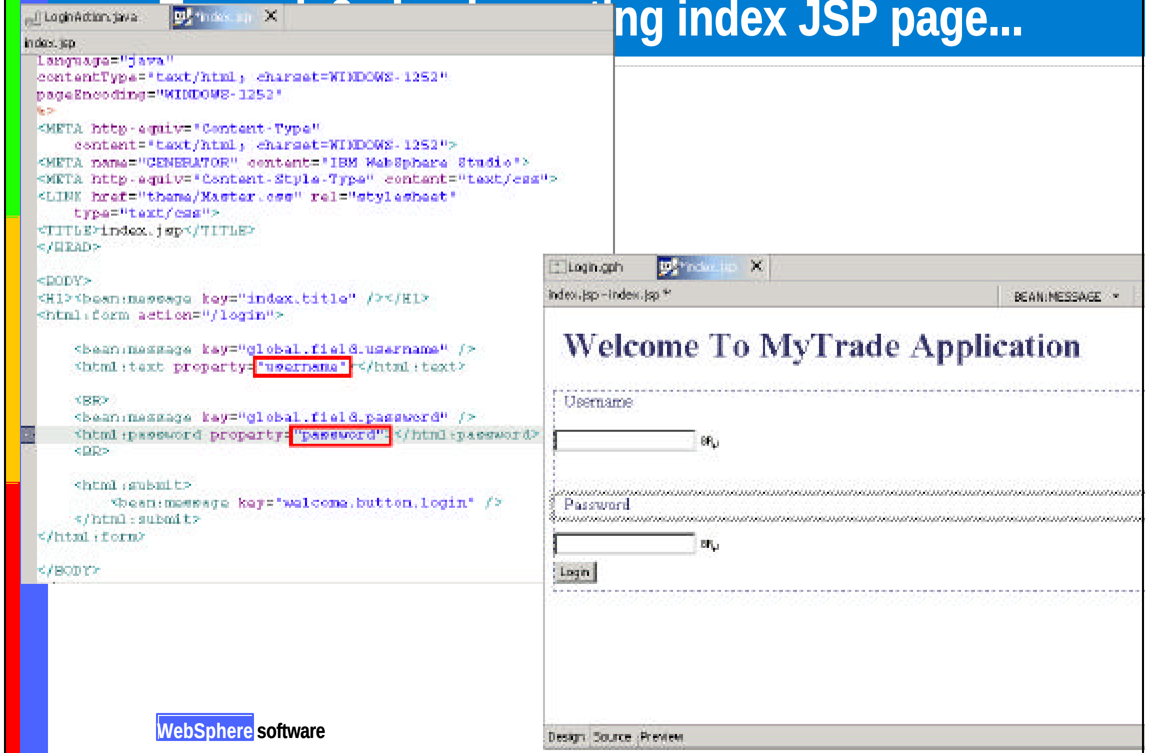
Example2 : Implementing index JSP page…



Example2 : Implementing index JSP page…



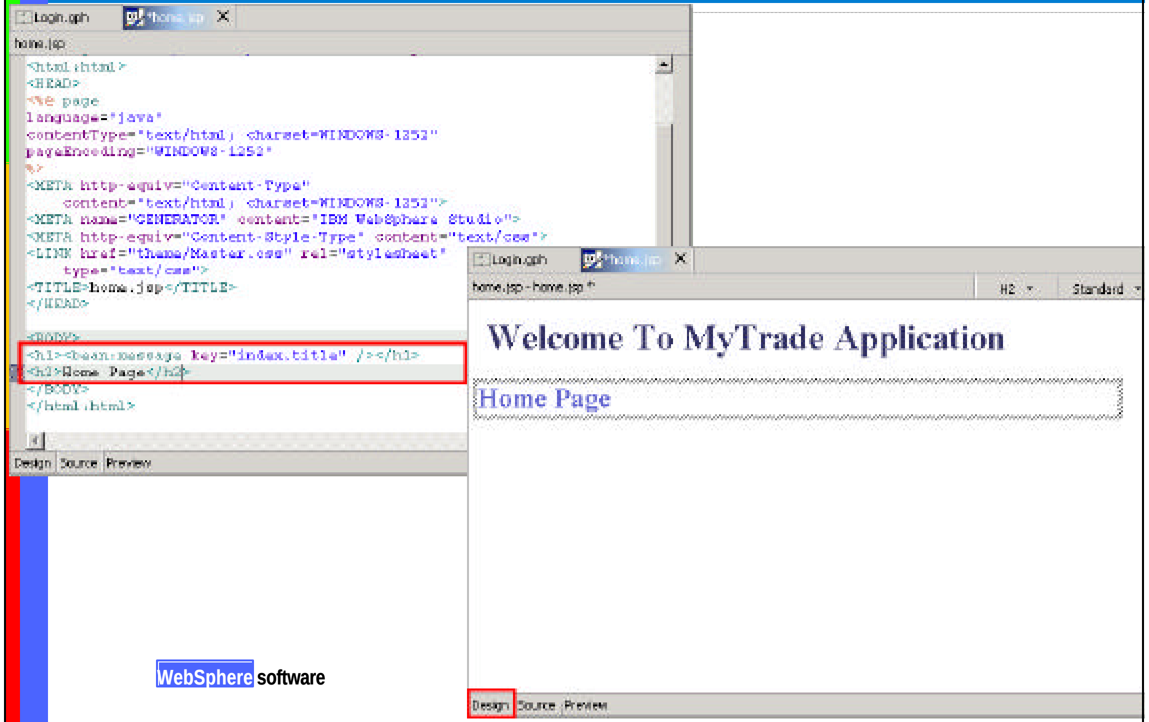
ng index JSP page...



```
language="java"
contentType="text/html; charset=WINDOWS-1252"
pageEncoding="WINDOWS-1252"
%>
<META http-equiv="Content-Type"
    content="text/html; charset=WINDOWS-1252">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet"
    type="text/css">
<TITLE>index.jsp</TITLE>
</HEAD>

<BODY>
<H1><bean:message key="index.title" /></H1>
<html:form action="/login">

    <bean:message key="global.field.username" />
    <html:text property="username"></html:text>

    <BR>
    <bean:message key="global.field.password" />
    <html:password property="password"></html:password>
    <BR>

    <html:submit>
        <bean:message key="welcome.button.login" />
    </html:submit>
</html:form>

</BODY>
```

WebSphere software

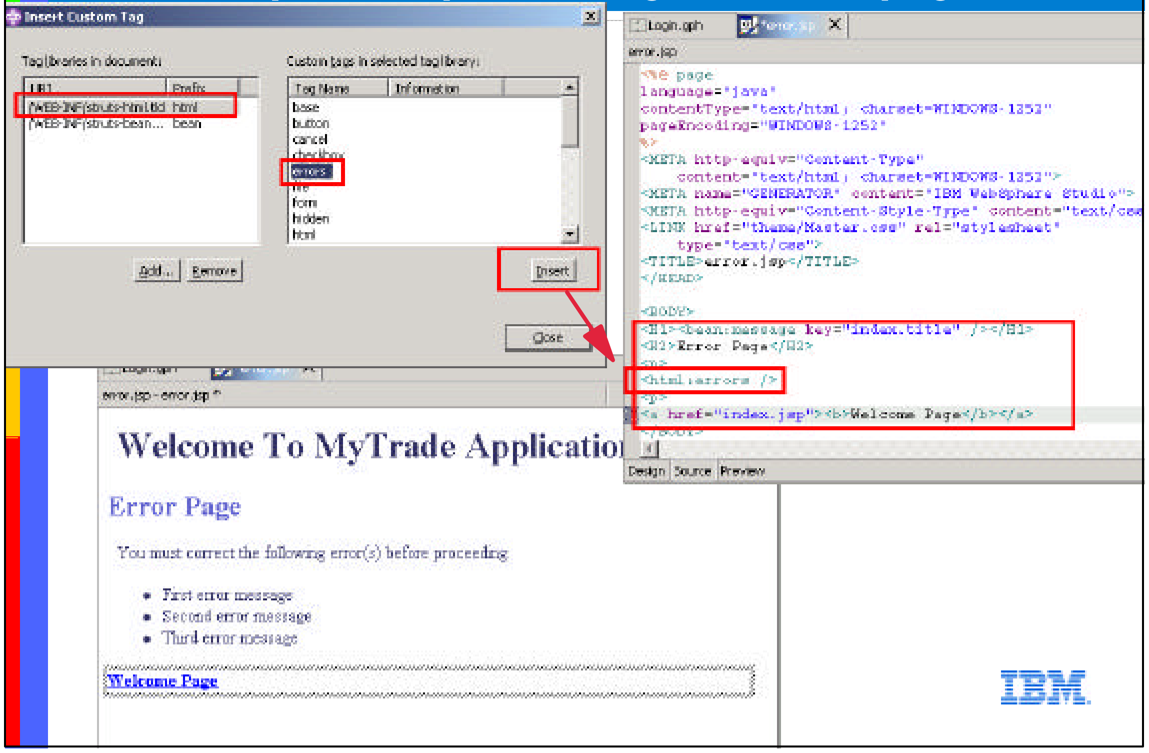
Example2 : Implementing home JSP page



```
<html:html>
<HEAD>
<%@ page
language="java"
contentType="text/html; charset=WINDOWS-1252"
pageEncoding="WINDOWS-1252"
%>
<META http-equiv="Content-Type"
    content="text/html; charset=WINDOWS-1252">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css">
<LINK href="theme/Master.css" rel="stylesheet"
    type="text/css">
<TITLE>home.jsp</TITLE>
</HEAD>

<BODY>
<h1><bean:message key="index.title" /></h1>
<h2>Home Page</h2>
</BODY>
</html:html>
```

WebSphere software

Example2 : Implementing error JSP page



Insert Custom Tag

Tag libraries in document:

URI	Prefix
/WEB-INF/struts-html.tld	html
/WEB-INF/struts-bean...	bean

Custom tags in selected tag library:

Tag Name	Information
base	
button	
cancel	
checkbox	
errors	
file	
form	
hidden	
html	

Add... Remove Insert Close

Login.gph error.jsp

error.jsp

```
<%@ page
language="java"
contentType="text/html; charset=WINDOWS-1252"
pageEncoding="WINDOWS-1252"
%>
<META http-equiv="Content-Type"
    content="text/html; charset=WINDOWS-1252">
<META name="GENERATOR" content="IBM WebSphere Studio">
<META http-equiv="Content-Style-Type" content="text/css
<LINK href="theme/Master.css" rel="stylesheet"
    type="text/css">
<TITLE>error.jsp</TITLE>
</HEAD>

<BODY>
<H1><bean:message key="index.title" /></H1>
<H2>Error Page</H2>
<p>
<html:errors />
<p>
<a href="index.jsp"><b>Welcome Page</b></a>
</BODY>
```

Design Source Preview

error.jsp - error.jsp *

Welcome To MyTrade Applicatio

Error Page

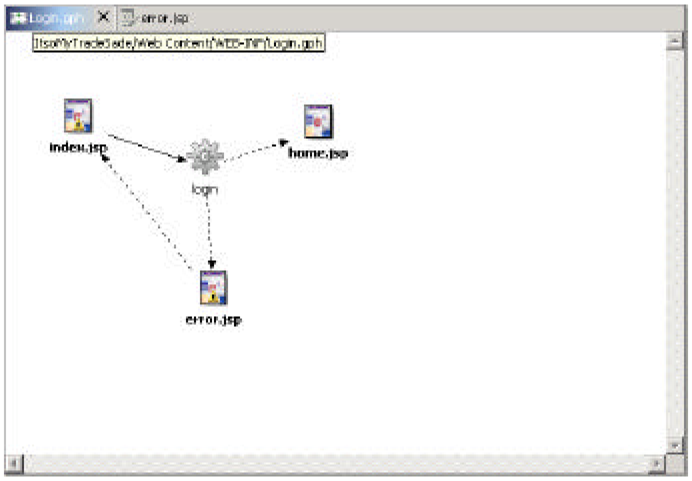
You must correct the following error(s) before proceeding

- First error message
- Second error message
- Third error message

Welcome Page

IBM

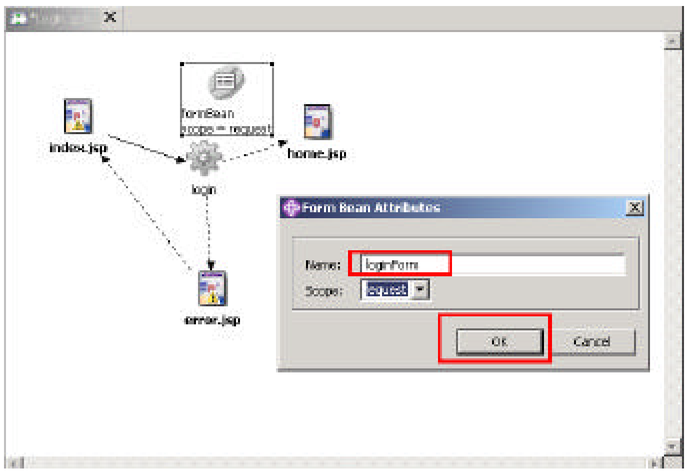
Example2 : Diagram with JSP's realized



WebSphere software

IBM

Example2 : Creating the form bean



Form Bean Attributes

Name: loginForm

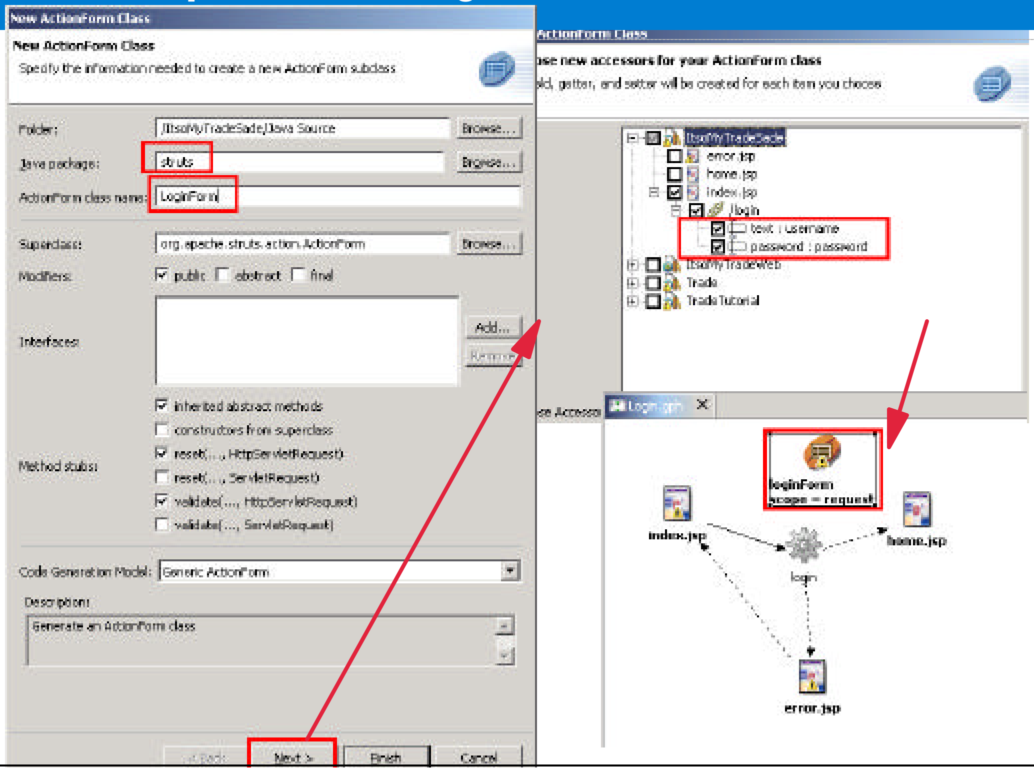
Scope: request

Just double click on formBean....

WebSphere software

IBM

Example2 : Creating the form bean.....



New ActionForm Class

New ActionForm Class

Specify the information needed to create a new ActionForm subclass

Folder: /ItsoMyTradeSade/Java Source

Java package: struts

ActionForm class name: LoginForm

Superclass: org.apache.struts.action.ActionForm

Modifiers: [✓ public] abstract final

Interfaces:

Method stubs: [✓ inherited abstract methods] constructors from superclass [✓ reset(..., HttpServletRequest)] reset(..., ServletRequest) [✓ validate(..., HttpServletRequest)] validate(..., ServletRequest)

Code Generation Model: Generic ActionForm

Description:

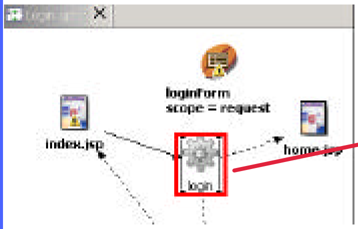
Generate an ActionForm class

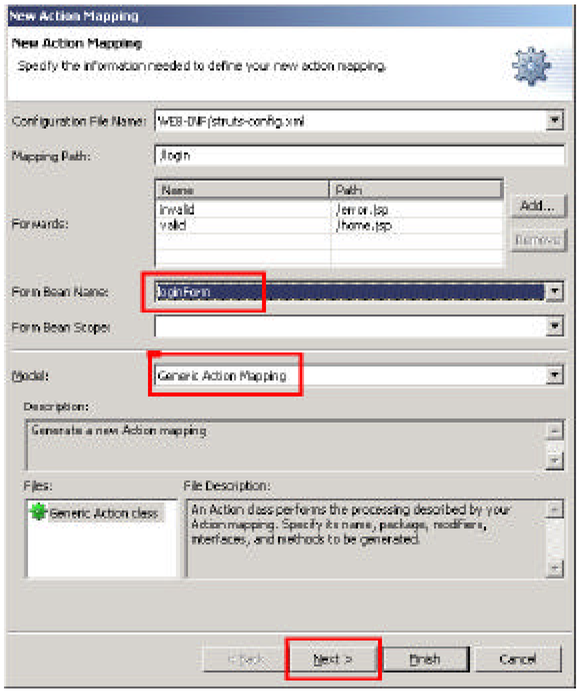
ActionForm Class

ose new accessors for your ActionForm class

eld, getter, and setter will be created for each item you choose

Example2 : Creating action Mapping




New Action Mapping

New Action Mapping

Specify the information needed to define your new action mapping.

Configuration File Name: WEB-INF/struts-config.xml

Mapping Path: /login

Forwards:

Name	Path
invalid	/error.jsp
valid	/home.jsp

Form Bean Name: loginForm

Form Bean Scope:

Model: Generic Action Mapping

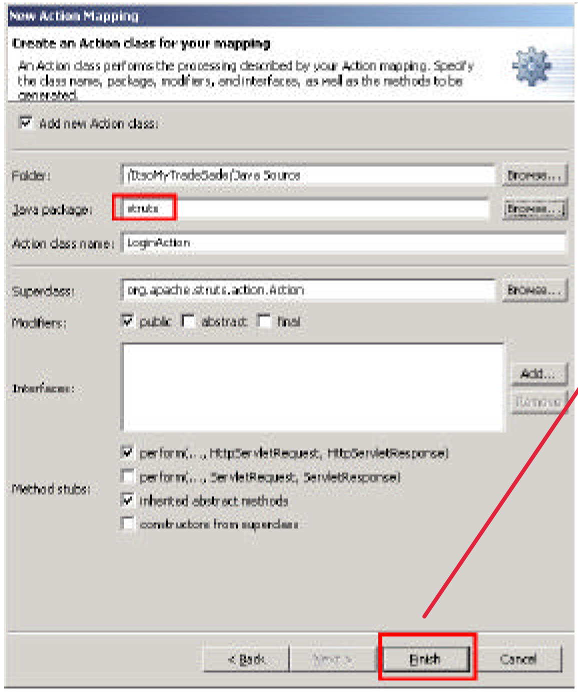
Description: Generate a new Action mapping

Files: Generic Action class

File Description: An Action class performs the processing described by your Action mapping. Specify its name, package, modifiers, interfaces, and methods to be generated.

WebSphere software

Example2 : Creating action Mapping and class...


New Action Mapping

Create an Action class for your mapping

An Action class performs the processing described by your Action mapping. Specify the class name, package, modifiers, and interfaces, as well as the methods to be generated.

[✓] Add new Action class:

Folder: /ItsoMyTradeSade/Java Source

Java package: struts

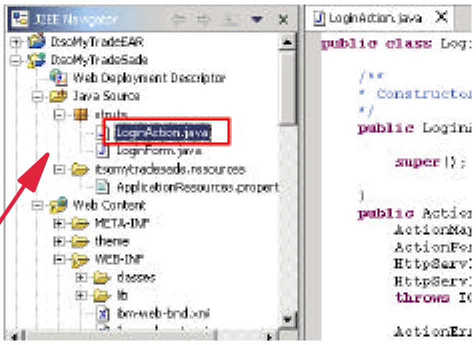
Action class name: LoginAction

Superclass: org.apache.struts.action.Action

Modifiers: [✓] public [ ] abstract [ ] final

Interfaces:

Method stubs:
[✓] perform(..., HttpServletRequest, HttpServletResponse)
[ ] perform(..., ServletRequest, ServletResponse)
[✓] inherited abstract methods
[ ] constructors from superclass


J2EE Navigator

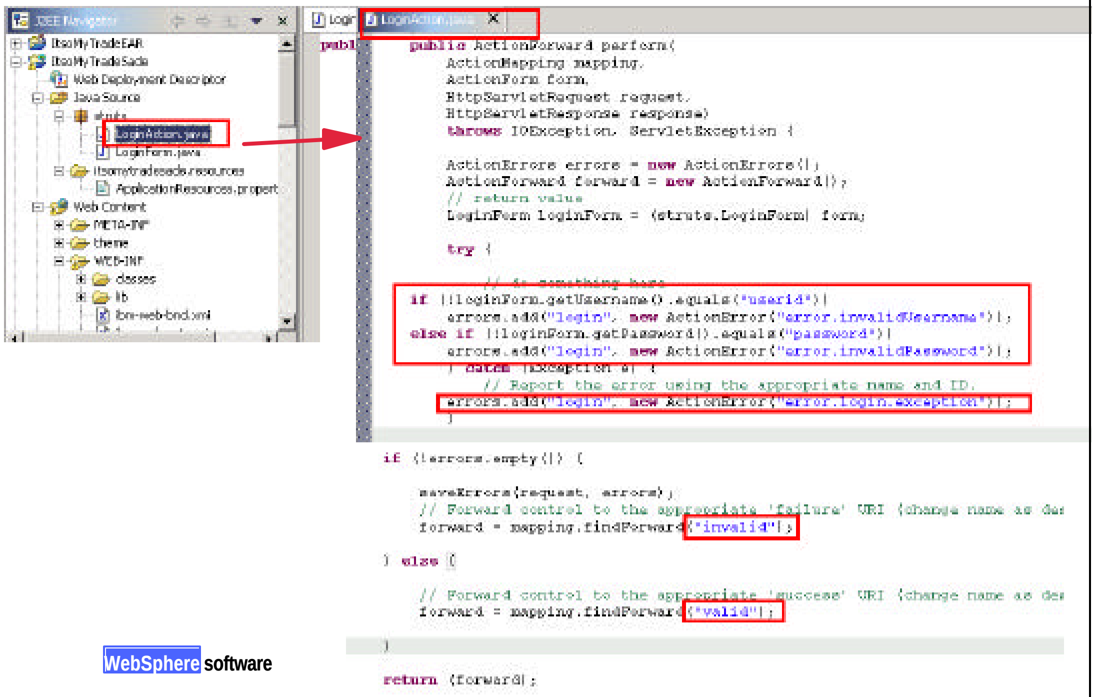
```
ItsoMyTradeEAR
ItsoMyTradeSade
  Web Deployment Descriptor
  Java Source
    struts
      LoginAction.java
      LoginForm.java
    itsomytradesade.resources
      ApplicationResources.propert
  Web Content
    META-INF
    theme
    WEB-INF
      classes
      lib
      ibm-web-bnd.xmi
```

LoginAction.java

```
public class Log
    /**
     * Constructe
     */
    public Login
        super();
    }
    public Actio
        ActionMa
        ActionFo
        HttpServ
        HttpServ
        throws IO
        ActionEr
```

IBM

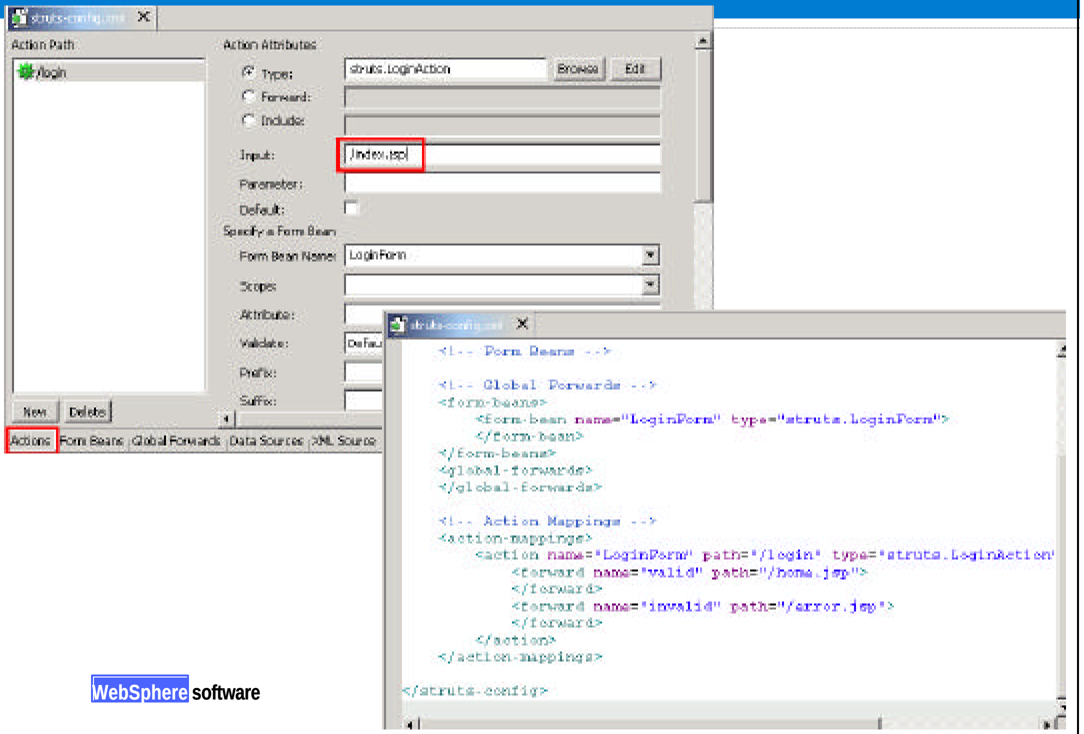
Example2 : Complete method in action class



```
public ActionForward perform(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException {

    ActionErrors errors = new ActionErrors();
    ActionForward forward = new ActionForward();
    // return value
    LoginForm loginForm = (struts.LoginForm) form;

    try {

        // do something here
if (!loginForm.getUsername().equals("userid"))
    errors.add("login", new ActionError("error.invalidUsername"));
else if (!loginForm.getPassword().equals("password"))
    errors.add("login", new ActionError("error.invalidPassword"));
    } catch (Exception e) {
        // Report the error using the appropriate name and ID.
    errors.add("login", new ActionError("error.login.exception"));
    }

 if (!errors.empty()) {

     saveErrors(request, errors);
     // Forward control to the appropriate 'failure' URI (change name as des
     forward = mapping.findForward("invalid");

 } else {

     // Forward control to the appropriate 'success' URI (change name as des
     forward = mapping.findForward("valid");

 }

 return (forward);
```

WebSphere software

Example2 : Complete the Struts configuration file



Action Path: /login

Type: struts.LoginAction

Forward:

Include:

Input: /index.jsp

Parameter:

Default:

Form Bean Name: LoginForm

Scope:

Attribute:

Validate: Defau

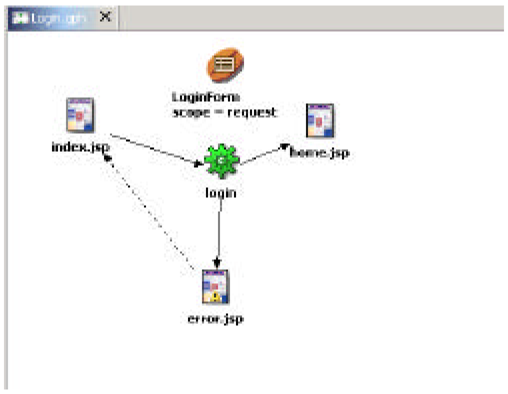
Prefix:

Suffix:

```
    <!-- Form Beans -->

    <!-- Global Forwards -->
    <form-beans>
        <form-bean name="LoginForm" type="struts.LoginForm">
        </form-bean>
    </form-beans>
    <global-forwards>
    </global-forwards>

    <!-- Action Mappings -->
    <action-mappings>
        <action name="LoginForm" path="/login" type="struts.LoginAction"
            <forward name="valid" path="/home.jsp">
            </forward>
            <forward name="invalid" path="/error.jsp">
            </forward>
        </action>
    </action-mappings>

</struts-config>
```

WebSphere software

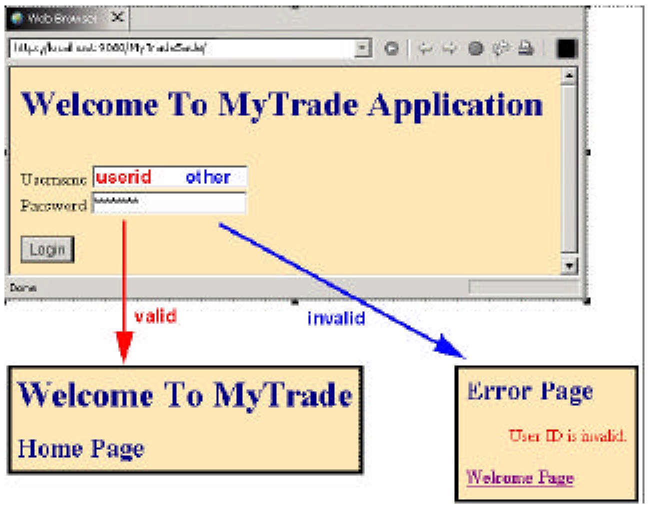
Example2 : Diagram completed



WebSphere software

IBM

Example2 : Testing



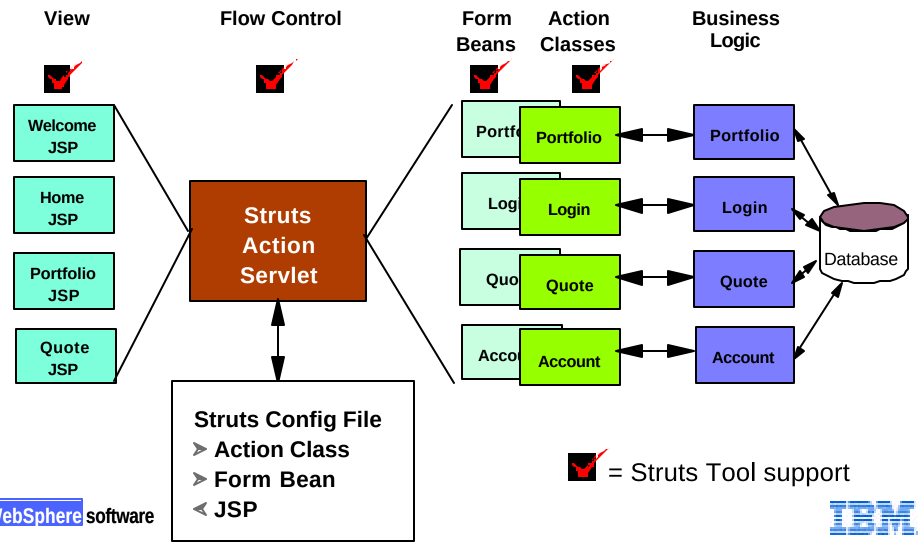
must enter userid and password

WebSphere software

IBM

Struts Sample Application

- Struts is an open source implementation of MVC2
- Struts Tools enable faster development with less errors



✓ = Struts Tool support

WebSphere software

IBM

VG Web Transaction today

VG Web Transactions (simplified)

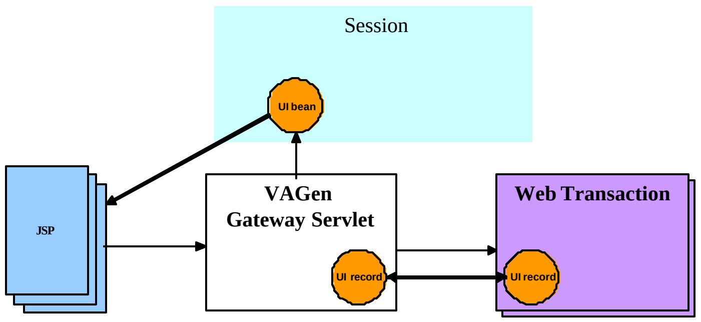
The User enters data on a JSP form and presses "Submit" button

The VAGen Gateway Servlet places relevant data in an input UI record and passes it to the appropriate Web transaction for processing

The Web transaction performs its "action" and returns an output UI record which includes "where to go next"

The VAGen Gateway Servlet takes the UI record, places its data in a UI bean and calls the appropriate JSP

The JSP uses the data stored in the UI bean to populate it's own fields and the process repeats



WebSphere software

IBM

Struts Action Servlet Overview

Struts Action Servlet Overview (simplified)

The User enters data on a JSP form and presses "Submit" button

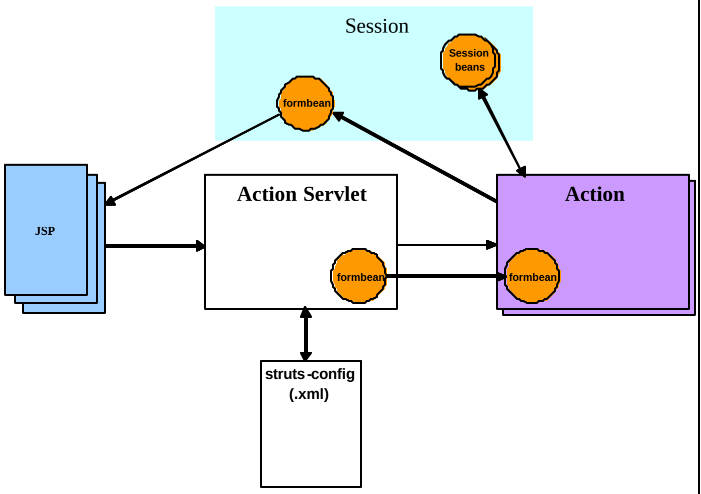
The Action Servlet checks the struts-config.xml file to determine which action should be invoked

It then takes the JSP's form data, places it in a formbean and invokes the appropriate action, passing the form bean

The Action takes the data from the formbean and performs its 'action' (updating other beans as needed)

The Action may place a formbean in the session, and sends a "forward" to the Action Servlet which invokes the next appropriate JSP (or Action, as needed)

The JSP uses the data stored in the formbean to populate it's own fields and the process repeats

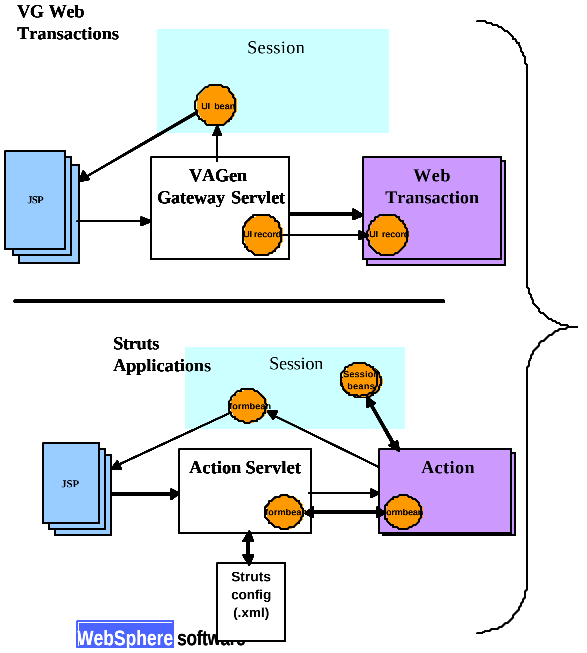


WebSphere software

IBM

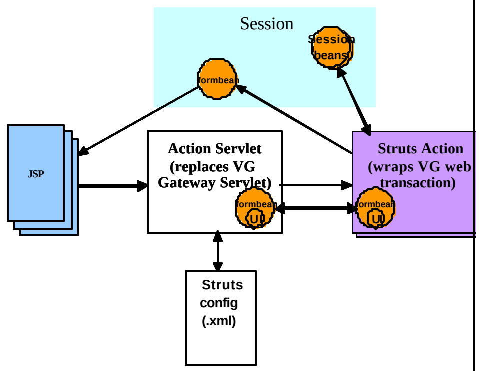
Struts and VG Web Transactions

Struts and VG Web Transactions (quick comparison)



Struts Applications and Web Transactions are actually very similar:

JSPs for Views
Beans hold data for JSP usage
Specialized Servlet Controller
Struts Actions fairly analogous to Web Transactions


Session
Session beans
formbean
JSP
Action Servlet (replaces VG Gateway Servlet)
Struts Action (wraps VG web transaction)
formbean
UI
formbean
UI
Struts config (.xml)

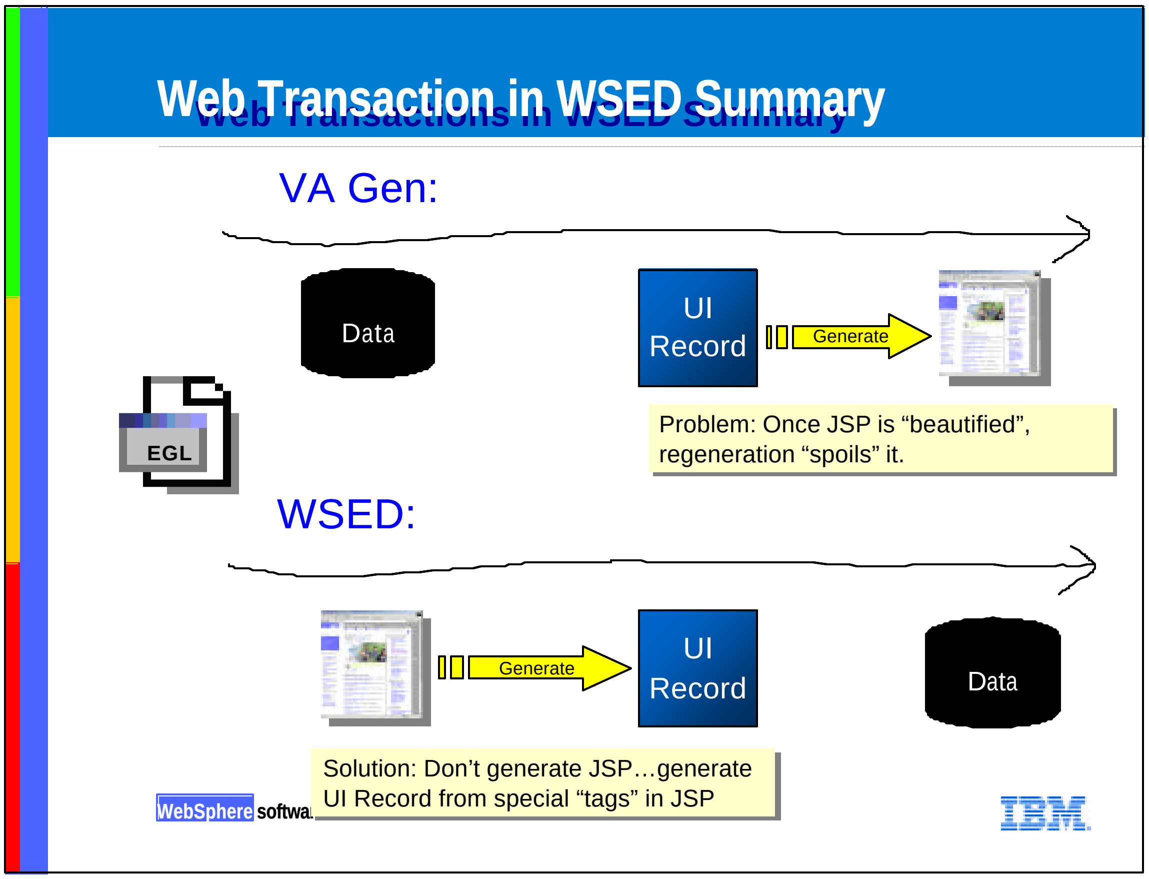
WSED

Replace VAGen Gateway Servlet with Struts Action Servlet

Make UI record a subclass of formbean

Wrap Web Transactions within Actions

WebSphere software



Web Transaction in WSED Summary

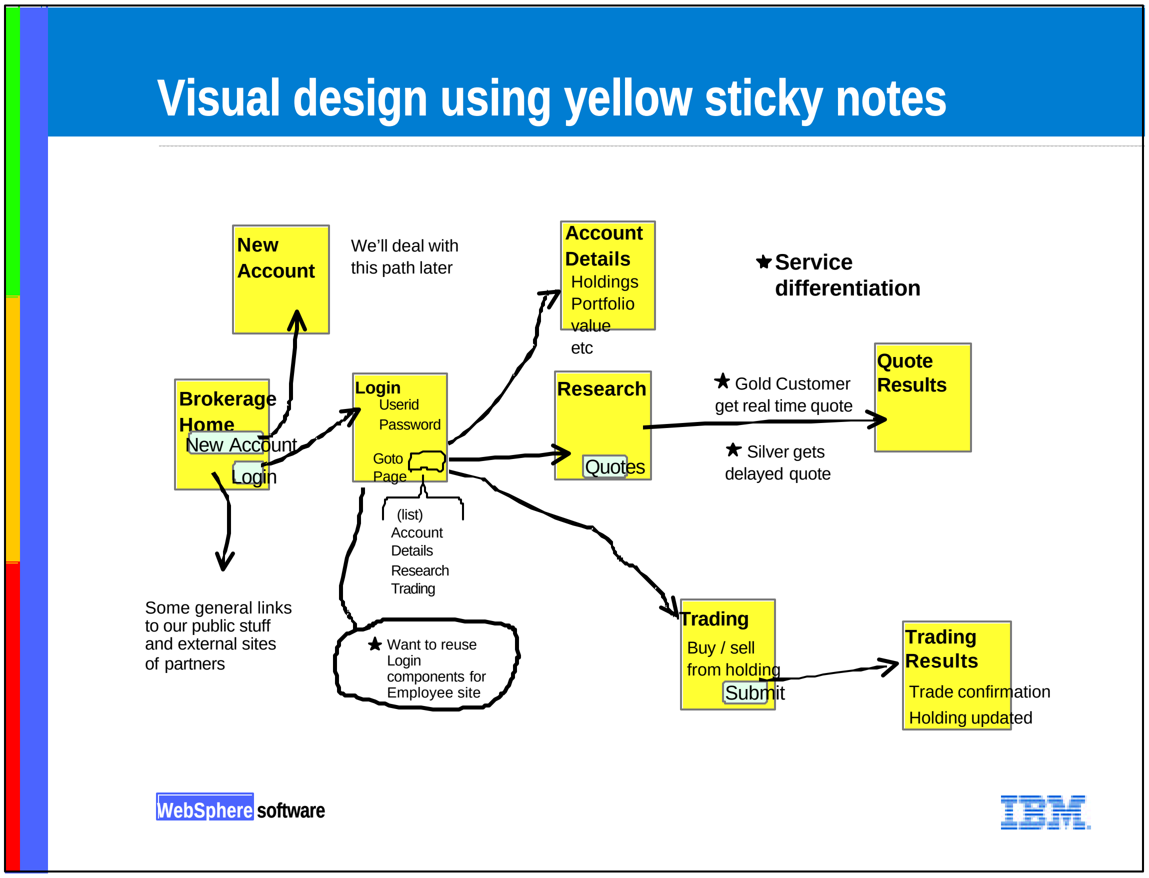
Web Transactions in WSED Summary

VA Gen:

Problem: Once JSP is “beautified”, regeneration “spoils” it.

WSED:

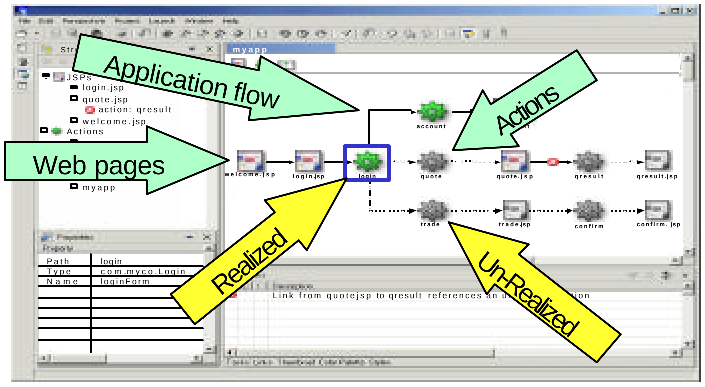
Solution: Don’t generate JSP…generate UI Record from special “tags” in JSP



Visual design using yellow sticky notes

Web Diagram Editor

- Design
  - ►Add web pages, actions
  - ►Define application flow
- Create
  - ►Web pages: JSP, HTML
  - ►Actions: Java, COBOL, EGL



Path	login
Type	com.myco.Login
Name	loginForm

Link from quotejsp to qresult references an u... ...ion

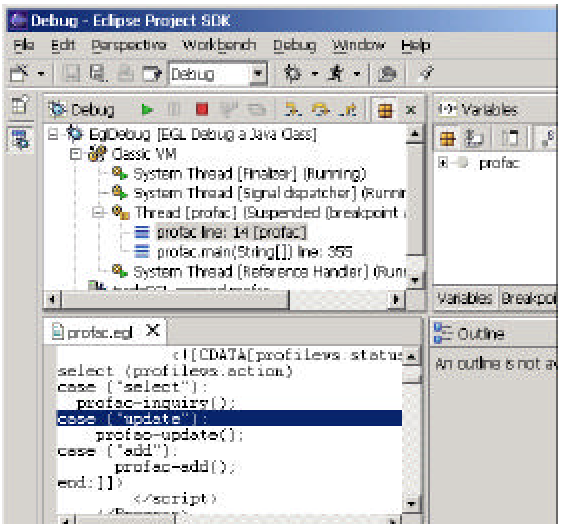
Benefit: Faster construction of web applications

WebSphere software

IBM

Testing and Debugging

- End-to-end Debugging
  - ►Java and JSP debugger
  - ►COBOL, PL/I debugger
  - ►EGL debugger
- Verifying application Flow
  - ►breakpoints
  - ►changing variable values
- WebSphere Test Environment
  - ►integrated in Workbench
  - ►choice of versions



```
EglDebug [EGL Debug a Java Class]
  Classic VM
    System Thread [Finalizer] (Running)
    System Thread [Signal dispatcher] (Runni
    Thread [profac] (Suspended (breakpoint
      profac line: 14 [profac]
      profac.main(String[]) line: 355
    System Thread [Reference Handler] (Runi
```

```
              <![CDATA[profilews.status
select (profilews.action)
case ("select"):
  profac-inquiry();
case ("update"):
    profac-update();
case ("add"):
      profac-add();
end;]]>
          </script>
```

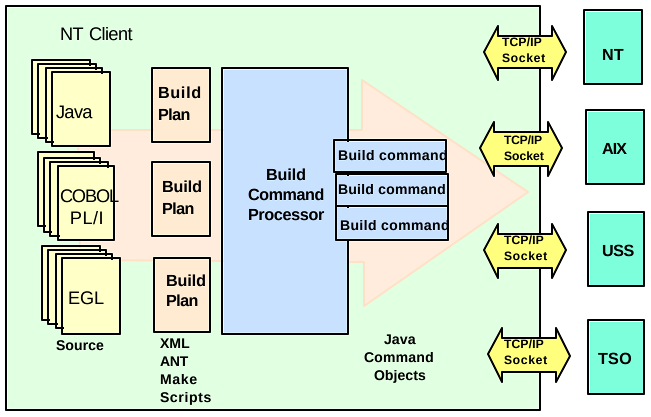
Benefit: End-to-end test and debug from the Workbench

WebSphere software

IBM

Building and Deploying

- Automated Build based on Build Plans (XML)
- Automatic transfer to target machine
- Run build commands on target machine



Benefit: Developer spends less time in the build process

WebSphere software

IBM

Functions planned* for GA release

- Enhancements - usability, function, ...
  - Struts
    - Simplified definition for the non-Java programmer
  - EGL
    - Web Transaction support
    - generation of Web Services interface
  - z/OS Application Development
    - Code Assist for COBOL, PL/I
- WSAD-IE dependencies
  - new Web Service for XML enabled z/OS applications
    - for XML pass through support
  - JCA connectors
    - for Struts to communicate to z/OS programs (EGL, COBOL)
    - for Web Service to connect to z/OS Driver Transaction

* Not all planned functions may make GA release

WebSphere software

IBM

Functions planned* for GA release

- Enhancements - usability, function, ...
  - ►Struts
    - Simplified definition for the non-Java programmer
  - ►EGL
    - Web Transaction support
    - generation of Web Services interface
  - ►z/OS Application Development
    - Code Assist for COBOL, PL/I
- WSAD-IE dependencies
  - ►new Web Service for XML enabled z/OS applications
    - for XML pass through support
  - ►JCA connectors
    - for Struts to communicate to z/OS programs (EGL, COBOL)
    - for Web Service to connect to z/OS Driver Transaction

* Not all planned functions may make GA release

Differences from VisualAge products

- VisualAge products bundled with WSED shipment
- VA COBOL
  - ►Some of VACOBOL function installed with WSED
  - ►Must install VACOBOL product for:
    - Data Assistant
    - BMS Map Editor
    - Performance Analyzer
- VisualAge Generator
  - ►No migration support for 4GL to EGL until 2H/2003
  - ►Must install VAGenerator for:
    - IMS DB/DC, 3270/5250 Text, Web Transactions
    - Templates

WebSphere software

IBM

e business software

WebSphere Studio Enterprise Developer V5.0

z/OS Application Development

Jan 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

WebSphere Studio Enterprise Developer V5.0

- Struts Tools
  - ►Set of Wizards, editors, and validation support
  - ►for the design and construction of Struts-based J2EE web applications
- Enterprise Generation Language (EGL)
  - ►Simple, high level programming specifications
  - ►for creating full-function COBOL and Java applications
- z/OS Application Development Tools
  - ►Interactive, workstation-based development
  - ►for mainframe COBOL, PL/I, ASM applications
- XML Enablement Enhancements for z/OS applications
  - ►Set of wizards to create XML transformation code
  - ►and web services for XML-enabled z/OS applications

WebSphere software

IBM

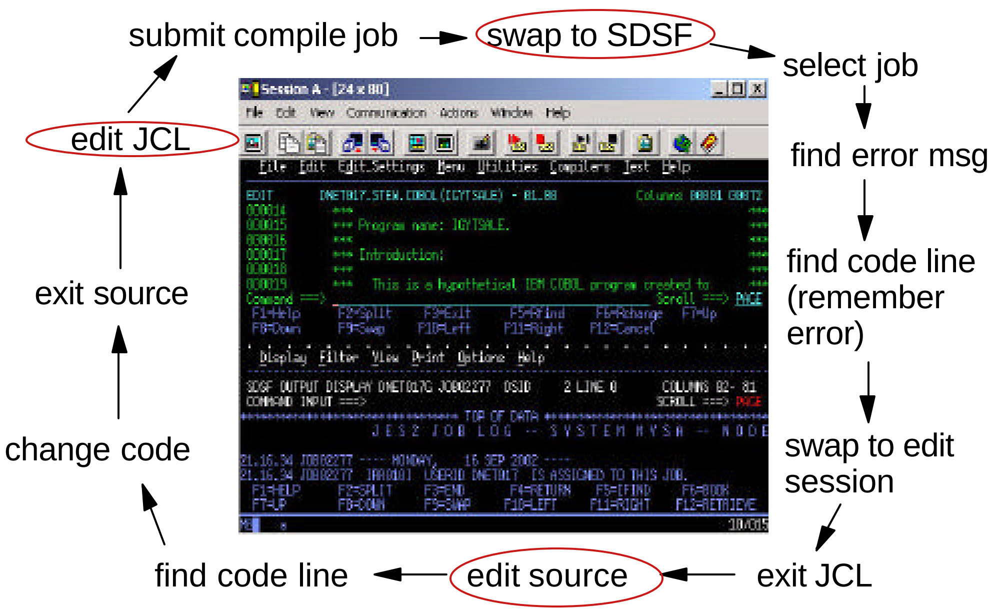
Key Benefits of Enterprise Developer

- Struts Tools
  - Rapid design and quicker understanding of complex web applications
  - Faster development with less errors of well-structured web applications
- Enterprise Generation Language
  - Rapid development
  - Cross platform applications (CICS, WebSphere Application Server)
  - Using existing programmers with traditional business skills
- z/OS Application Development
  - Extends benefits of WebSphere Studio Workbench features/tools to COBOL, PL/I, ASM programmers
  - and the z/OS applications they create and maintain
- XML Enablement Enhancements for z/OS applications
  - Faster development of XML-enabled z/OS applications and of supporting Web Services

WebSphere software

IBM

ISPF based development - Old

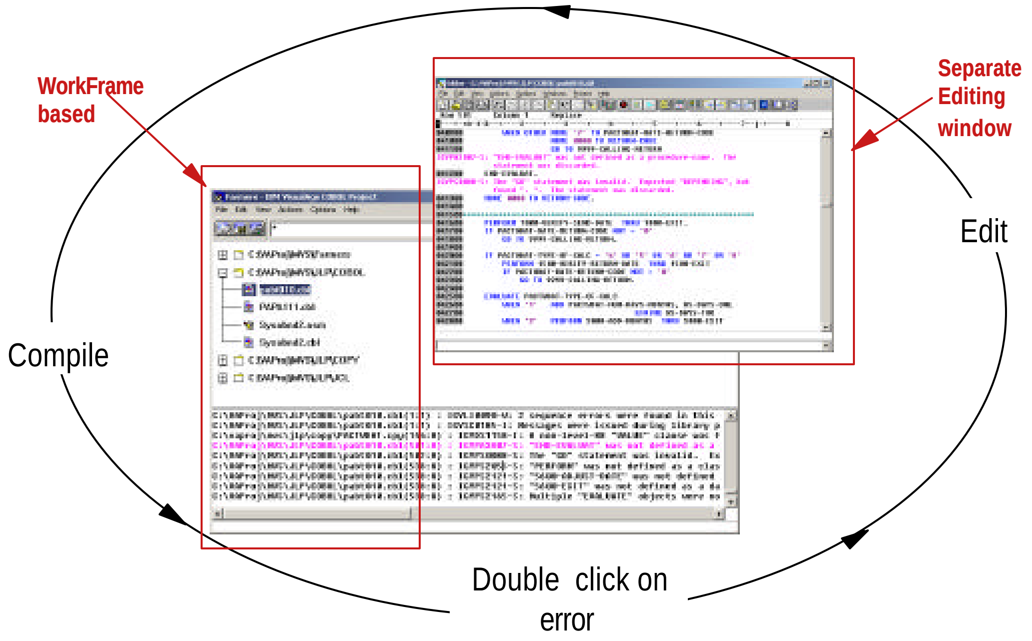


```
  File  Edit  Edit_Settings  Menu  Utilities  Compilers  Test  Help
EDIT       DNET017.STEM.COBOL(IGYTSALE) - 01.08            Columns 00001 00072
000014        ***                                                          ***
000015        *** Program name: IGYTSALE.                                  ***
000016        ***                                                          ***
000017        *** Introduction:                                            ***
000018        ***                                                          ***
000019        ***   This is a hypothetical IBM COBOL program created to    ***
Command ===>                                                  Scroll ===> PAGE
  F1=Help      F2=Split     F3=Exit      F5=Rfind     F6=Rchange   F7=Up
  F8=Down      F9=Swap     F10=Left     F11=Right    F12=Cancel

  Display  Filter  View  Print  Options  Help

SDSF OUTPUT DISPLAY DNET017G JOB02277  DSID     2 LINE 0       COLUMNS 02- 81
COMMAND INPUT ===>                                            SCROLL ===> PAGE
********************************* TOP OF DATA ********************************
                         J E S 2  J O B  L O G  --  S Y S T E M  M V S A  --  N O D E

21.16.34 JOB02277 ---- MONDAY,    16 SEP 2002 ----
21.16.34 JOB02277  IRR010I  USERID DNET017  IS ASSIGNED TO THIS JOB.
  F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
  F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RETRIEVE
                                                                        10/015
```

WebSphere software

IBM

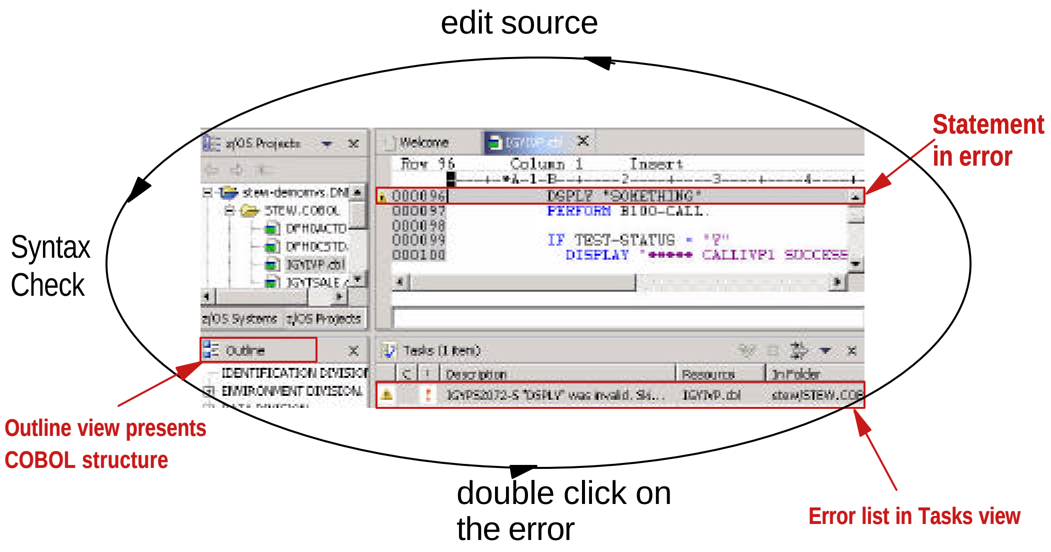
VisualAge COBOL - Current



WebSphere software

IBM

WebSphere Studio Workbench - New



```
Row 96        Column 1       Insert
000096              DSPLY "SOMETHING"
000097              PERFORM B100-CALL.
000098
000099              IF TEST-STATUS = "Y"
000100                DISPLAY "***** CALLIVP1 SUCCESS
```

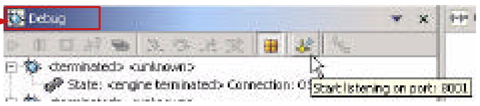
IDENTIFICATION DIVISION
ENVIRONMENT DIVISION.

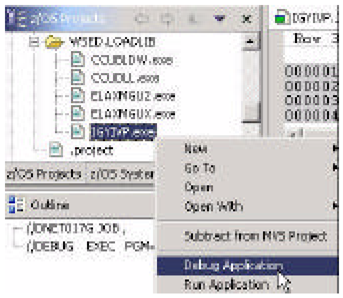
C	!	Description	Resource	In Folder
	!	IGYPS2072-S "DSPLY" was invalid. Sk...	IGYIVP.cbl	stew/STEW.COB

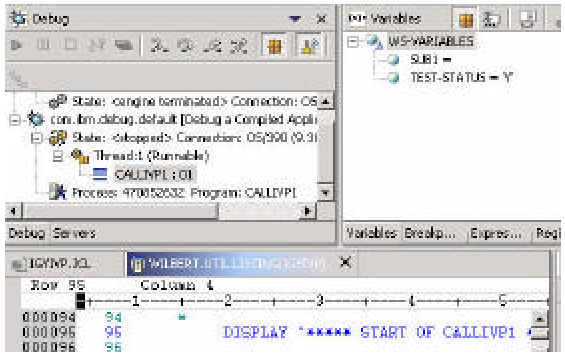
Benefit: Simplified development for COBOL and PL/I on a common development environment

WebSphere Studio Workbench - New ...

Same Debug Perspective


Debug
<terminated> <unknown>
State: <engine terminated> Connection: O
Start listening on port: 8001



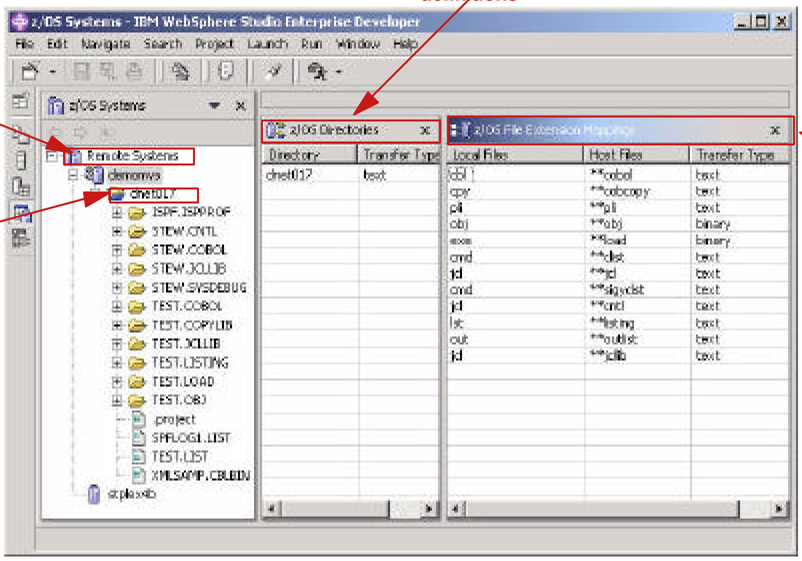

Debug
Variables
WS-VARIABLES
SUB1 =
TEST-STATUS = 'Y'
State: <engine terminated> Connection: OS
com.ibm.debug.default [Debug a Compiled Appli
State: <stopped> Connection: OS/390
Thread:1 (Runnable)
CALLIVP1 : 01
Process: 470852632 Program: CALLIVP1
Debug
Servers
Variables
Breakp...
Expres...
IGYIVP.JCL
Row 95
Column 4
DISPLAY "***** START OF CALLIVP1

Benefit: Consistent debugging environment for COBOL, PL/I, Java

Benefits of z/OS Application Development

- Utilizes Workbench features/tools to support COBOL, PL/I, Assembler development for the z/OS platform
  - ►Simplifies development process
  - ►Provides consistent development environment
- Provides development support for traditional runtimes
  - ►CICS, IMS, DB2, batch

z/OS Systems Perspective

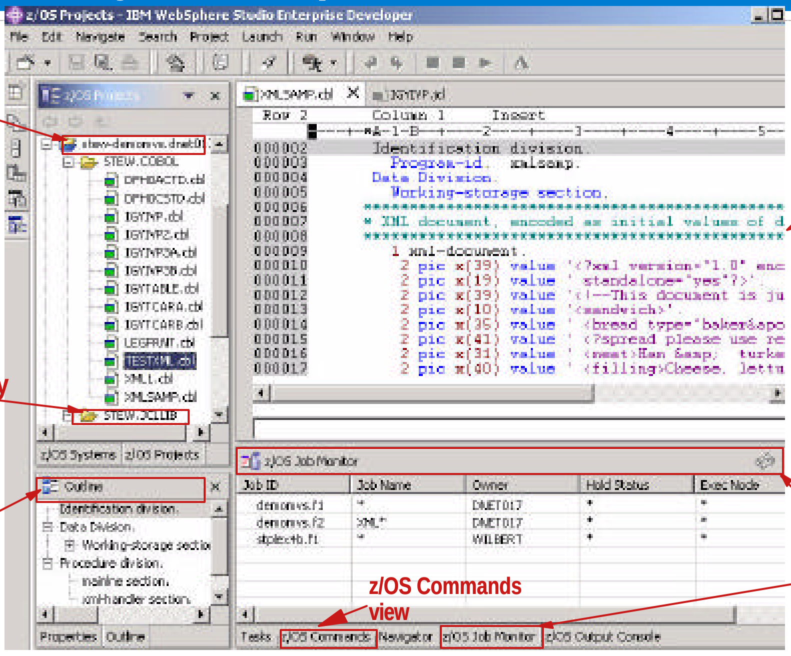


Directory	Transfer Type
dnet017	text

Local Files	Host Files	Transfer Type
cbl	**cobol	text
cpy	**cobcopy	text
pli	**pli	text
obj	**obj	binary
exe	**load	binary
cmd	**clist	text
jcl	**jcl	text
cmd	**sigyclst	text
jcl	**cntl	text
lst	**listing	text
out	**outlist	text
jcl	**jcllib	text

WebSphere software

z/OS Projects Perspective

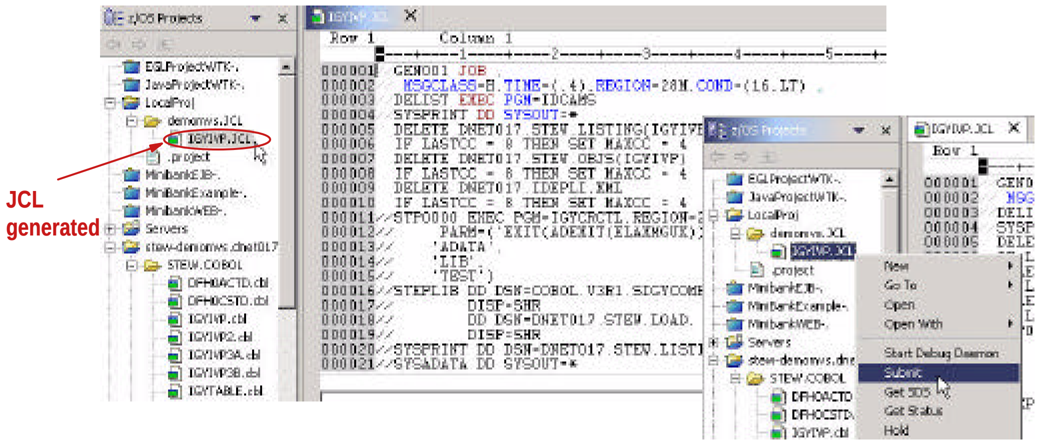


```
 Row 2         Column 1      Insert
     ----+-*A-1-B--+----2----+----3----+----4----+----5--
       Identification division.
         Program-id.  xmlsamp.
       Data Division.
         Working-storage section.
      ****************************************************
      * XML document, encoded as initial values of d
      ****************************************************
         1 xml-document.
          2 pic x(39) value '<?xml version="1.0" enc
          2 pic x(19) value ' standalone="yes"?>'.
          2 pic x(39) value '<!--This document is ju
          2 pic x(10) value '<sandwich>'.
          2 pic x(35) value '  <bread type="baker&apo
          2 pic x(41) value '  <?spread please use re
          2 pic x(31) value '  <meat>Ham &amp;  turke
          2 pic x(40) value '  <filling>Cheese, lettu
```

Job ID	Job Name	Owner	Hold Status	Exec Node
demomvs.f1	*	DNET017	*	*
demomvs.f2	XML*	DNET017	*	*
stplex4b.f1	*	WILBERT	*	*

WebSphere software

JCL Generation and Submission

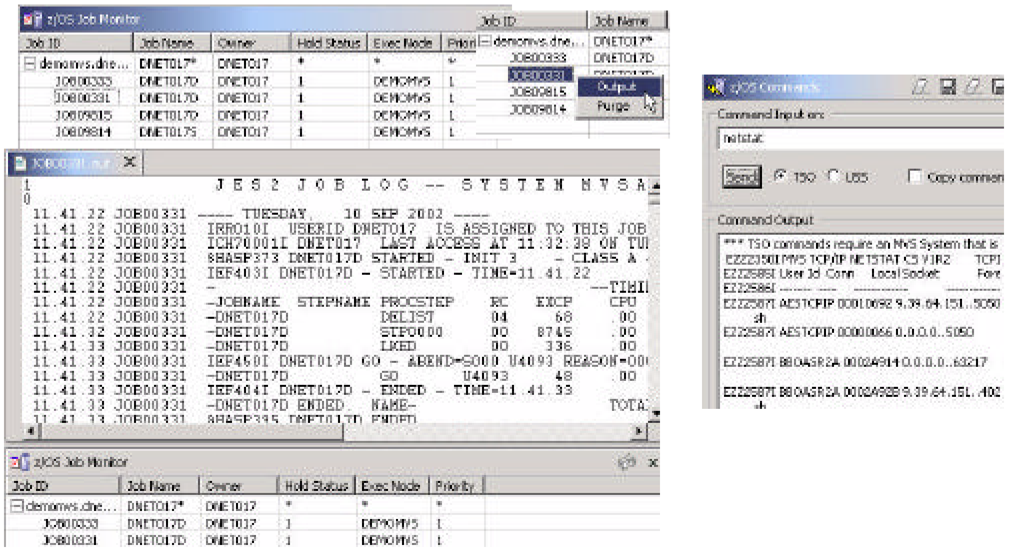


```
Row 1        Column 1
 ----+----1----+----2----+----3----+----4----+----5----+-
GEN001 JOB ,
 MSGCLASS=H,TIME=(,4),REGION=28M,COND=(16,LT)
DELIST EXEC PGM=IDCAMS
SYSPRINT DD SYSOUT=*
DELETE DNET017.STEW.LISTING(IGYIVP
IF LASTCC = 8 THEN SET MAXCC = 4
DELETE DNET017.STEW.OBJS(IGYIVP)
IF LASTCC = 8 THEN SET MAXCC = 4
DELETE DNET017.IDEPLI.XMI
IF LASTCC = 8 THEN SET MAXCC = 4
//STP0000 EXEC PGM=IGYCRCTL,REGION=2
//      PARM=('EXIT(ADEXIT(ELAXMGUK)
//     'ADATA',
//     'LIB',
//     'TEST')
//STEPLIB DD DSN=COBOL.V3R1.SIGYCOMP
//         DISP=SHR
//         DD DSN=DNET017.STEW.LOAD,
//         DISP=SHR
//SYSPRINT DD DSN=DNET017.STEW.LISTI
//SYSADATA DD SYSOUT=*
```

Benefit: Developers focused on business logic and not on writing JCL

WebSphere software IBM

Monitoring Job Output / Issuing Commands



Job ID	Job Name	Owner	Hold Status	Exec Node	Priority
demomvs.dne...	DNET017*	DNET017	*	*	*
JOB00333	DNET017D	DNET017	1	DEMOMVS	1
JOB00331	DNET017D	DNET017	1	DEMOMVS	1
JOB09815	DNET017D	DNET017	1	DEMOMVS	1
JOB09814	DNET017S	DNET017	1	DEMOMVS	1

```
1                    J E S 2  J O B  L O G  --  S Y S T E M  M V S A
0
 11.41.22 JOB00331 ---- TUESDAY,   10 SEP 2002 ----
 11.41.22 JOB00331  IRR010I  USERID DNET017  IS ASSIGNED TO THIS JOB
 11.41.22 JOB00331  ICH70001I DNET017  LAST ACCESS AT 11:32:38 ON TU
 11.41.22 JOB00331  $HASP373 DNET017D STARTED - INIT 3    - CLASS A
 11.41.22 JOB00331  IEF403I DNET017D - STARTED - TIME=11.41.22
 11.41.22 JOB00331  -                                         --TIMI
 11.41.22 JOB00331  -JOBNAME  STEPNAME PROCSTEP    RC   EXCP    CPU
 11.41.22 JOB00331  -DNET017D          DELIST      04     68    .00
 11.41.32 JOB00331  -DNET017D          STP0000     00   8745    .00
 11.41.33 JOB00331  -DNET017D          LKED        00    336    .00
 11.41.33 JOB00331  IEF450I DNET017D GO - ABEND=S000 U4093 REASON=00
 11.41.33 JOB00331  -DNET017D          GO       U4093     48    .00
 11.41.33 JOB00331  IEF404I DNET017D - ENDED - TIME=11.41.33
 11.41.33 JOB00331  -DNET017D ENDED.  NAME-                     TOTA
 11.41.33 JOB00331  $HASP395 DNET017D ENDED
```

Job ID	Job Name	Owner	Hold Status	Exec Node	Priority
demomvs.dne...	DNET017*	DNET017	*	*	*
JOB00333	DNET017D	DNET017	1	DEMOMVS	1
JOB00331	DNET017D	DNET017	1	DEMOMVS	1

Command Input area: netstat

[✓ TSO] USS

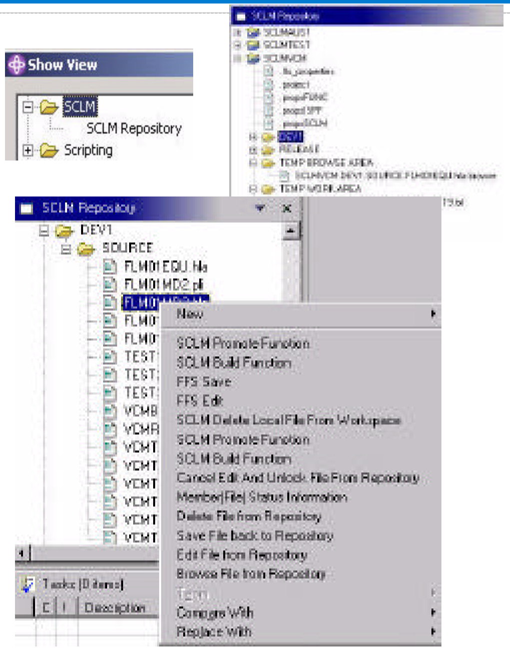
Command Output

```
*** TSO commands require an MVS System that is
EZZ2350I MVS TCP/IP NETSTAT CS V1R2       TCPI
EZZ2585I User Id  Conn     Local Socket        Fore
EZZ2586I -------- ----     ------------        ----------
EZZ2587I AESTCPIP 00010092 9.39.64.151..5050
     sh
EZZ2587I AESTCPIP 00000066 0.0.0.0..5050

EZZ2587I BBOASR2A 00024914 0.0.0.0..63217

EZZ2587I BBOASR2A 0002492B 9.39.64.151..402
     sh
```

Benefit: Developers do not have to continually switch between systems

SCLM Support

- Uses VCM adapter
- Access to SCLM services on z/OS
  - Connect to SCLM repository
  - View a list of projects
  - List project members
  - Execute SCLM actions
- Check-in/check-out support
  - TEMP WORK AREA
- No SCLM administrative functions
  - Create SCLM project
  - Delete SCLM project
- No automatic synchronization
  - Manual refresh

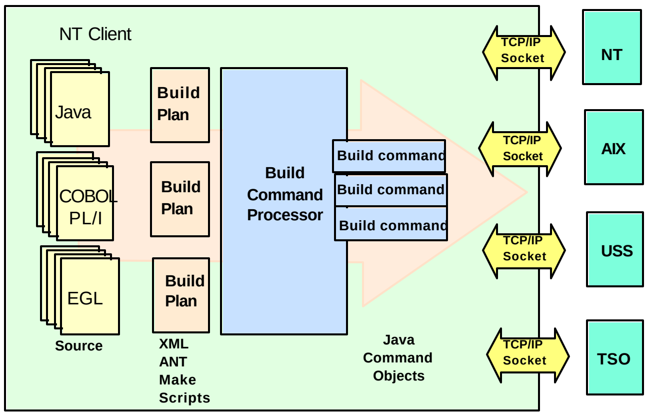


WebSphere software

IBM

Building and Deploying

- Automated Build based on Build Plans (XML)
- Automatic transfer to target machine
- Run build commands on target machine



Benefit: Developer spends less time in the build process

WebSphere software

IBM

Summary

- Increases programmer productivity
  - ►Simplified development process
  - ►Unified development environment
- Reduces Total Cost of Ownership (TCO) by adopting a consistent development environment for the enterprise
  - ►Single development environment to manage and deploy vs. multiple
  - ►Simplified training requirements
- Facilitates the building and testing of z/OS applications by providing development support for traditional runtimes like CICS, IMS, DB2, and batch

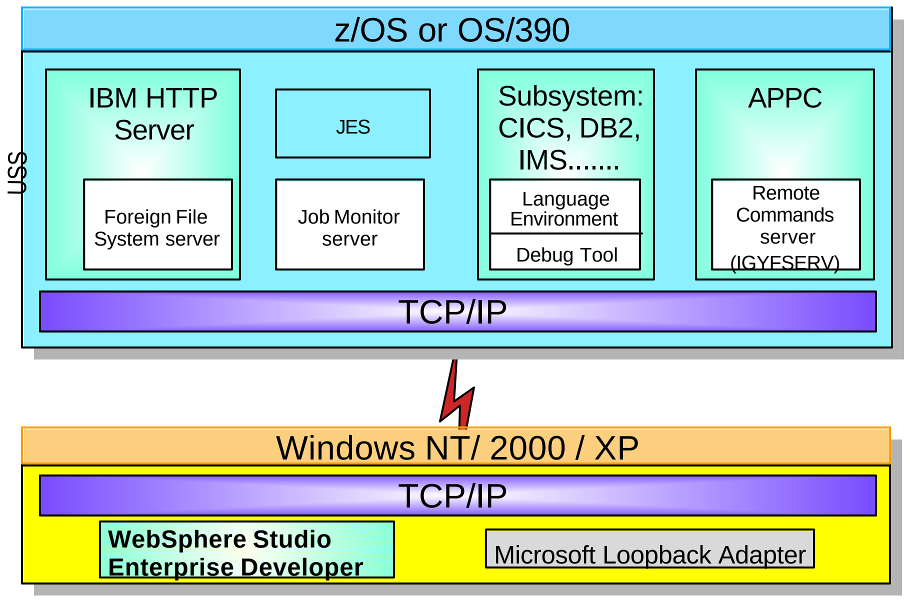
z/OS Application Development

Software prerequisites

Workstation Prerequisites

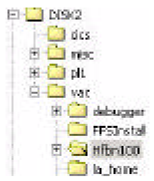
- Install and configure Microsoft Loopback Adapter
- Install the z/OS Application Development Tools on disk 2 of WebSphere Studio Enterprise Developer
- Modify HOSTS file (Windows 2000 only)

Host Software Requirements


z/OS or OS/390
USS
IBM HTTP Server
Foreign File System server
JES
Job Monitor server
Subsystem: CICS, DB2, IMS.......
Language Environment
Debug Tool
APPC
Remote Commands server (IGYFSERV)
TCP/IP
Windows NT/ 2000 / XP
TCP/IP
WebSphere Studio Enterprise Developer
Microsoft Loopback Adapter

z/OS Prereq. Software List

- IBM HTTP Server
- JES PTF to enable job monitor support
- Language Environment PTFs to enable z/OS IDE support
- IBM Enterprise COBOL for z/OS and OS/390
- IBM Enterprise PL/I for z/OS and OS/390
- IBM Debug Tool for z/OS and OS/390
- IBM Foreign File System Server
- IBM Job Monitor Server



WebSphere software

Host Software Install Steps

- Make sure required software and service updates installed
  - TCP/IP
  - Language Environment
  - IBM HTTP Server
  - RACF or equivalent
  - IBM Enterprise COBOL
  - IBM Debug Tool
- Install foreign file system server and job monitor server
- Configure the IBM HTTP Server
- Configure the software that comes with WebSphere Studio Enterprise Developer for the host to support remote edit-compile-debug
  - Foreign file system server
  - Job monitor server

Host Software Install Steps ...

- Install and configure the TSO command server to support issuing TSO commands from the workstation
- Configure Debug Tool for remote debugging under CICS
- Test the connections

Troubleshooting

- Ensure connectivity to host systems
  - Can you ping the host?
  - Can you access the web server?
    - http://hostsys:port/
  - Can you open the web page for the host FFS system?
    - http://hostsys:port/FFDS
  - Are you using the right ports for the web and job monitor?
    - default is 80 and 6715 respectively

WebSphere software

IBM

e-business software

WebSphere Studio Enterprise Developer V5.0

XML Enablement for z/OS

Jan, 2003

Reginaldo Barosa

Certified IT Specialist
IBM Boston
rbarosa@us.ibm.com

IBM Software Group

WebSphere Studio Enterprise Developer V5.0

- Struts Tools
  - ►Set of Wizards, editors, and validation support
  - ►for the design and construction of Struts-based J2EE web applications
- Enterprise Generation Language (EGL)
  - ►Simple, high level programming specifications
  - ►for creating full-function COBOL and Java applications
- z/OS Application Development Tools
  - ►Interactive, workstation-based development
  - ►for mainframe COBOL, PL/I, ASM applications
- XML Enablement Enhancements for z/OS applications
  - ►Set of wizards to create XML transformation code
  - ►and web services for XML-enabled z/OS applications

WebSphere software

IBM

Key Benefits of Enterprise Developer

- Struts Tools
  - Rapid design and quicker understanding of complex web applications
  - Faster development with less errors of well-structured web applications
- Enterprise Generation Language
  - Rapid development
  - Cross platform applications (CICS, WebSphere Application Server)
  - Using existing programmers with traditional business skills
- z/OS Application Development
  - Extends benefits of WebSphere Studio Workbench features/tools to COBOL, PL/I, ASM programmers
  - and the z/OS applications they create and maintain
- XML Enablement Enhancements for z/OS applications
  - Faster development of XML-enabled z/OS applications and of supporting Web Services

WebSphere software

IBM

What is XML Enablement?

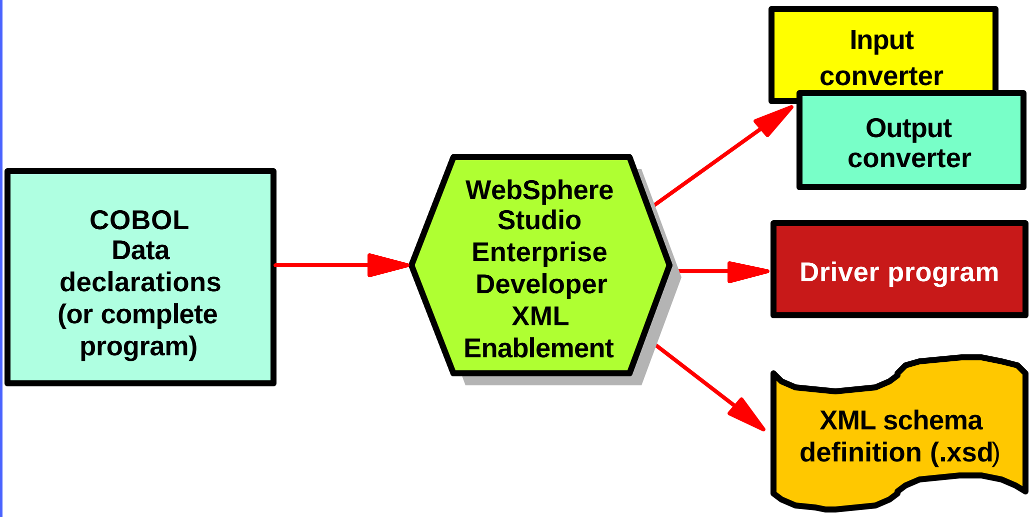
Enables COBOL-based applications to consume and produce XML messages

- Leverages XML parsing capabilities of IBM Enterprise COBOL V3.1
- Creates COBOL converter programs
  - Inbound to convert XML messages into native COBOL data
  - Outbound to convert native COBOL data into XML messages
- Creates template COBOL driver program
  - Illustrate the invocation of converters
  - Illustrate the interaction with existing application
  - Needs to be updated before run
- Enables communication with XML based systems

WebSphere software

IBM

XML Enablement

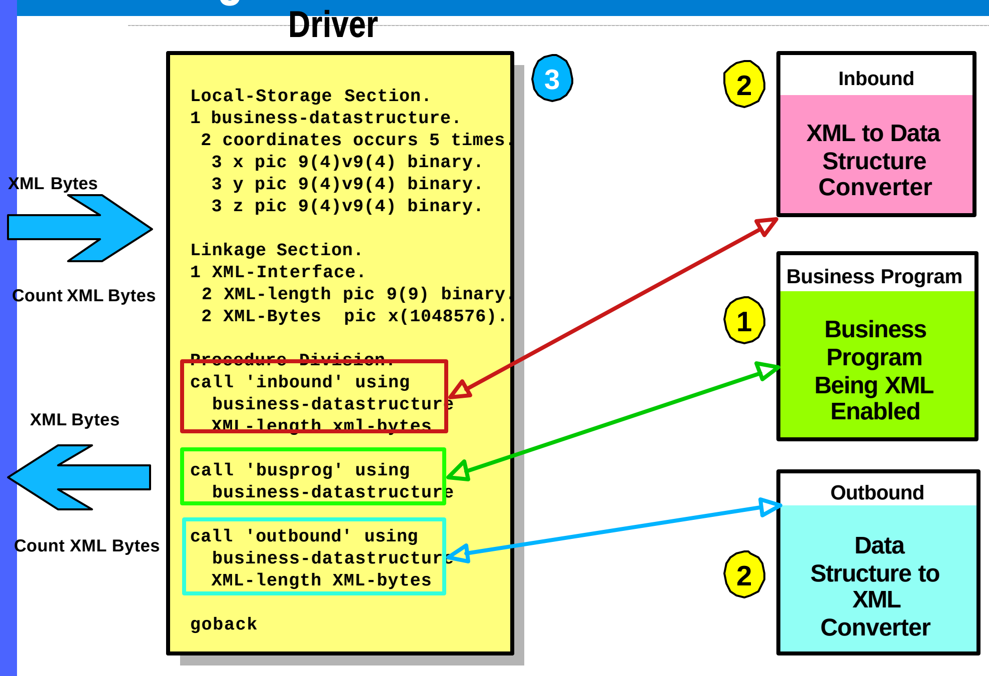

COBOL Data declarations (or complete program)
WebSphere Studio Enterprise Developer XML Enablement
Input converter
Output converter
Driver program
XML schema definition (.xsd)

- Enables COBOL-based applications to consume and produce XML messages
  - ►Original COBOL program unchanged

Benefits of XML Enablement

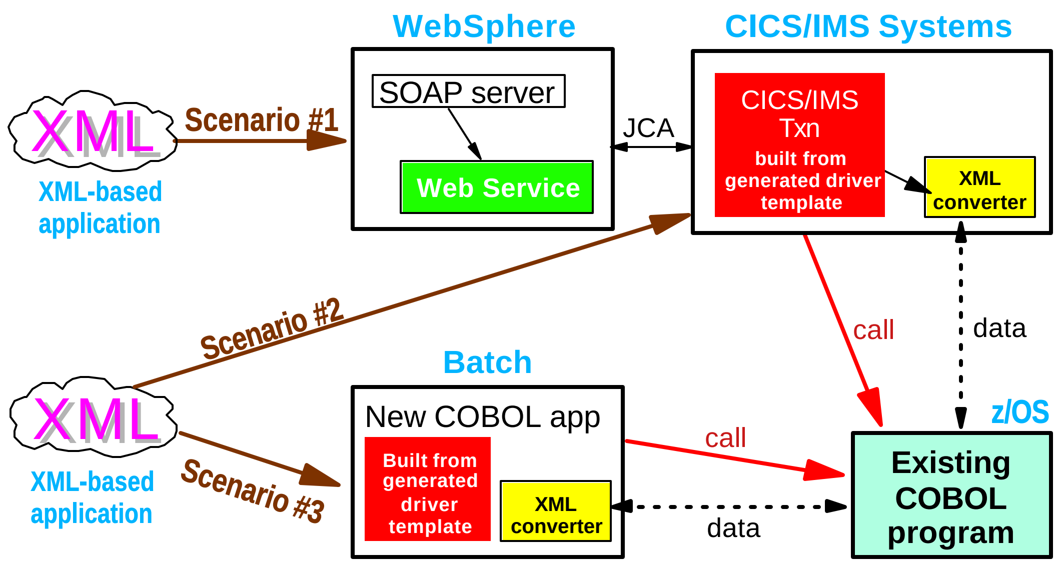
- Enterprise Modernization:
  - ►Easy to "reface" existing COBOL applications to support XML messages
- Programmer Productivity:
  - ►Converter programs are generated to easily convert between XML and COBOL datatypes
  - ►Template program generated which illustrates how converter programs are used with existing COBOL
  - ►Exploits customers' existing assets/skills/literacy
- Performance
  - ►XML parsing/conversions run on z/OS

Using the XML Converters


Driver
3
XML Bytes
Count XML Bytes
XML Bytes
Count XML Bytes
2
Inbound
XML to Data Structure Converter
Business Program
1
Business Program Being XML Enabled
Outbound
2
Data Structure to XML Converter

```
Local-Storage Section.
1 business-datastructure.
 2 coordinates occurs 5 times.
  3 x pic 9(4)v9(4) binary.
  3 y pic 9(4)v9(4) binary.
  3 z pic 9(4)v9(4) binary.

Linkage Section.
1 XML-Interface.
 2 XML-length pic 9(9) binary.
 2 XML-Bytes  pic x(1048576).

Procedure Division.
call 'inbound' using
  business-datastructure
  XML-length xml-bytes
call 'busprog' using
  business-datastructure
call 'outbound' using
  business-datastructure
  XML-length XML-bytes

goback
```

XML Enablement - Runtime Scenarios



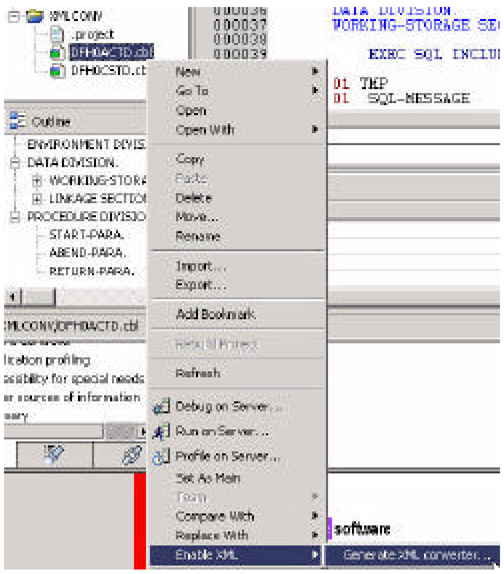
General Limitations

- Workbench
  - ►MVS Project cannot be source and target (must use local project)
  - ►Copy books must be fully expanded
- z/OS Runtime
  - ►Usage "COMP-X" not supported
  - ►Error handling via Language Environment exceptions
  - ►Attributes for inbound and outbound messages not supported
  - ►REDEFINING items are ignored
- Inbound message processing
  - ►Occurs-Depending-On (ODO) is supported
    - – No validation that group repetitions don't exceed **depending on** variable
  - ►Entire XML message must be scanned
- Outbound message generation
  - ►Complex Occurs-Depending-On (ODO) not supported

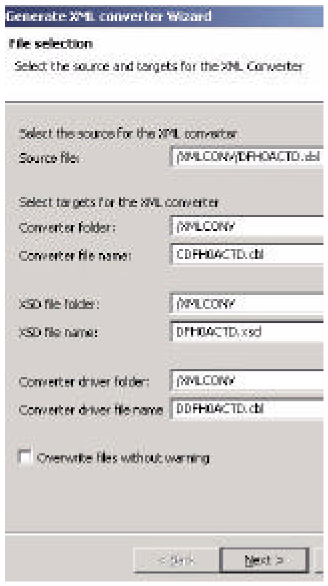
Early Availability Limitations

- Workbench
  - ►No online help
    - – ***XML for the Enterprise*** white paper
- Inbound message processing
  - ►Unicode UTF-16 is not supported
- Outbound message generation
  - ►Simple Occurs-Depending-On (ODO) not supported
  - ►Trailing/leading blanks in character content not removed
  - ►Trailing/leading zeroes in numeric content not removed
  - ►**<**, **>**, **'**, **"**, **&** not allowed in character content

Using the Generate XML Converter Wizard



```
DATA DIVISION.
WORKING-STORAGE SE
    EXEC SQL INCLU
01 TMP
01  SQL-MESSAGE
```



File selection

Select the source and targets for the XML Converter

Select the source for the XML converter

Source file:	/XMLCONV/DFHOACTD.cbl

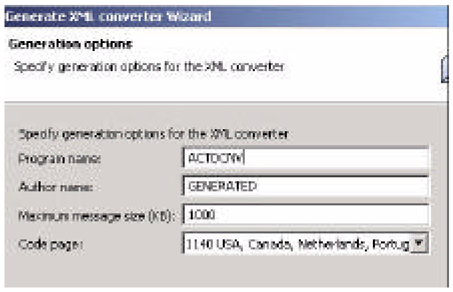
Select targets for the XML converter

Converter folder:	/XMLCONV
Converter file name:	CDFHOACTD.cbl
XSD file folder:	/XMLCONV
XSD file name:	DFHOACTD.xsd
Converter driver folder:	/XMLCONV
Converter driver file name	DDFHOACTD.cbl

☐ Overwrite files without warning

WebSphere software

Using the Generate XML Converter Wizard ...

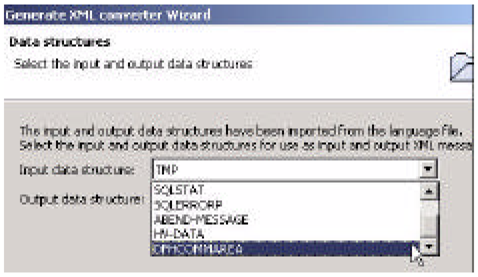


Generation options

Specify generation options for the XML converter

Specify generation options for the XML converter

Program name:	ACTOCNV
Author name:	GENERATED
Maximum message size (KB):	1000
Code page:	1140 USA, Canada, Netherlands, Portug

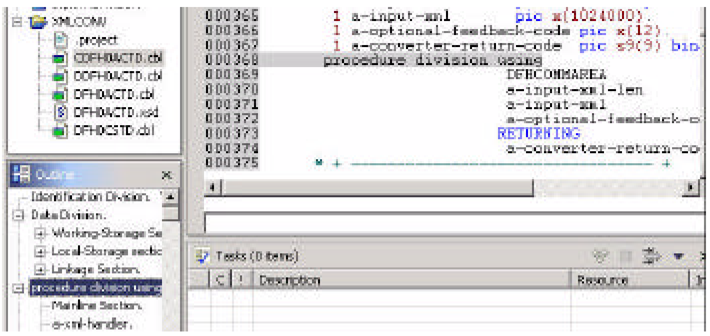


Data structures

Select the input and output data structures

The input and output data structures have been imported from the language file.
Select the input and output data structures for use as input and output XML messa

Input data structure:	TMP
Output data structure:	SQLSTAT SQLERRORP ABEND-MESSAGE HV-DATA DFHCOMMAREA



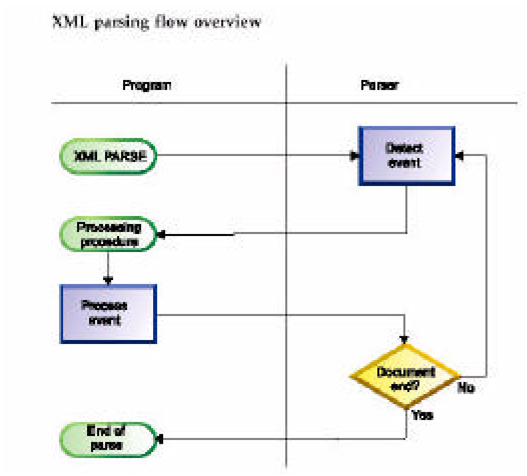
```
       1 a-input-xml          pic x(1024000).
       1 a-optional-feedback-code pic x(12).
       1 a-converter-return-code  pic s9(9) bin
      procedure division using
                       DFHCOMMAREA
                       a-input-xml-len
                       a-input-xml
                       a-optional-feedback-c
                      RETURNING
                       a-converter-return-co
```

WebSphere software

COBOL Compiler Support for XML

- Introduced with IBM Enterprise COBOL for z/OS and OS/390 V3R1
- High-speed XML parser
  - ►Consumes inbound XML messages
  - ►Verifies well-formedness
  - ►Transforms contents into COBOL data structures
  - ►Supports XML documents encoded in Unicode UTF-16, EBCDIC, ASCII
- New **XML PARSE** statement
  - ►Begins XML parse
  - ►Identifies document to be processed
  - ►Identifies processing procedure
- Processing procedure
  - ►Controls the parse
  - ►Receives and processes XML events
  - ►Handles exceptions

XML Parsing Flow



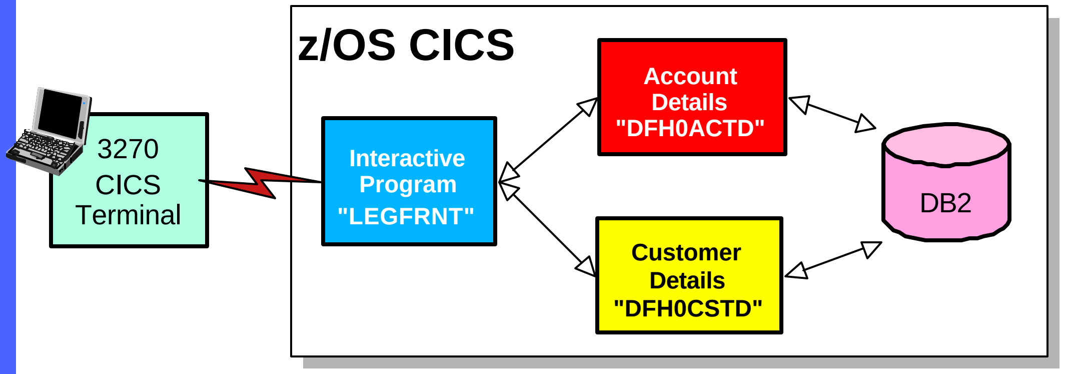
Parsing XML Documents

```
XML PARSE xml-document
        PROCESSING PROCEDURE xmlevent-handler
    ON EXCEPTION
        DISPLAY 'XML document error' XML-ERROR
        STOP RUN
    NOT ON EXCEPTION
        DISPLAY 'XML document was succesfully parsed.'
END-XML
```

XML Processing Procedure

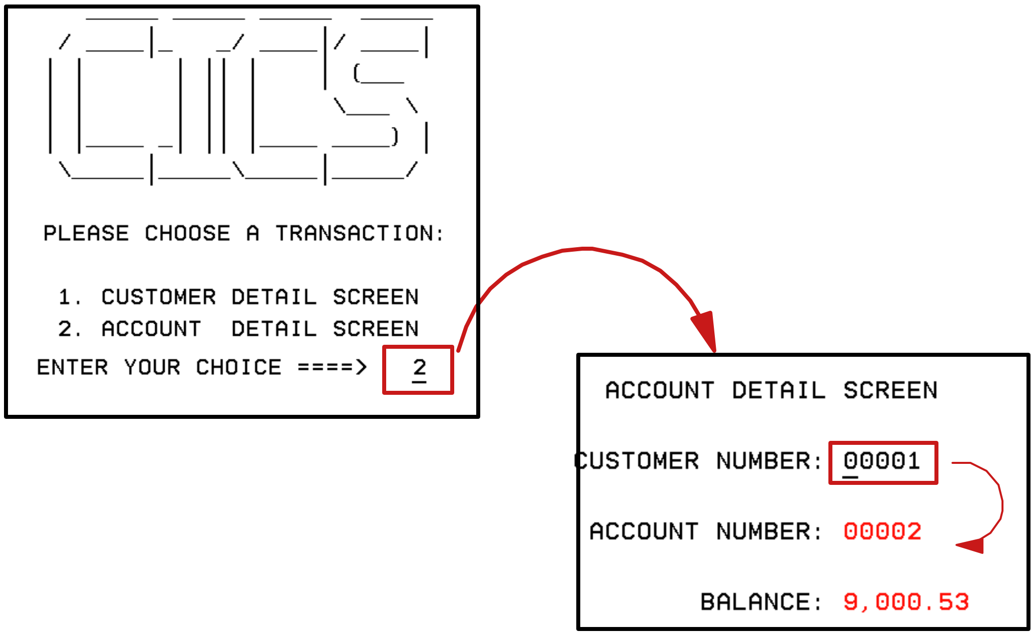
```
xmlevent-handler section.
   evaluate XML-EVENT
*==>Order XML events most frequent first
      when 'START-OF-ELEMENT'
         display 'Start elementtag:<'XML-TEXT '>'
         move XML-TEXT to current-element
      when 'CONTENT-CHARACTERS'
         display 'Content characters:<'XML-TEXT '>'
*==>Transform XML content to operational COBOL data item...
         evaluate current-element
            when 'listprice'
*==>Using function NUMVAL-C...
            compute list-price =function numval-c(XML-TEXT)
            when 'discount'
*==>Using de-editing of a numeric edited item...
            move XML-TEXT to xfr-ed
            move xfr-ed-1 to discount
         end-evaluate
      when 'END-OF-ELEMENT'
      ....
```

Sample application topology


z/OS CICS
3270 CICS Terminal
Interactive Program "LEGFRNT"
Account Details "DFH0ACTD"
Customer Details "DFH0CSTD"
DB2

WebSphere software

Running the existing 3270 CICS legacy application

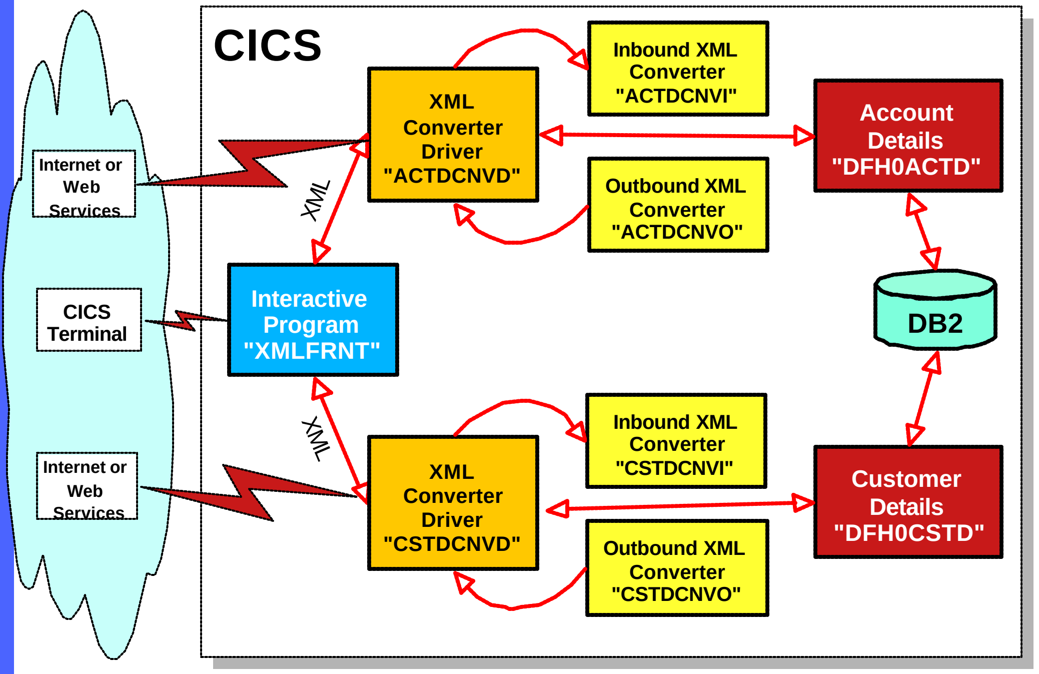


```
PLEASE CHOOSE A TRANSACTION:

 1. CUSTOMER DETAIL SCREEN
 2. ACCOUNT  DETAIL SCREEN
ENTER YOUR CHOICE ====>  2
```

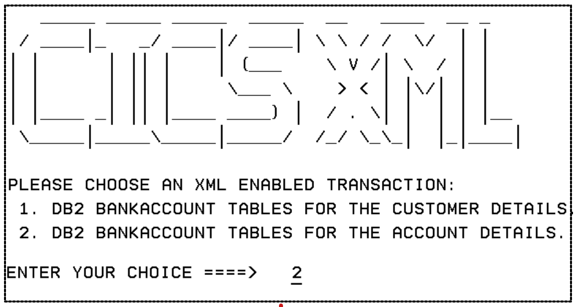
```
  ACCOUNT DETAIL SCREEN

CUSTOMER NUMBER: 00001

 ACCOUNT NUMBER: 00002

        BALANCE: 9,000.53
```

WebSphere software

Requirements for changing the existing application


Internet or Web Services
CICS Terminal
Internet or Web Services
CICS
XML Converter Driver "ACTDCNVD"
Inbound XML Converter "ACTDCNVI"
Outbound XML Converter "ACTDCNVO"
Account Details "DFH0ACTD"
XML
Interactive Program "XMLFRNT"
DB2
XML
XML Converter Driver "CSTDCNVD"
Inbound XML Converter "CSTDCNVI"
Outbound XML Converter "CSTDCNVO"
Customer Details "DFH0CSTD"

Running the XML enabled CICS legacy application


```
PLEASE CHOOSE AN XML ENABLED TRANSACTION:
 1. DB2 BANKACCOUNT TABLES FOR THE CUSTOMER DETAILS
 2. DB2 BANKACCOUNT TABLES FOR THE ACCOUNT DETAILS.

ENTER YOUR CHOICE ====>   2
```

```
<?xml version="1.0"?><message>  <custno>1</custno>  <acctno>0</acctno>  <balance
>0.0</balance></message>_
```

```
<?xml version="1.0"?><DFHCOMMAREA><custno> 00001</custno><acctno> 00002</acctno>
<balance> 9000.53</balance></DFHCOMMAREA>
```

Errors messages parsing input XML data

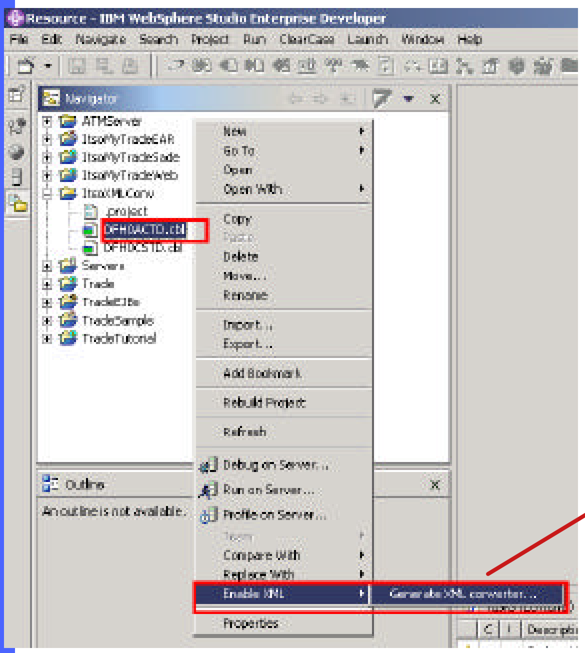
```
<?xml version="1.0"?><message> <custno>1</custno> <acctno>0</acctno> <balance
>0.0<balance></message>
```

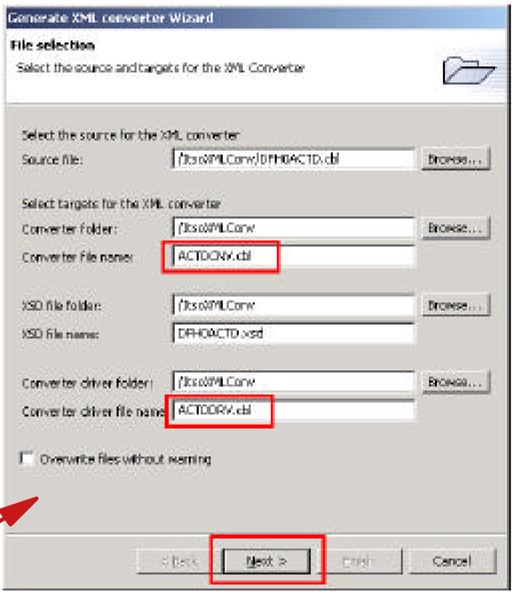
```
<?xml version="1.0"?><failureResponse> <errorMessageNumber>000000280</errorMessa
geNumber><errorCode>000000005</errorCode><errorMessage><!ÝCDATAÝ    IGZ0280S XML
 to data structure conversion could not complete in program "ACTDCNVI" because a
n error return code of 5 was received from the XML PARSE statement. The error oc
curred at element "balance" with the character content "???".
                    "">"</errorMessage>    </failureResponse>
```

```
<?xml version="1.0"?><message> <custno>xxx</custno> <acctno>0</acctno> <balan
ce>0.0</balance></message>
```

```
<?xml version="1.0"?><failureResponse> <errorMessageNumber>000000284</errorMessa
geNumber><errorCode>000000284</errorCode><errorMessage><!ÝCDATAÝ    IGZ0284S XML
 to data structure conversion could not complete in program "ACTDCNVI" because c
onversion of the character content of an element that is mapped as numeric faile
d. The error occurred at element "custno" with the character content "xxx".
                    "">"</errorMessage>    </failureResponse>
```

XML enable tools on WSED - Example





Select the source and targets for the XML Converter

Select the source for the XML converter

Source file: /ItsoXMLConv/DFH0ACTD.cbl Browse...

Select targets for the XML converter

Converter folder: /ItsoXMLConv Browse...

Converter file name: ACTDCNV.cbl

XSD file folder: /ItsoXMLConv Browse...

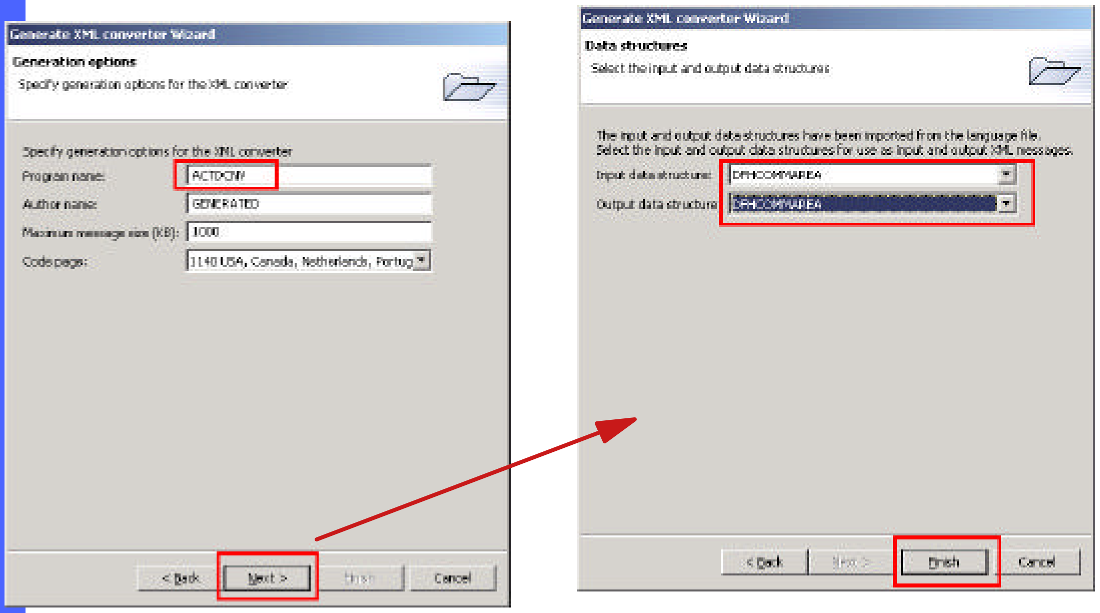
XSD file name: DFH0ACTD.xsd

Converter driver folder: /ItsoXMLConv Browse...

Converter driver file name: ACTDDRV.cbl

☐ Overwrite files without warning

XML enable tools on WSED - Example



Generate XML converter Wizard

Generation options

Specify generation options for the XML converter

Specify generation options for the XML converter

Program name:	ACTDCNV
Author name:	GENERATED
Maximum message size (KB):	1000
Code page:	1140 USA, Canada, Netherlands, Portug

Generate XML converter Wizard

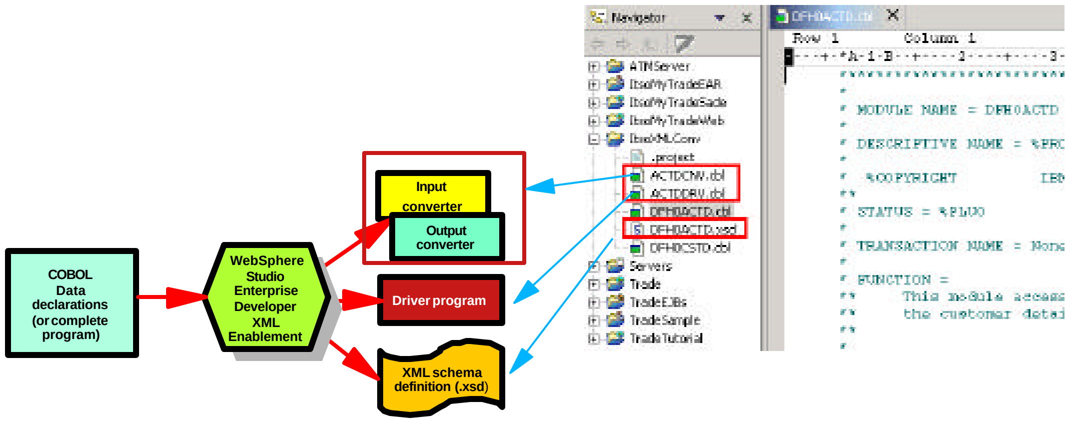
Data structures

Select the input and output data structures

The input and output data structures have been imported from the language file.
Select the input and output data structures for use as input and output XML messages.

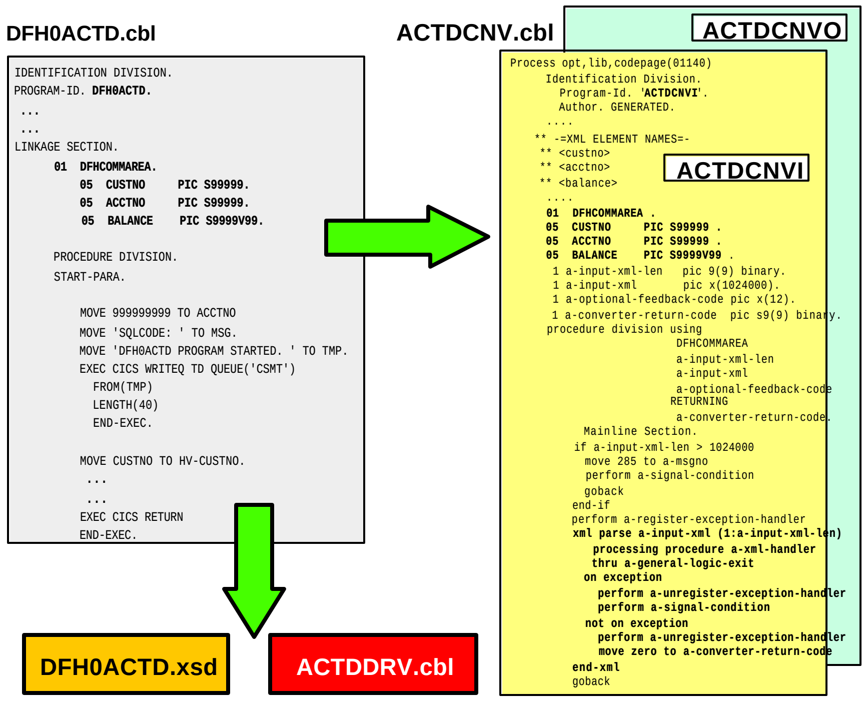
Input data structure:	DFHCOMMAREA
Output data structure:	DFHCOMMAREA

XML enable tools on WSED - Example...



```
Row 1          Column 1
----+-*A-1-B--+----2----+----3-
      ******************************
      *
      * MODULE NAME = DFH0ACTD
      *
      * DESCRIPTIVE NAME = %PRO
      *
      *  %COPYRIGHT           IBM
      **
      * STATUS = %PLUO
      *
      * TRANSACTION NAME = None
      *
      * FUNCTION =
      **      This module access
      **      the customer detai
      **
      *
```

Inbound converter - Example...



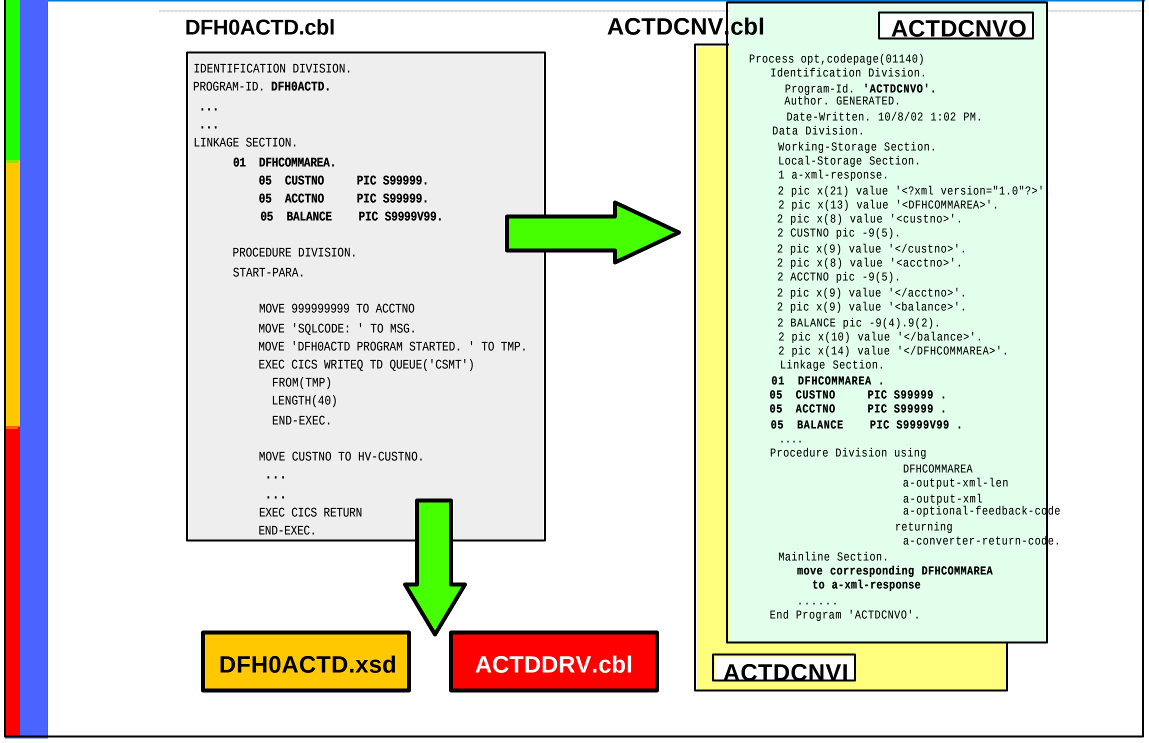
```
IDENTIFICATION DIVISION.
PROGRAM-ID. DFH0ACTD.
 ...
 ...
LINKAGE SECTION.
      01  DFHCOMMAREA.
          05  CUSTNO     PIC S99999.
          05  ACCTNO     PIC S99999.
          05  BALANCE    PIC S9999V99.

      PROCEDURE DIVISION.
      START-PARA.

          MOVE 999999999 TO ACCTNO
          MOVE 'SQLCODE: ' TO MSG.
          MOVE 'DFH0ACTD PROGRAM STARTED. ' TO TMP.
          EXEC CICS WRITEQ TD QUEUE('CSMT')
            FROM(TMP)
            LENGTH(40)
            END-EXEC.

          MOVE CUSTNO TO HV-CUSTNO.
           ...
           ...
          EXEC CICS RETURN
          END-EXEC.
```

```
Process opt,lib,codepage(01140)
      Identification Division.
        Program-Id. 'ACTDCNVI'.
        Author. GENERATED.
      ....
    ** -=XML ELEMENT NAMES=-
     ** <custno>
     ** <acctno>
     ** <balance>
      ....
      01  DFHCOMMAREA .
      05  CUSTNO     PIC S99999 .
      05  ACCTNO     PIC S99999 .
      05  BALANCE    PIC S9999V99 .
       1 a-input-xml-len   pic 9(9) binary.
       1 a-input-xml       pic x(1024000).
       1 a-optional-feedback-code pic x(12).
       1 a-converter-return-code  pic s9(9) binary.
      procedure division using
                          DFHCOMMAREA
                          a-input-xml-len
                          a-input-xml
                          a-optional-feedback-code
                         RETURNING
                          a-converter-return-code.
         Mainline Section.
        if a-input-xml-len > 1024000
         move 285 to a-msgno
         perform a-signal-condition
         goback
        end-if
        perform a-register-exception-handler
        xml parse a-input-xml (1:a-input-xml-len)
           processing procedure a-xml-handler
           thru a-general-logic-exit
          on exception
            perform a-unregister-exception-handler
            perform a-signal-condition
          not on exception
            perform a-unregister-exception-handler
            move zero to a-converter-return-code
        end-xml
        goback
```

Outbound converter - Parser used in converter

```
xml parse a-input-xml (1:a-input-xml-len)
        processing procedure a-xml-handler
        thru a-general-logic-exit
       on exception
         perform a-unregister-exception-handler
         perform a-signal-condition
      not on exception
         perform a-unregister-exception-handler
         move zero to a-converter-return-code
    end-xml
```

Outbound converter - Example...



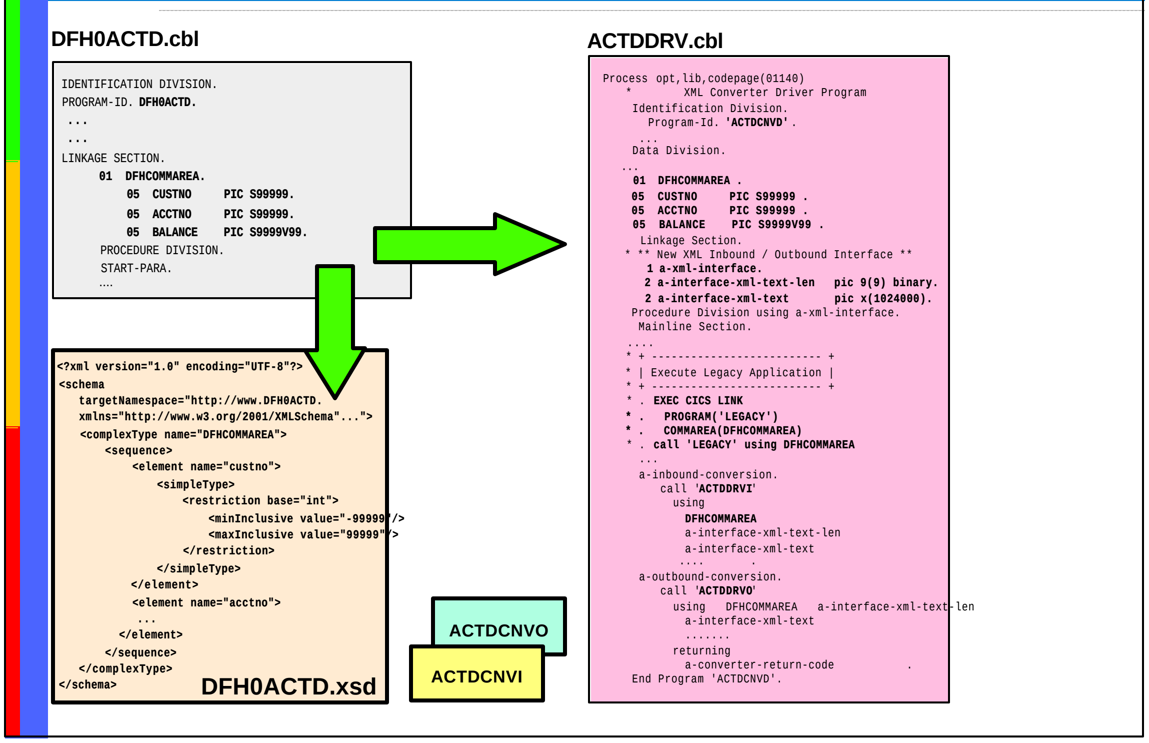
DFH0ACTD.cbl

```
IDENTIFICATION DIVISION.
PROGRAM-ID. DFH0ACTD.
 ...
 ...
LINKAGE SECTION.
      01  DFHCOMMAREA.
          05  CUSTNO     PIC S99999.
          05  ACCTNO     PIC S99999.
          05  BALANCE    PIC S9999V99.

      PROCEDURE DIVISION.
      START-PARA.

          MOVE 999999999 TO ACCTNO
          MOVE 'SQLCODE: ' TO MSG.
          MOVE 'DFH0ACTD PROGRAM STARTED. ' TO TMP.
          EXEC CICS WRITEQ TD QUEUE('CSMT')
            FROM(TMP)
            LENGTH(40)
            END-EXEC.

          MOVE CUSTNO TO HV-CUSTNO.
           ...
           ...
          EXEC CICS RETURN
          END-EXEC.
```

ACTDCNV.cbl — ACTDCNVO

```
Process opt,codepage(01140)
   Identification Division.
     Program-Id. 'ACTDCNVO'.
     Author. GENERATED.
     Date-Written. 10/8/02 1:02 PM.
   Data Division.
    Working-Storage Section.
    Local-Storage Section.
    1 a-xml-response.
    2 pic x(21) value '<?xml version="1.0"?>'.
    2 pic x(13) value '<DFHCOMMAREA>'.
    2 pic x(8) value '<custno>'.
    2 CUSTNO pic -9(5).
    2 pic x(9) value '</custno>'.
    2 pic x(8) value '<acctno>'.
    2 ACCTNO pic -9(5).
    2 pic x(9) value '</acctno>'.
    2 pic x(9) value '<balance>'.
    2 BALANCE pic -9(4).9(2).
    2 pic x(10) value '</balance>'.
    2 pic x(14) value '</DFHCOMMAREA>'.
    Linkage Section.
   01  DFHCOMMAREA .
   05  CUSTNO     PIC S99999 .
   05  ACCTNO     PIC S99999 .
   05  BALANCE    PIC S9999V99 .
    ....
   Procedure Division using
                     DFHCOMMAREA
                     a-output-xml-len
                     a-output-xml
                     a-optional-feedback-code
                    returning
                     a-converter-return-code.
    Mainline Section.
      move corresponding DFHCOMMAREA
        to a-xml-response
      ......
   End Program 'ACTDCNVO'.
```

Converter driver and XML schema - Example...



DFH0ACTD.cbl

```
IDENTIFICATION DIVISION.
PROGRAM-ID. DFH0ACTD.
 ...
 ...
LINKAGE SECTION.
     01  DFHCOMMAREA.
         05  CUSTNO     PIC S99999.
         05  ACCTNO     PIC S99999.
         05  BALANCE    PIC S9999V99.
     PROCEDURE DIVISION.
     START-PARA.
     ....
```

DFH0ACTD.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
   targetNamespace="http://www.DFH0ACTD.
   xmlns="http://www.w3.org/2001/XMLSchema"...">
   <complexType name="DFHCOMMAREA">
       <sequence>
           <element name="custno">
               <simpleType>
                   <restriction base="int">
                       <minInclusive value="-99999"/>
                       <maxInclusive value="99999"/>
                   </restriction>
               </simpleType>
           </element>
           <element name="acctno">
            ...
         </element>
       </sequence>
   </complexType>
</schema>
```

ACTDDRV.cbl

```
Process opt,lib,codepage(01140)
   *        XML Converter Driver Program
    Identification Division.
      Program-Id. 'ACTDCNVD' .
     ...
    Data Division.
   ...
    01  DFHCOMMAREA .
    05  CUSTNO     PIC S99999 .
    05  ACCTNO     PIC S99999 .
    05  BALANCE    PIC S9999V99 .
     Linkage Section.
   * ** New XML Inbound / Outbound Interface **
      1 a-xml-interface.
      2 a-interface-xml-text-len   pic 9(9) binary.
      2 a-interface-xml-text       pic x(1024000).
    Procedure Division using a-xml-interface.
     Mainline Section.
    ....
   * + ------------------------- +
   * | Execute Legacy Application |
   * + ------------------------- +
   * . EXEC CICS LINK
   * .   PROGRAM('LEGACY')
   * .   COMMAREA(DFHCOMMAREA)
   * . call 'LEGACY' using DFHCOMMAREA
     ...
     a-inbound-conversion.
         call 'ACTDDRVI'
           using
             DFHCOMMAREA
             a-interface-xml-text-len
             a-interface-xml-text
            ....       .
     a-outbound-conversion.
         call 'ACTDDRVO'
           using   DFHCOMMAREA   a-interface-xml-text-len
             a-interface-xml-text
             .......
           returning
             a-converter-return-code           .
    End Program 'ACTDCNVD'.
```

Modifying the converter driver programs - Example

```
 Process opt,lib,codepage(01140),cics
* XML Converter Driver Program *
 Identification Division.
  Program-Id. 'ACTDCNVD'.
  ....
  Data Division.
  ....
* ** Legacy Application Inbound / Outbound Binary Interface **
* ***************************************************************
 01  ~~DFHCOMMAREA~~ BUSINESS-DATASTRUCT .
 05  CUSTNO     PIC S99999 .
 05  ACCTNO     PIC S99999 .
 05  BALANCE    PIC S9999V99 .
  Linkage Section.
* *******************************************
* ** New XML Inbound / Outbound Interface **
* *******************************************
   1 ~~a-xml-interface~~ DFHCOMMAREA .
   2 a-interface-xml-text-len   pic 9(9) binary.
   2 a-interface-xml-text       pic x(1024000).
```

Modifying the converter driver programs - Example

```
  Procedure Division using ~~a-xml-interface~~ DFHCOMMAREA.
  Mainline Section.
  ,....
             move a-failure-response
      to a-interface-xml-text(1:a-interface-xml-text-len)
     perform a-unregister-exception-handler
    ~~goback~~
     exec cis return
     end-exec
    end-if
* + ------------------------- +
 * | Execute Legacy Application |
* + ------------------------- +
* . EXEC CICS LINK
* .   PROGRAM('LEGACY')
 * .    COMMAREA(DFHCOMMAREA)
  * . END-EXEC ...OR
* .
 * . call 'LEGACY' using DFHCOMMAREA
   exec cics link
     program('DFH0ACTD')
     commarea(BUSINESS-DATASTRUCT)
   end-exec
* + ------------------------------- +
 * | Execute Outbound XML Transformer |
* + ------------------------------- +
    perform a-outbound-conversion
* + --------------------------- +
 * | Unregister Exception Handler |
* + --------------------------- +
    perform a-unregister-exception-handler
* + -------- +
* | Finished |
* + -------- +
   ~~goback~~
     exec cis return
     end-exec
```

Summary

- Facilitate enterprise modernization by refacing existing COBOL applications to support XML messages
- Achieve significant productivity gains by utilizing the converter and driver template generators
- Gain performance benefits by running XML parsing and conversions on the z/OS systems
- Reduce Total Cost of Ownership (TCO) by having one development environment

WebSphere software

IBM

e business software

WebSphere Studio Asset Analyzer V2

Rapid Impact Analysis and Component Reuse for e-business

IBM Software Group

Enterprise Modernization

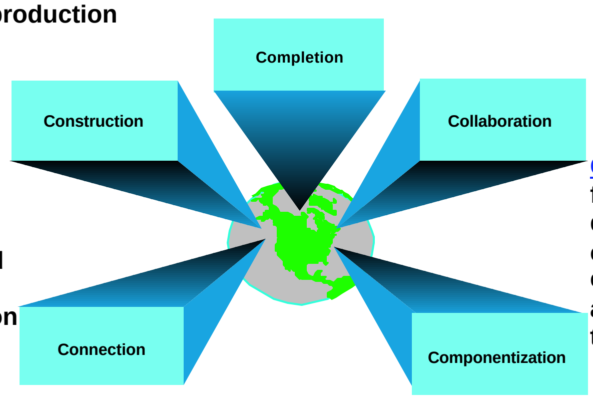
- What is it?
  - ►Enterprise Modernization is the liberation of core business assets to satisfy new e-business application development
- Why modernize?
  - ►Reuse is cheaper than writing new
  - ►Bridge the development skills gap with a common toolset for both Web and Legacy programmers. (1.3 Milion COBOL developers)
  - ►Leverage existing systems, applications and skills to create Dynamic e-business with excellent Returns on Investment.
  - ►Accelerate the e-business adoption for competitive advantage
- Why IBM?
  - ►IBM customers have significant business knowledge invested in enterprise systems (over 180 billion LOC and 5 Billion new each year)
  - ►Customers want to leverage Qualities of Service built in enterprise systems
  - ►Web Services makes it easier to leverage these assets today

Enterprise Modernization Challenges/Hurdles

- Legacy not ready for integration into Web Applications
  - ► Tedious and costly manual analysis and harvesting
- Scarcity of Skills and Steep learning curve
  - ► Complex new/emerging technologies
  - ► Massive amounts of traditional technologies
  - ► Need to include new developer communities
- Multiple Artifacts
  - ► More complex Application design
  - ► Fragmented development process
  - ► Multiple point tools and multiple skills must be in place
- Clashes between development groups
  - ► COBOL developers know the enterprise applications, hold the business knowledge, Java developers have the web knowledge
- Too much backlog and not enough time to deliver

Enterprise Modernization Strategy

Completion - speeds the movement of applications from the development process through system test to production



Construction - provides visual tooling to include traditional and business oriented developers in the delivery of mission critical J2EE applications

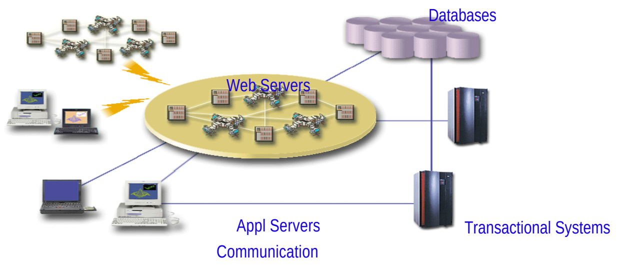
Collaboration - facilitates team development of component based e-business applications across the enterprise

Connection - helps connect and reuse legacy enterprise applications for e-business by using connectors

Componentization - promotes the transformation of legacy Enterprise Applications into components and the integration of these components into new e-business applications.

The Vision

- Deliver ***knowledge*** across applications to ***all*** enterprise developers by enabling
  - ►rapid application understanding
  - ►rapid application change
  - ►rapid application reuse including componentization



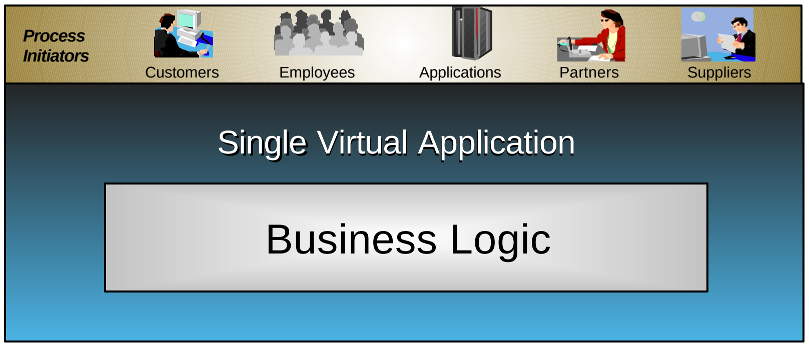
Goal: providing business knowledge through all phases of the development lifecycle

WebSphere software

IBM

Assumptions

- **A typical site has invested 100M in S/390 applications**
  - ►50 developers x 100K loaded cost x 20 years
- **Sites want to reuse these assets**
  - ►Too much spent on Y2K refurbishment to throw away
  - ►Reuse will get you to market faster withhigher quality
- **Internet, batch, and integration to core processing are requirements for success**

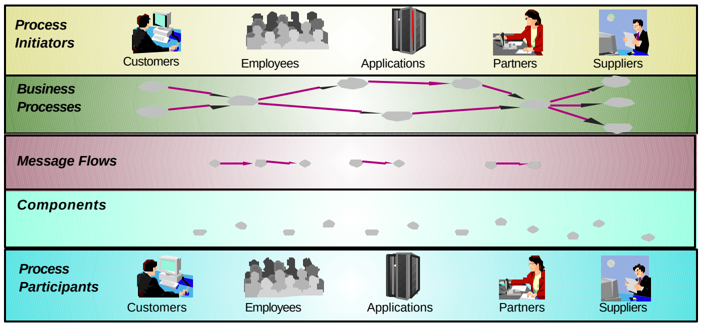


WebSphere software

IBM

Integration is Key

- There are two ways to integrate or connect existing applications
  - via information presented in a UI (user interface)
  - via well defined interfaces or connectors-Service Oriented Architectures*(* Gartner Group Definition)
- For both however, a detailed understanding of how information moves through the application or is processed by the connector is required
- Web enablement requires separation of UI, Control, and Data Processes



e-Business Change/Scenario

- Goal
  - Deliver the capability for a customer to do a direct lookup of their order status.
- **The Application**
  - Create a web application from an existing 3270 CICS or IMS based application using WebSphere Studio Asset Analyzer and VisualAge Enterprise Suite Tooling or WebSphere Studio Enterprise Developer
- **The Players**
  - **System Analyst/Project Lead**
  - **CICS COBOL/PL/I Developers**
  - **eBusiness Developers**
  - **QA Specialists**
  - **Managers**

WebSphere Studio Asset Analyzer

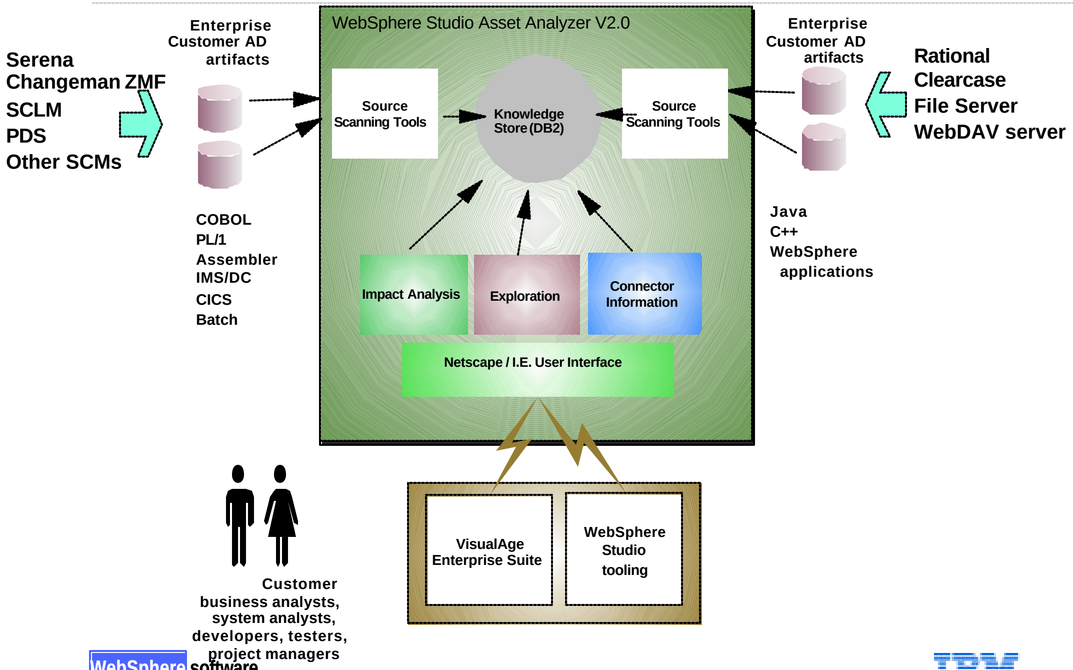
End-to-End Asset Analysis

- End to End zOS and Distributed Infrastructure.
- End to End Understanding and Community
- End to End Component Identification and Reuse
- End to End Development and Process

WebSphere software

IBM

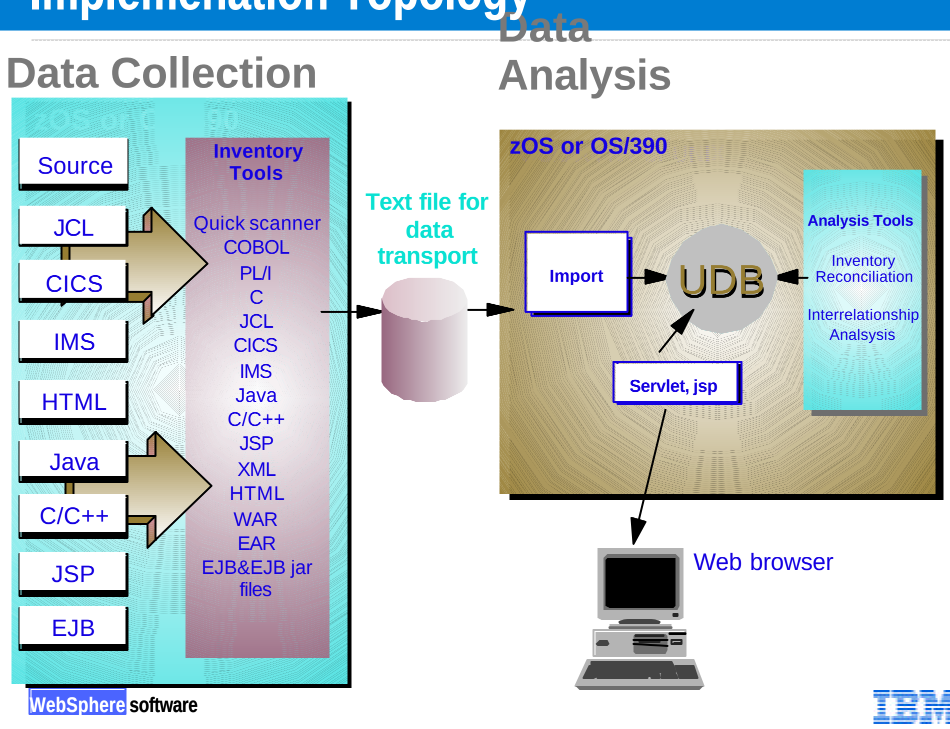
End-to-End zOS and Distributed Infrastructure



WebSphere software

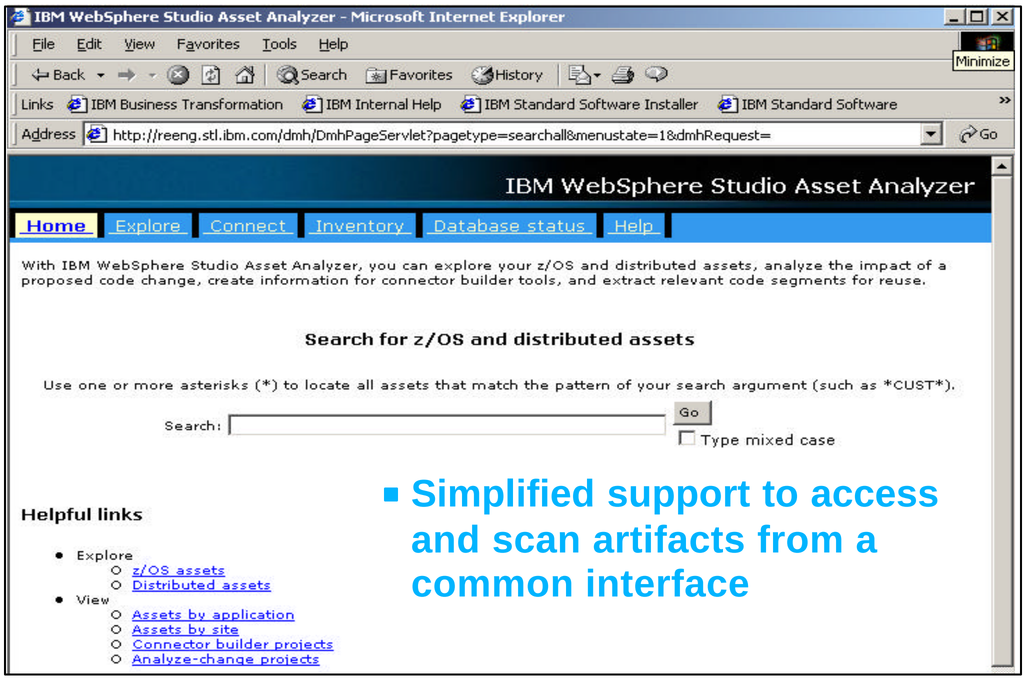
IBM

Implemenation Topology



IBM

End-to-End Understanding and Community


IBM WebSphere Studio Asset Analyzer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Minimize
Back Search Favorites History
Links IBM Business Transformation IBM Internal Help IBM Standard Software Installer IBM Standard Software
Address http://reeng.stl.ibm.com/dmh/DmhPageServlet?pagetype=searchall&menustate=1&dmhRequest=
Go
IBM WebSphere Studio Asset Analyzer
Home Explore Connect Inventory Database status Help

With IBM WebSphere Studio Asset Analyzer, you can explore your z/OS and distributed assets, analyze the impact of a proposed code change, create information for connector builder tools, and extract relevant code segments for reuse.

Search for z/OS and distributed assets

Use one or more asterisks (*) to locate all assets that match the pattern of your search argument (such as *CUST*).

Search: [ ] Go

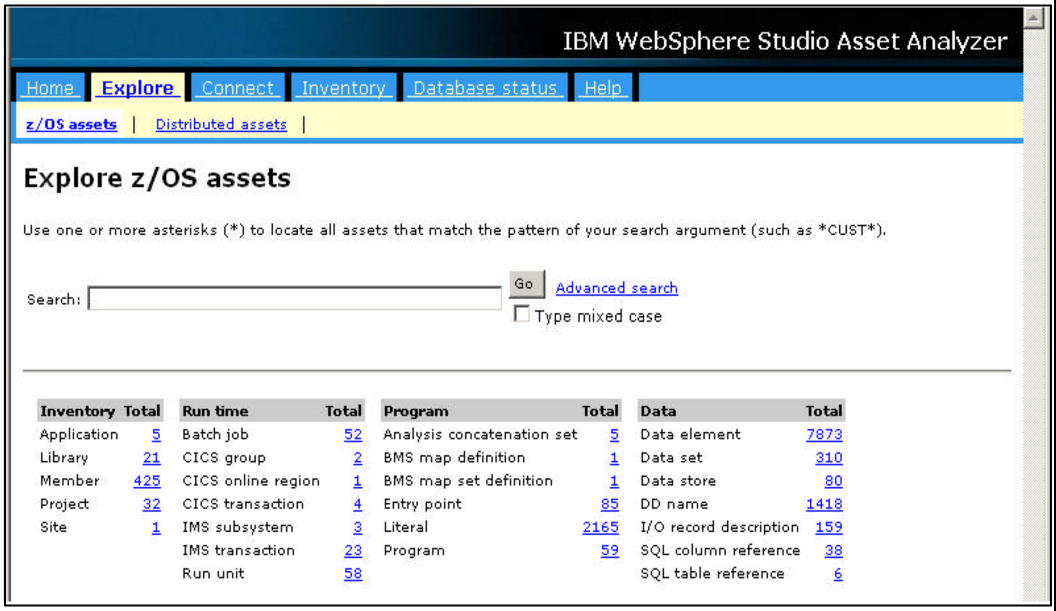
☐ Type mixed case

Helpful links

- Explore
  - z/OS assets
  - Distributed assets
- View
  - Assets by application
  - Assets by site
  - Connector builder projects
  - Analyze-change projects

- Simplified support to access and scan artifacts from a common interface

Understanding z/OS assets



IBM WebSphere Studio Asset Analyzer

Home | Explore | Connect | Inventory | Database status | Help

z/OS assets | Distributed assets |

Explore z/OS assets

Use one or more asterisks (*) to locate all assets that match the pattern of your search argument (such as *CUST*).

Search: [ ] Go Advanced search

☐ Type mixed case

Inventory	Total	Run time	Total	Program	Total	Data	Total
Application	5	Batch job	52	Analysis concatenation set	5	Data element	7873
Library	21	CICS group	2	BMS map definition	1	Data set	310
Member	425	CICS online region	1	BMS map set definition	1	Data store	80
Project	32	CICS transaction	4	Entry point	85	DD name	1418
Site	1	IMS subsystem	3	Literal	2165	I/O record description	159
		IMS transaction	23	Program	59	SQL column reference	38
		Run unit	58			SQL table reference	6

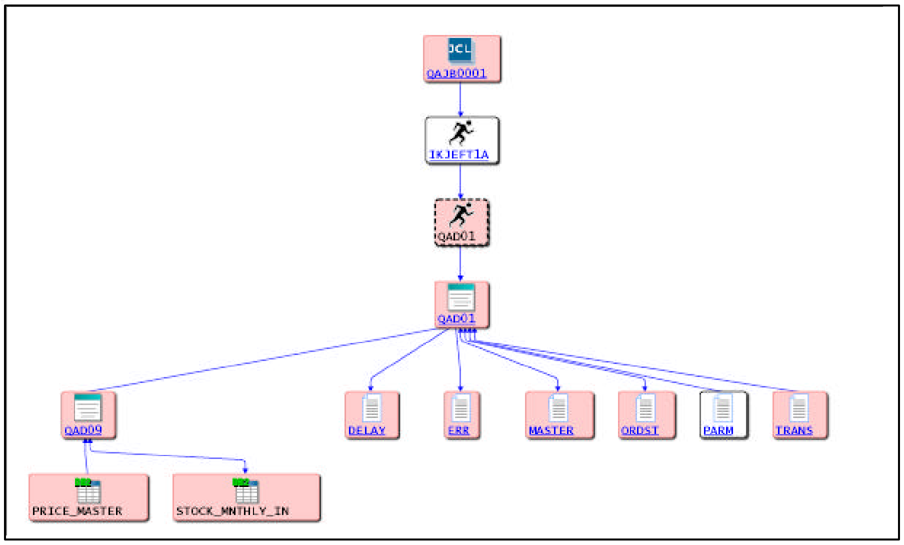
WebSphere software

IBM

z/OS Visualization

Graph actions

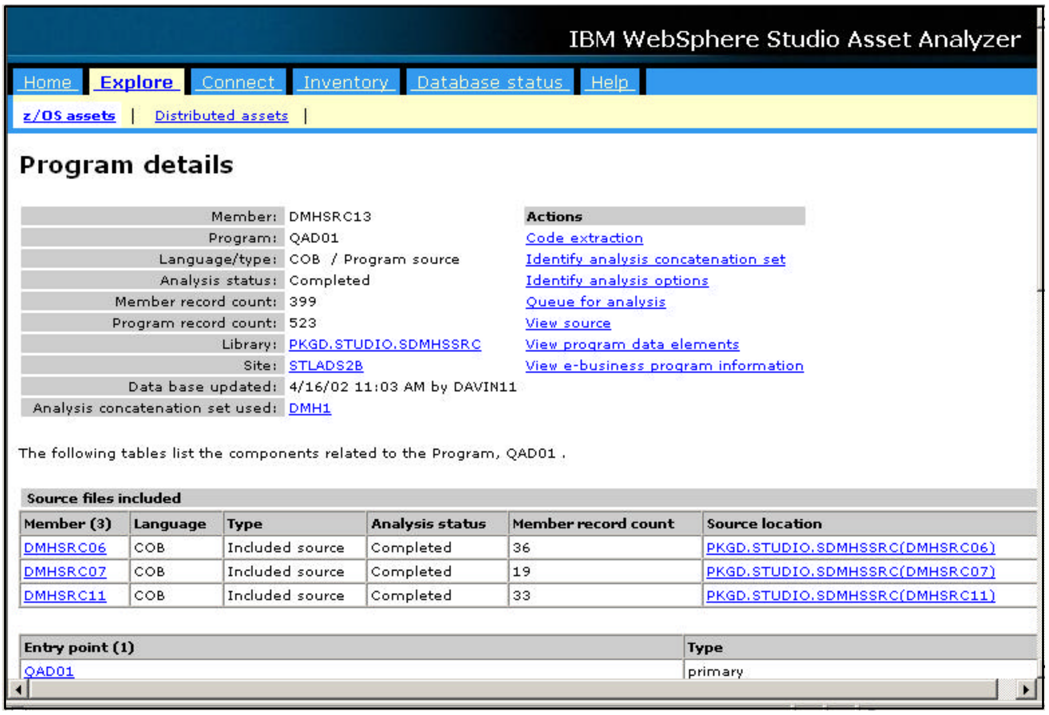
Zoom in Zoom out Zoom all



WebSphere software

IBM

z/OS Program Details



IBM WebSphere Studio Asset Analyzer
Home | Explore | Connect | Inventory | Database status | Help
z/OS assets | Distributed assets |

Program details

Member:	DMHSRC13
Program:	QAD01
Language/type:	COB / Program source
Analysis status:	Completed
Member record count:	399
Program record count:	523
Library:	PKGD.STUDIO.SDMHSSRC
Site:	STLADS2B
Data base updated:	4/16/02 11:03 AM by DAVIN11
Analysis concatenation set used:	DMH1

Actions
- Code extraction
- Identify analysis concatenation set
- Identify analysis options
- Queue for analysis
- View source
- View program data elements
- View e-business program information

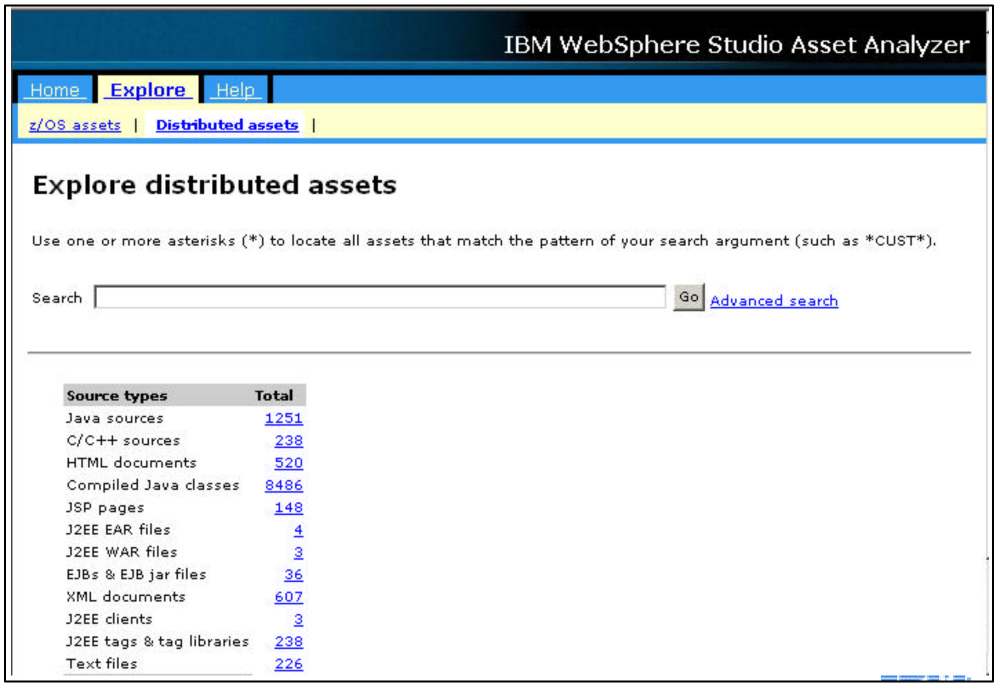
The following tables list the components related to the Program, QAD01 .

Source files included

Member (3)	Language	Type	Analysis status	Member record count	Source location
DMHSRC06	COB	Included source	Completed	36	PKGD.STUDIO.SDMHSSRC(DMHSRC06)
DMHSRC07	COB	Included source	Completed	19	PKGD.STUDIO.SDMHSSRC(DMHSRC07)
DMHSRC11	COB	Included source	Completed	33	PKGD.STUDIO.SDMHSSRC(DMHSRC11)

Entry point (1)	Type
QAD01	primary

Understanding Distributed Assets



IBM WebSphere Studio Asset Analyzer
Home | Explore | Help
z/OS assets | Distributed assets |

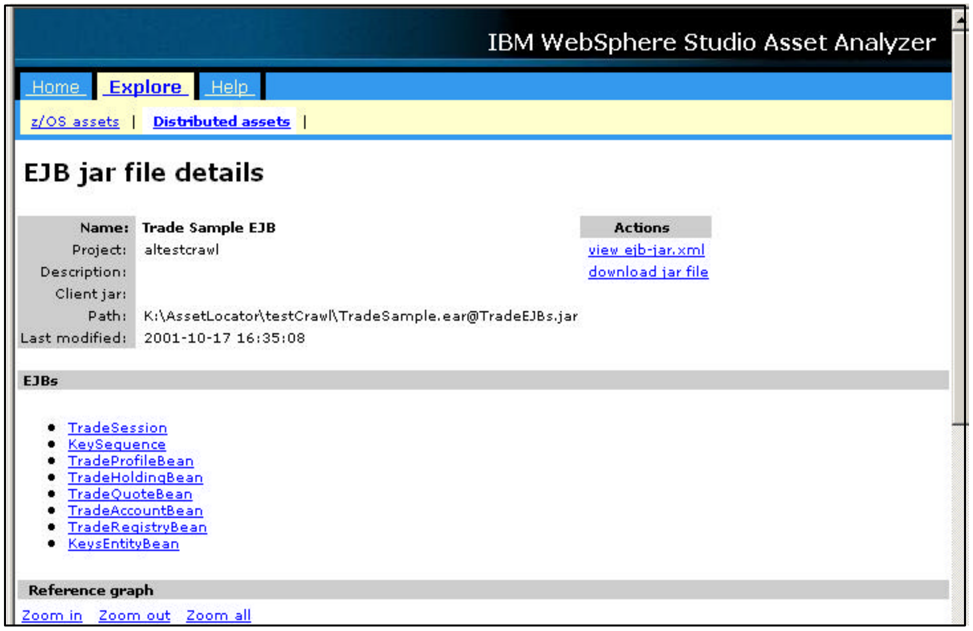
Explore distributed assets

Use one or more asterisks (*) to locate all assets that match the pattern of your search argument (such as *CUST*).

Search
Go
Advanced search

Source types	Total
Java sources	1251
C/C++ sources	238
HTML documents	520
Compiled Java classes	8486
JSP pages	148
J2EE EAR files	4
J2EE WAR files	3
EJBs & EJB jar files	36
XML documents	607
J2EE clients	3
J2EE tags & tag libraries	238
Text files	226

Distributed Asset Details



IBM WebSphere Studio Asset Analyzer

Home | Explore | Help

z/OS assets | Distributed assets |

EJB jar file details

		Actions
Name:	Trade Sample EJB	view ejb-jar.xml
Project:	altestcrawl	download jar file
Description:		
Client jar:		
Path:	K:\AssetLocator\testCrawl\TradeSample.ear@TradeEJBs.jar	
Last modified:	2001-10-17 16:35:08	

EJBs

- TradeSession
- KeySequence
- TradeProfileBean
- TradeHoldingBean
- TradeQuoteBean
- TradeAccountBean
- TradeRegistryBean
- KeysEntityBean

Reference graph

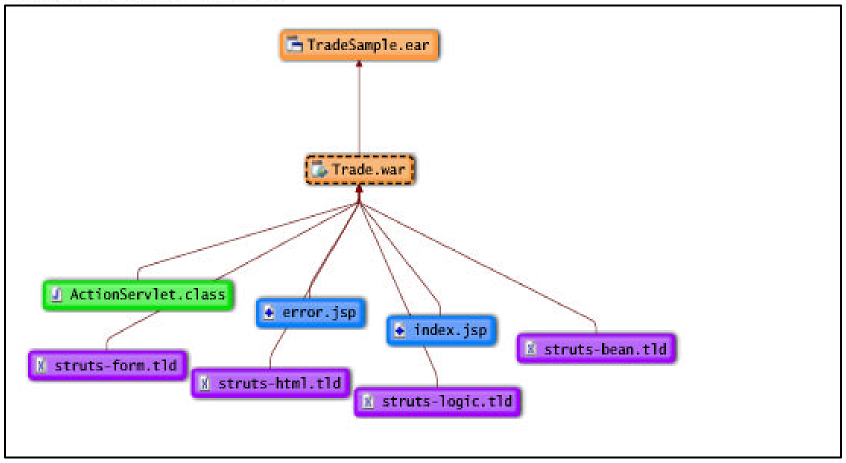
Zoom in   Zoom out   Zoom all

WebSphere software

Distributed Asset Visualization

Reference graph

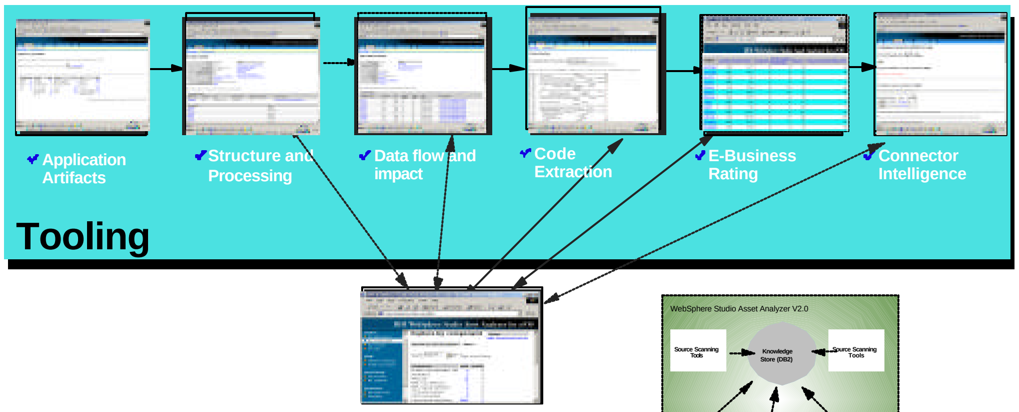
Zoom in   Zoom out   Zoom all



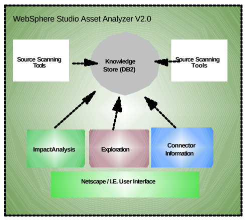
WebSphere software

WebSphere Studio Asset Analyzer V2

Discovery/Plan	Application Knowledge	Business Knowledge	Change Knowledge	Data Knowledge	E-Business Knowledge
• Understand business requirements • Define application boundaries	• Understand components • Define relationships • Find Transactions • Find JCL • Narrow Scope	• Define program flows • Define data items • Create data flows	• Generate statement of work • Create schedule	• Find Databases • Find Files • Find Relationships • Generate test cases	• Define candidate connectors • Generate



WebSphere Studio Asset Analyzer



WebSphere software

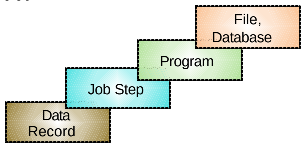
Application Knowlege

Application Knowledge	List Artifacts	Perform initial scope	Reduce Scope
• Get problem space • Research solutions • Get training • Find parts	• Define application • Get reusable parts • Define interactions points between application	• Define artifacts of interest • Eliminate non-affected artifacts	• Find Transactions • Find JCL • Add or subtract artifacts as necessary

Used by
- System Analyst/Project Lead
- CICS/IMS Developers
- COBOL/PL/I Programmers
- eBusiness programmer
- Business Analysis
- Managers
- QA Analysts

To
- Initially define knowledge or impact
- Iteratively reduce scope
- Search for information
- Communication
- Workflow
- Training and product documentation

Output:
- Impact analysis project
- Knowledge of application artifacts or parts
- Definition of scope



WebSphere software

Business Knowledge

Business Knowledge
- Define process understanding

Identify process flows
- Define business processing of interest

Perform process scope
- Define artifacts of interest
- Eliminate non-affected artifacts

Reduction of scope
- Find Transactions
- Find JCL
- Add or subtract artifacts as necessary

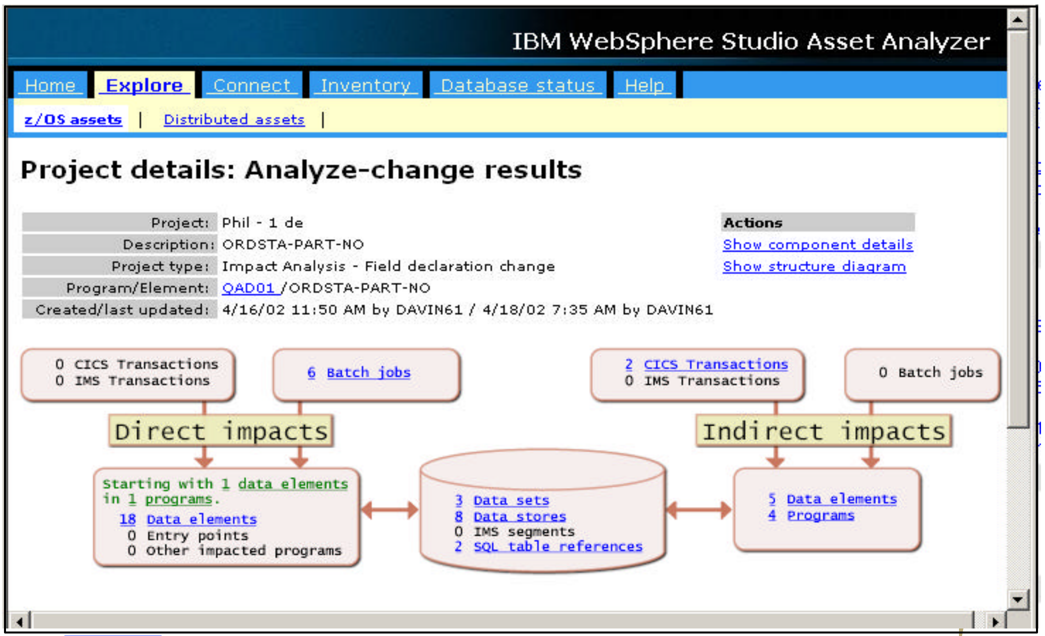
Used by
- System Analyst/Project Lead
- CICS/IMS Developers
- COBOL/PL/I Programmers
- eBusiness programmer
- Business Analysis
- Managers
- QA Analysts

To
- Initially define business processing
- Iteratively reduce scope
- Search for information
- Communication
- Workflow
- Training and product documentation

WebSphere software

IBM

Impact Analysis


IBM WebSphere Studio Asset Analyzer

Home | Explore | Connect | Inventory | Database status | Help

z/OS assets | Distributed assets |

Project details: Analyze-change results

Project:	Phil - 1 de
Description:	ORDSTA-PART-NO
Project type:	Impact Analysis - Field declaration change
Program/Element:	QAD01 /ORDSTA-PART-NO
Created/last updated:	4/16/02 11:50 AM by DAVIN61 / 4/18/02 7:35 AM by DAVIN61

Actions
Show component details
Show structure diagram

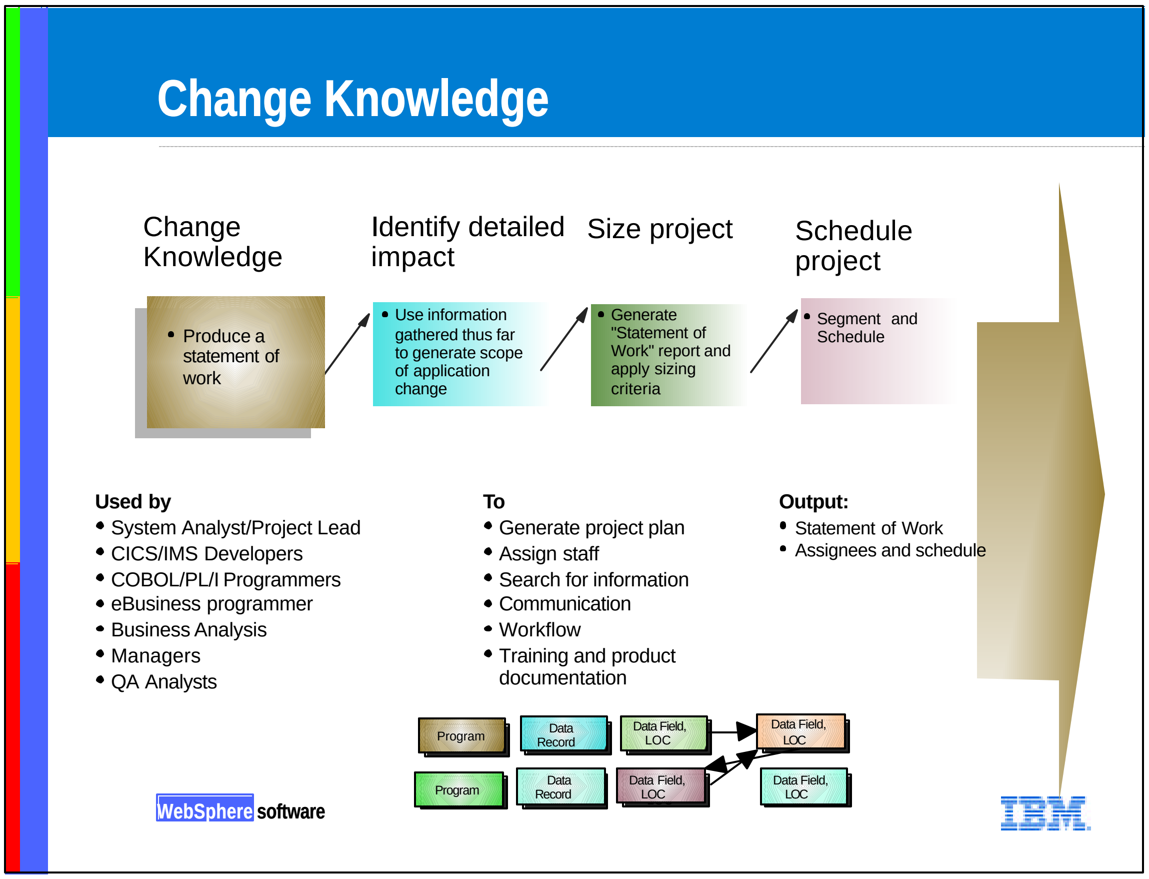
Direct impacts:
- 0 CICS Transactions
- 0 IMS Transactions
- 6 Batch jobs
- Starting with 1 data elements in 1 programs.
- 18 Data elements
- 0 Entry points
- 0 Other impacted programs

- 3 Data sets
- 8 Data stores
- 0 IMS segments
- 2 SQL table references

Indirect impacts:
- 2 CICS Transactions
- 0 IMS Transactions
- 0 Batch jobs
- 5 Data elements
- 4 Programs

WebSphere software

IBM



Change Knowledge

Change Knowledge
- Produce a statement of work

Identify detailed impact
- Use information gathered thus far to generate scope of application change

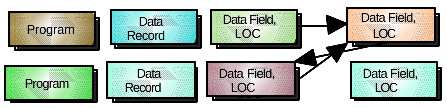
Size project
- Generate "Statement of Work" report and apply sizing criteria

Schedule project
- Segment and Schedule

Used by
- System Analyst/Project Lead
- CICS/IMS Developers
- COBOL/PL/I Programmers
- eBusiness programmer
- Business Analysis
- Managers
- QA Analysts

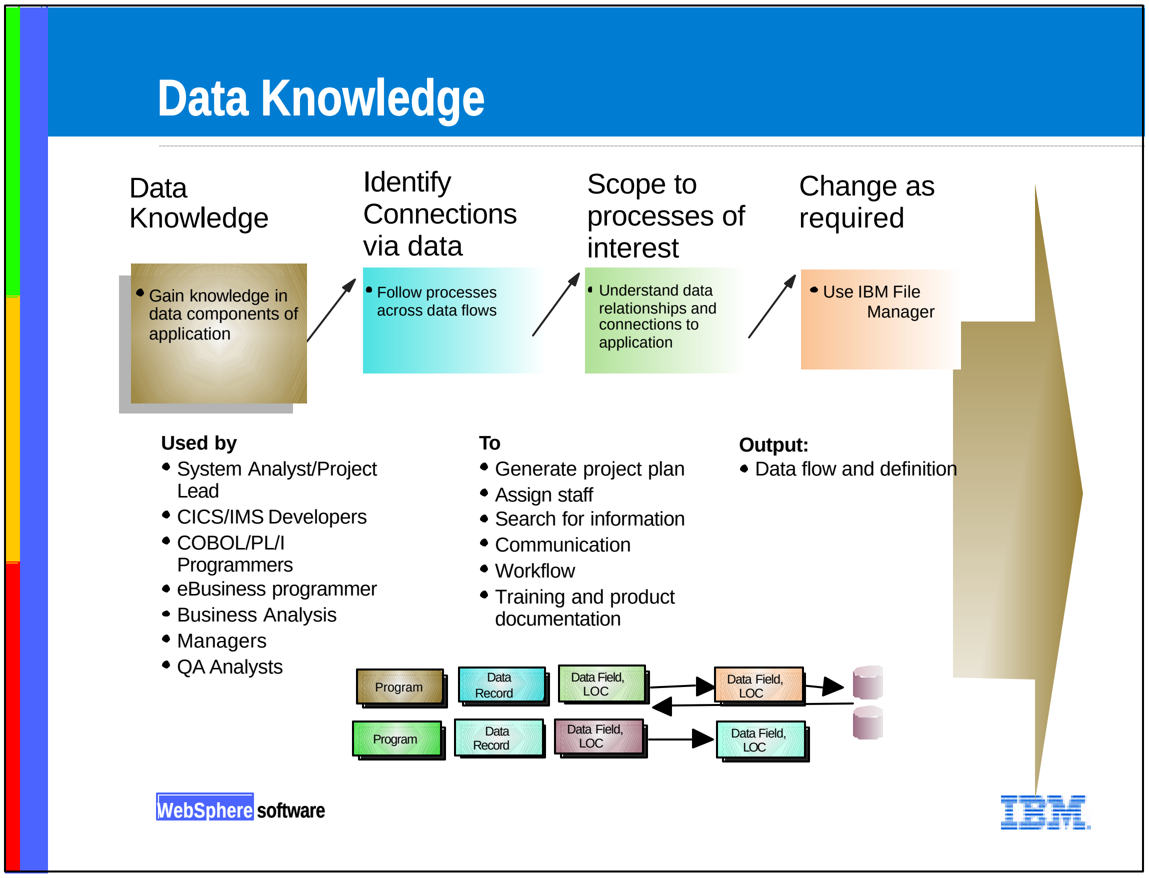
To
- Generate project plan
- Assign staff
- Search for information
- Communication
- Workflow
- Training and product documentation

Output:
- Statement of Work
- Assignees and schedule



WebSphere software

IBM



Data Knowledge

Data Knowledge
- Gain knowledge in data components of application

Identify Connections via data
- Follow processes across data flows

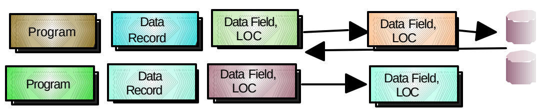
Scope to processes of interest
- Understand data relationships and connections to application

Change as required
- Use IBM File Manager

Used by
- System Analyst/Project Lead
- CICS/IMS Developers
- COBOL/PL/I Programmers
- eBusiness programmer
- Business Analysis
- Managers
- QA Analysts

To
- Generate project plan
- Assign staff
- Search for information
- Communication
- Workflow
- Training and product documentation

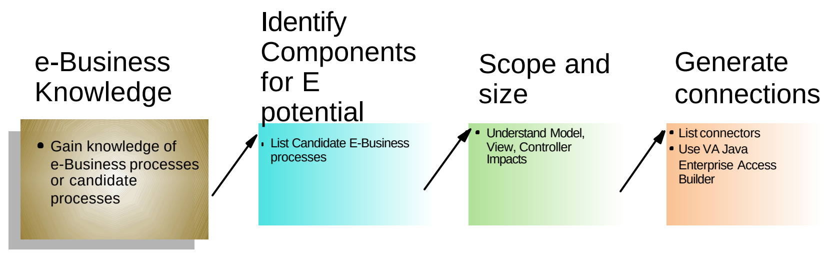
Output:
- Data flow and definition



WebSphere software

IBM

End to End Component Identification and Reuse


e-Business Knowledge
Gain knowledge of e-Business processes or candidate processes
Identify Components for E potential
List Candidate E-Business processes
Scope and size
Understand Model, View, Controller Impacts
Generate connections
List connectors
Use VA Java Enterprise Access Builder

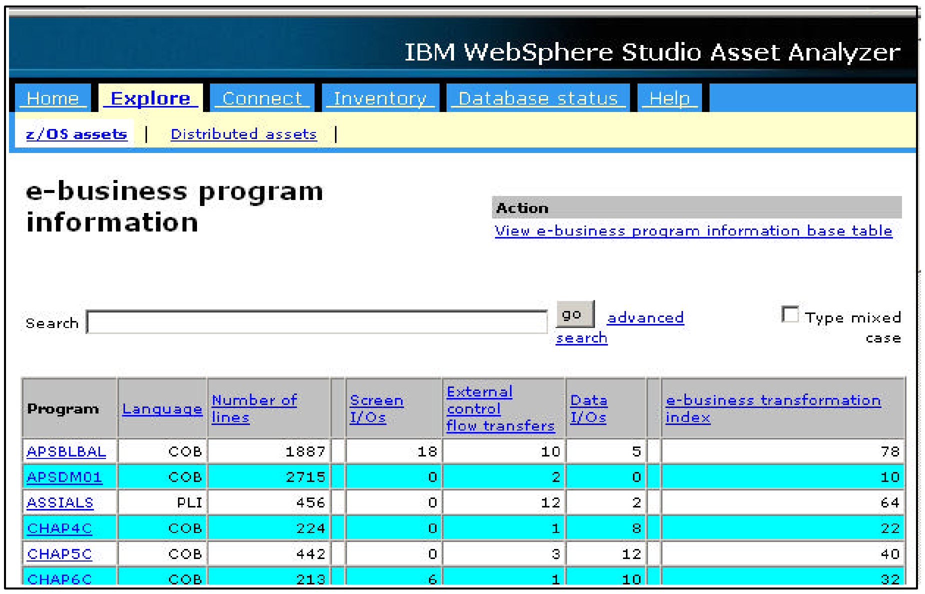
Used by
- System Analyst/Project Lead
- CICS Developers
- COBOL Programmers
- eBusiness programmer
- Business Analysis
- Managers
- QA Analysts

To
- Search for information
- Provide sizings and evaluation for e-business
- Generate connectors
- Communication
- Workflow
- Training and product documentation

Output:
- Connector Understanding
- Connector Generation

View
Controller
Model
Connector

e-business Ratings



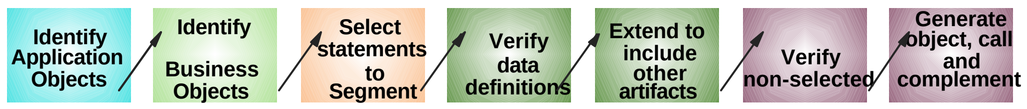
e-business program information

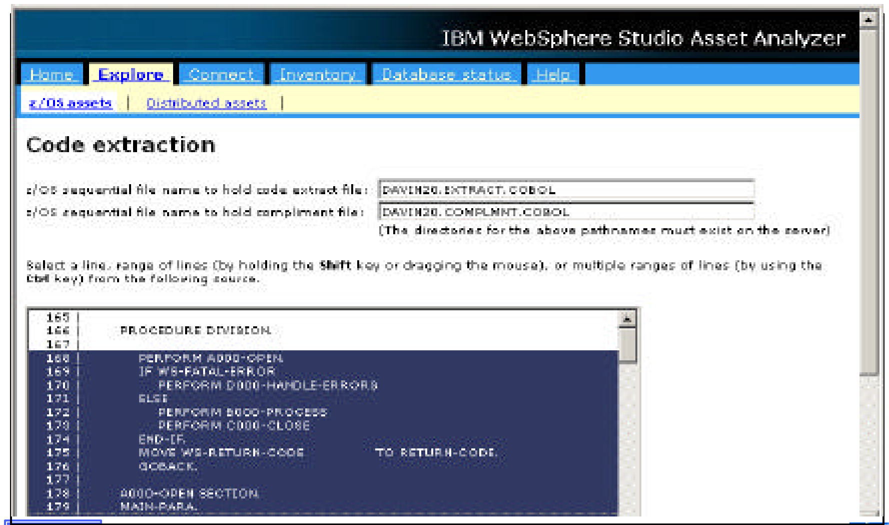
Action
View e-business program information base table

Search

Program	Language	Number of lines	Screen I/Os	External control flow transfers	Data I/Os	e-business transformation index
APSBLBAL	COB	1887	18	10	5	78
APSDM01	COB	2715	0	2	0	10
ASSIALS	PLI	456	0	12	2	64
CHAP4C	COB	224	0	1	8	22
CHAP5C	COB	442	0	3	12	40
CHAP6C	COB	213	6	1	10	32

End to End Component Identification and Reuse


Identify Application Objects
Identify Business Objects
Select statements to Segment
Verify data definitions
Extend to include other artifacts
Verify non-selected
Generate object, call and complement



Code extraction

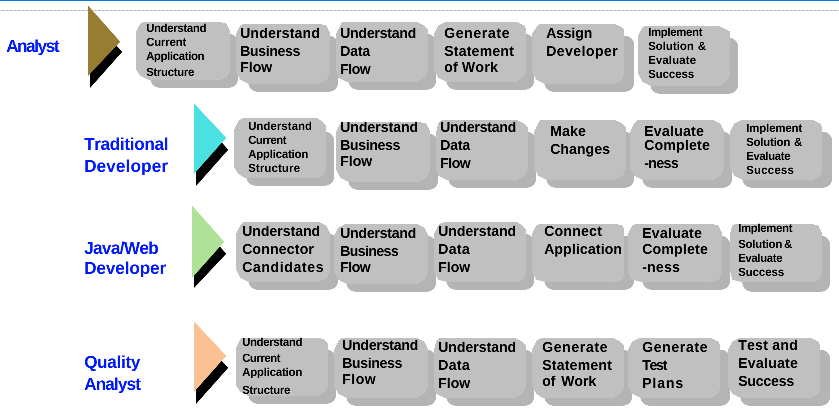
z/OS sequential file name to hold code extract file: DAVIN20.EXTRACT.COBOL
z/OS sequential file name to hold compliment file: DAVIN20.COMPLMNT.COBOL
(The directories for the above pathnames must exist on the server)

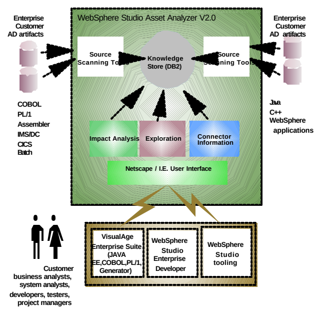
Select a line, range of lines (by holding the **Shift** key or dragging the mouse), or multiple ranges of lines (by using the **Ctrl** key) from the following source.

```

    PROCEDURE DIVISION.

        PERFORM A000-OPEN.
        IF WS-FATAL-ERROR
           PERFORM D000-HANDLE-ERRORS
        ELSE
           PERFORM B000-PROCESS
           PERFORM C000-CLOSE
        END-IF.
        MOVE WS-RETURN-CODE          TO RETURN-CODE.
        GOBACK.

    A000-OPEN SECTION.
    MAIN-PARA.
```

WebSphere software

End to End Development and Process: An Improved Process




WebSphere Studio Asset Analyzer V2.0
Enterprise Customer AD artifacts
COBOL
PL/1
Assembler
IMS/DC
CICS
Batch
Source Scanning Tools
Knowledge Store (DB2)
Source Scanning Tools
Enterprise Customer AD artifacts
Java
C++
WebSphere applications
Impact Analysis
Exploration
Connector Information
Netscape / I.E. User Interface
VisualAge Enterprise Suite (JAVA EE,COBOL,PL/1, Generator)
WebSphere Studio Enterprise Developer
WebSphere Studio tooling
Customer business analysts, system analysts, developers, testers, project managers

WebSphere software

Futures

- Complete End-to-End analysis
- Integration with other tools
  - Schedulers
  - Performance analyzer tools
  - etc.

Note: Plans subject to change.

More Information

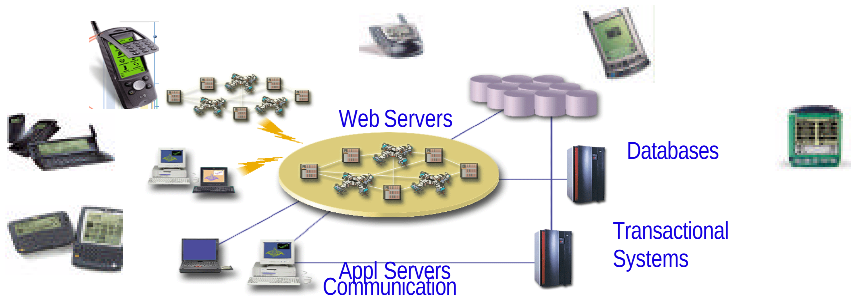
Visit our website:

www.ibm.com/software/ad/wsaa

- Whitepapers
- Fact sheets
- Online demo
- Services information
  - Installation/Configuration Assistance
  - Mentored workshop

WSAA Benefits

- Improved developer productivity
- Reduced time to market
- Higher application quality


Web Servers
Databases
Transactional Systems
Appl Servers
Communication