



IBM Software Group

Developing a Real-life Solution Using IBM Express Runtime Developer

IBM Virtual Innovation Center



IBM Express Runtime

© 2005 IBM Corporation

Notices

This information was developed for products and services offered in the U.S.A.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

ibm.com®

DB2 Universal Database™

DB2®

IBM®

Rational®

WebSphere®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Objectives

The objective of this module is to demonstrate how to:

- Create an Express Runtime solution containing:
 - Trade J2EE Application (Trade)
 - IBM WebSphere Application Server – Express V6.0
 - DB2 Universal Database (UDB) Express V8.2
 - IBM HTTP Server
 - Console for Express Runtime
- Develop the solution wrapper
- Develop the application wrapper for Trade

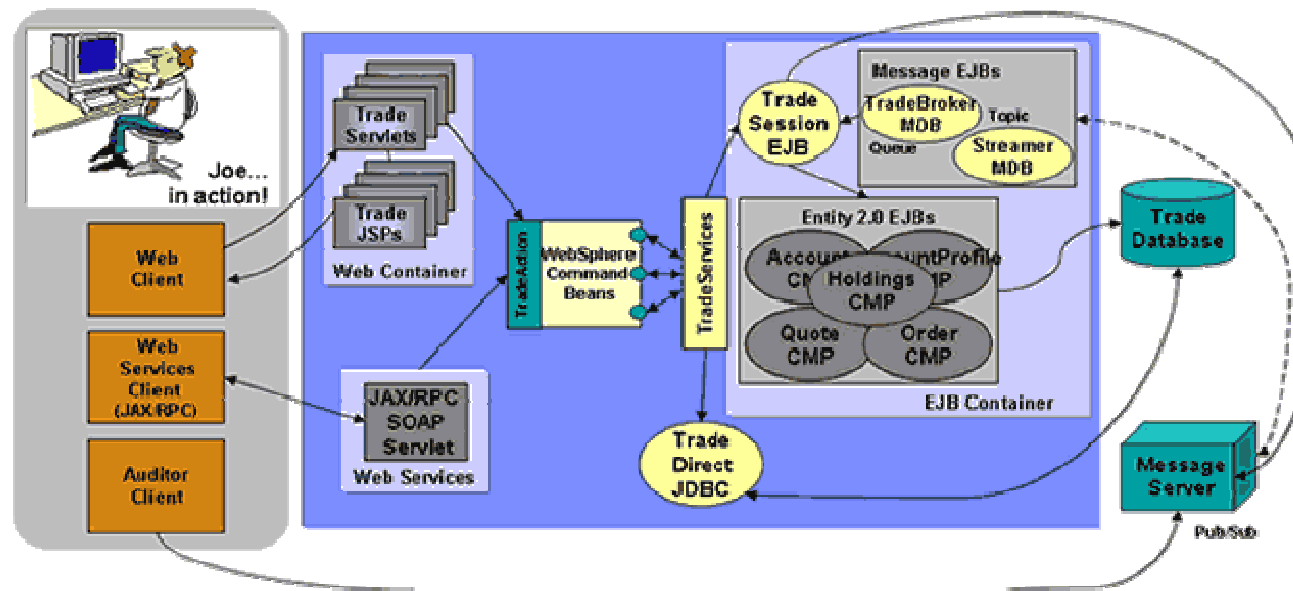
Agenda

- Trade application
- Creating solutions with Express Runtime Developer
 - Application projects
 - Solution projects
 - Creating user programs
- Creating the Trade solution
 - Trade application project
 - Trade solution project
- Testing the solution project

Overview of the Trade application

Trade

- Trade is a complete Java 2 Platform, Enterprise Edition (J2EE) V1.4 application
- It was created by IBM*



* - Trade Version 6 is J2EE 1.4 compliant, but is not yet available externally.

Details of the Trade application

- Trade uses many aspects of J2EE V1.4 and of WebSphere Application Server - Express V6.0.
 - Servlets, JavaServer Pages (JSPs), Enterprise JavaBeans (EJBs), Java Database Connectivity (JDBC), Java Messaging Service (JMS), Web Services
- This presentation uses Trade to illustrate how you can use IBM Express Runtime to create a complete solution that contains a J2EE application and the IBM middleware to support it.

Creating solutions with Express Runtime Developer

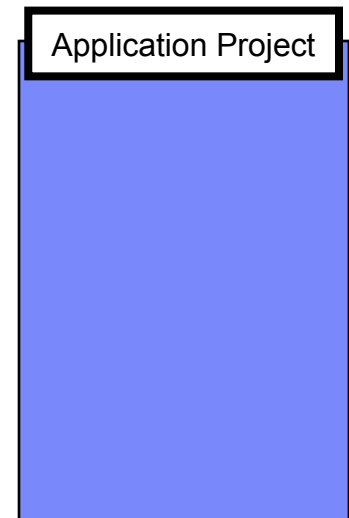
Express Runtime Developer

- An Eclipse-based tool
- Installation provides:
 - Rational Web Developer
 - Alternatively installs into an existing Eclipse 3.0 environment
 - Express Runtime development plug-ins
- Express Runtime Developer has two types of projects:
 - Application project
 - Solution project

Application projects

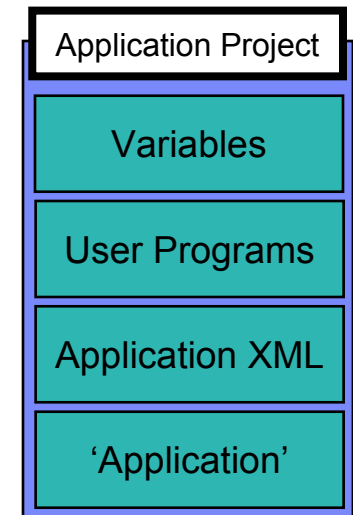
Application projects

- Responsible for deploying an application
 - Installation
 - Configuration
 - Verification
- Applications can be:
 - Middleware components
 - WebSphere Application Server - Express, DB2 UDB Server Express, IBM HTTP Server
 - Real applications
 - Trade, your application



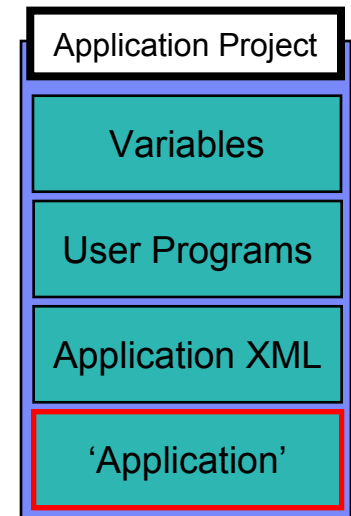
Application project: Contents

- The application itself
- Application XML
 - Defines the contents and behaviour of the project
- User programs
 - Check prerequisites
 - Silently invoke installers and scripts
 - Verify the success
- Variables
 - Values to be provided at deployment time



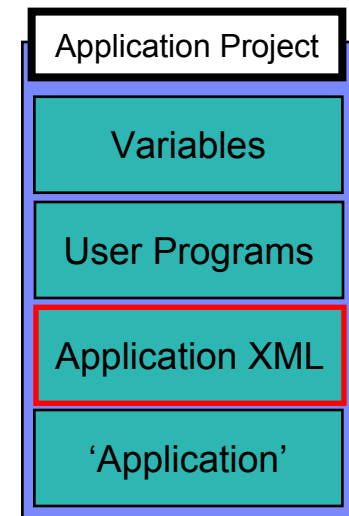
The 'application'

- All files related to the application itself
- For example:
 - Enterprise archive resource (EAR) file
 - Static content such as images



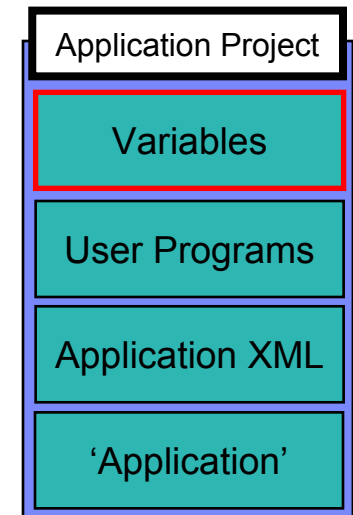
Application XML

- Defines all of the pieces of the Application project
 - Variables
 - Including validation checking
 - User programs
 - Names
- Operating system support
- Language support
- Named application.xml by default
- Edit using wizards (not raw Extensible Markup Language (XML))



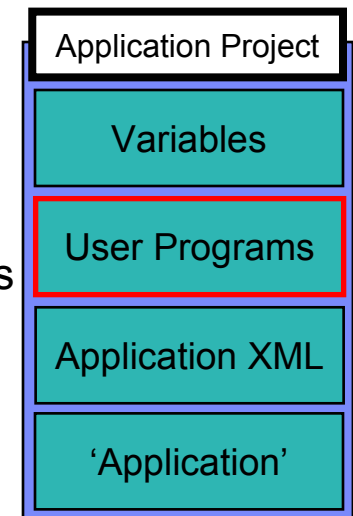
Variables

- Actually part of the application XML
- Define values that the Deployment Wizard asks for when the solution is deployed
- Can be associated with response files
 - When the deployer enters a value, the response file is automatically updated with this value.
- Can have validation defined
 - The Deployment Wizard checks that the deployer provides valid values.



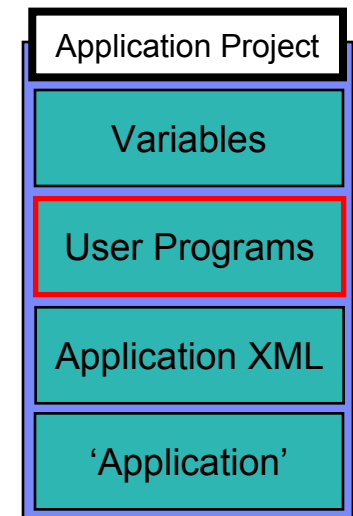
User programs: Part one

- These are the engine of your application project
- Responsible for deploying the application
- Often they invoke such installation procedures as silent installers or scripts:
 - InstallShield Multiplatform (ISMP)
 - Installs WebSphere Application Server - Express and IBM HTTP Server
 - InstallShield Professional (ISS)
 - Configuration, Installation, and Distribution (CID)
 - Installs DB2 UDB Express
 - Configuration scripts
 - wsadmin, ANT (installs WebSphere Application Server - Express applications and configures WebSphere Application Server - Express)



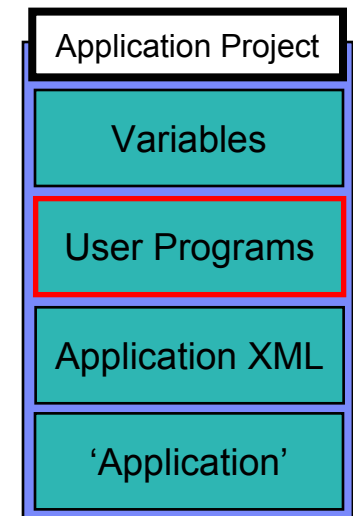
User programs: Part two

- User programs can also be used for:
 - Prerequisite checking
 - Variable manipulation
 - Copying files
 - Verifying installation success
 - Anything you like



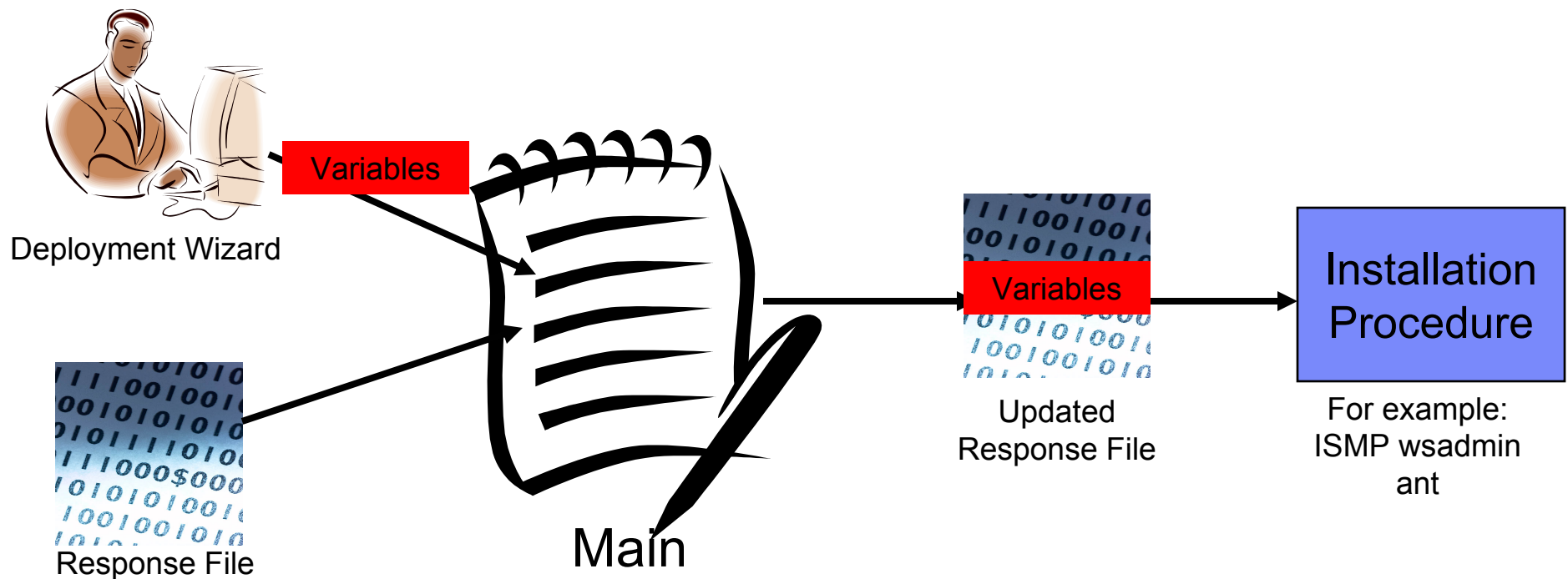
User programs: Part three

- Four types of user program can be defined
 - Predeployment Checkers (PDC). The most important benefit of the PDC is to determine if the conditions are not right for deployment. Then the application image is not transferred over the network to the target machine.
 - Entry
 - Main
 - Exit
- They execute in the order previously shown
 - Main is compulsory
 - PDC, Entry, and Exit are optional



Typical main user program

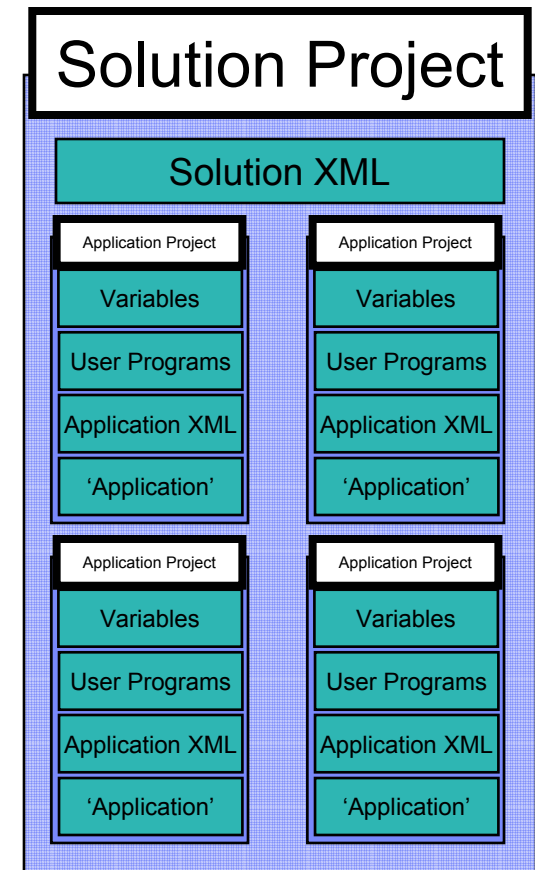
- Takes input from an initial response file
- Reads additional variables provided by the Deployment Wizard
- Updates the response file with variables
- Invokes the installation procedure



Solution projects

Solution projects

- Responsible for defining a solution
 - Consist of multiple applications
 - Each defined in an application project
- Share variables between applications
 - Deployer only enters shared values once
- Define Install Tasks
- Define application groupings
- Validate variables
 - Check deployer values are valid for the solution



Writing user programs

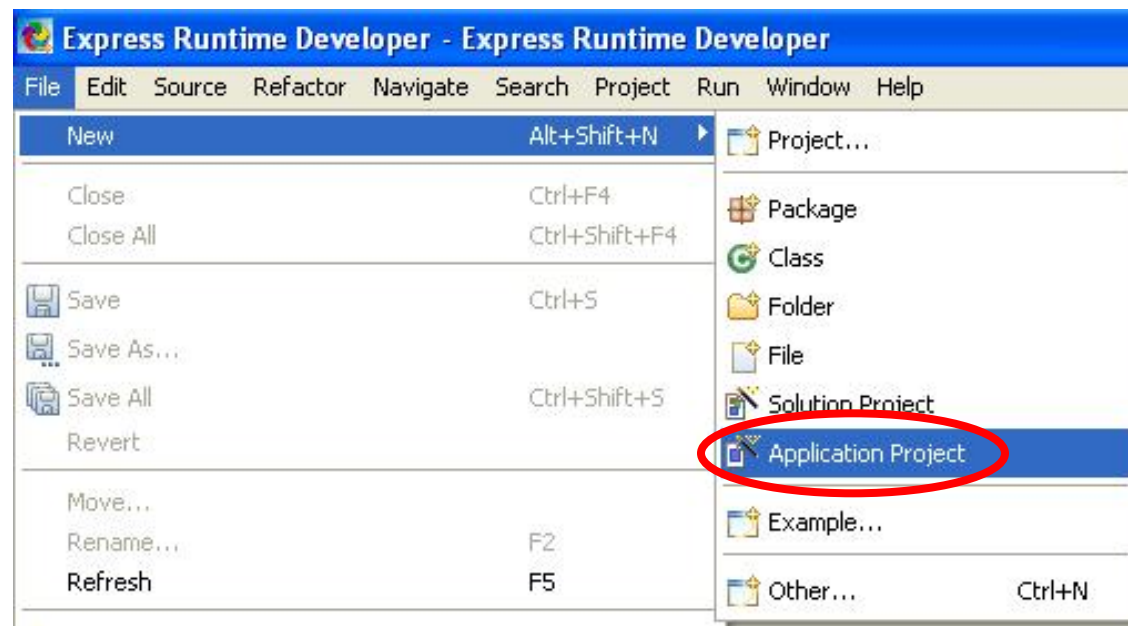
Writing user programs

- Typically written in Java
- Java user programs can use Express Runtime support application programming interfaces (APIs)
 - Simplify user program development
- JavaDoc for support API classes is available in <ER_install>\Runtime21\SolutionEnabler\Support_Framework\index.html
- Sample Application Project contains examples
 - (See Express Runtime Developer project [IRU2_1SampleWin](#))

Creating the Trade solution: The application project

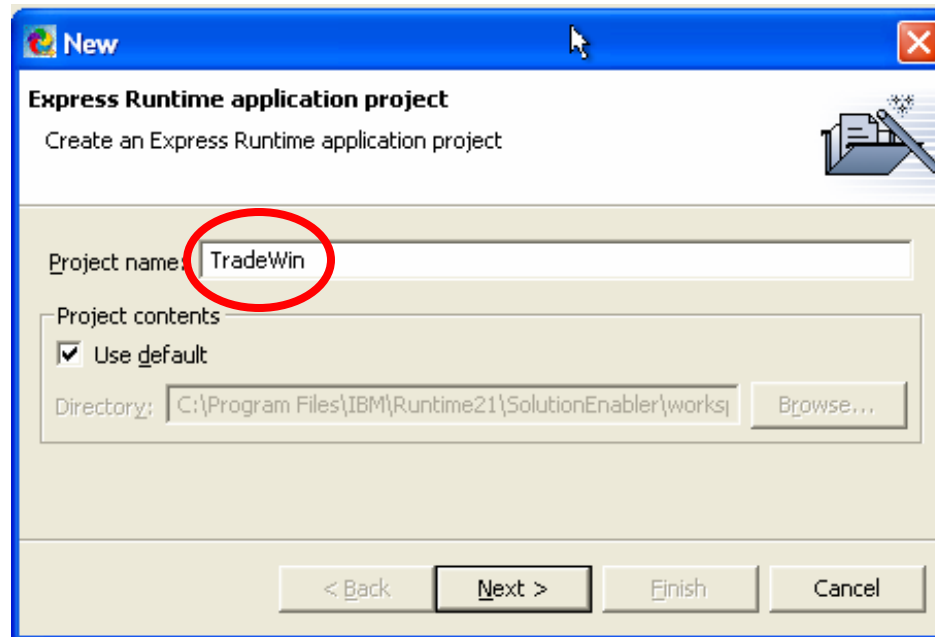
Create the Trade application project: Wizard part 1

- Create a new Application Project. Select **File -> New -> Application Project** (we do it from within Express Runtime Developer perspective)



Create the Trade application project: Wizard part 2

- Enter the project name.



Create the Trade application project: Wizard part 3

- Enter the details of the project.

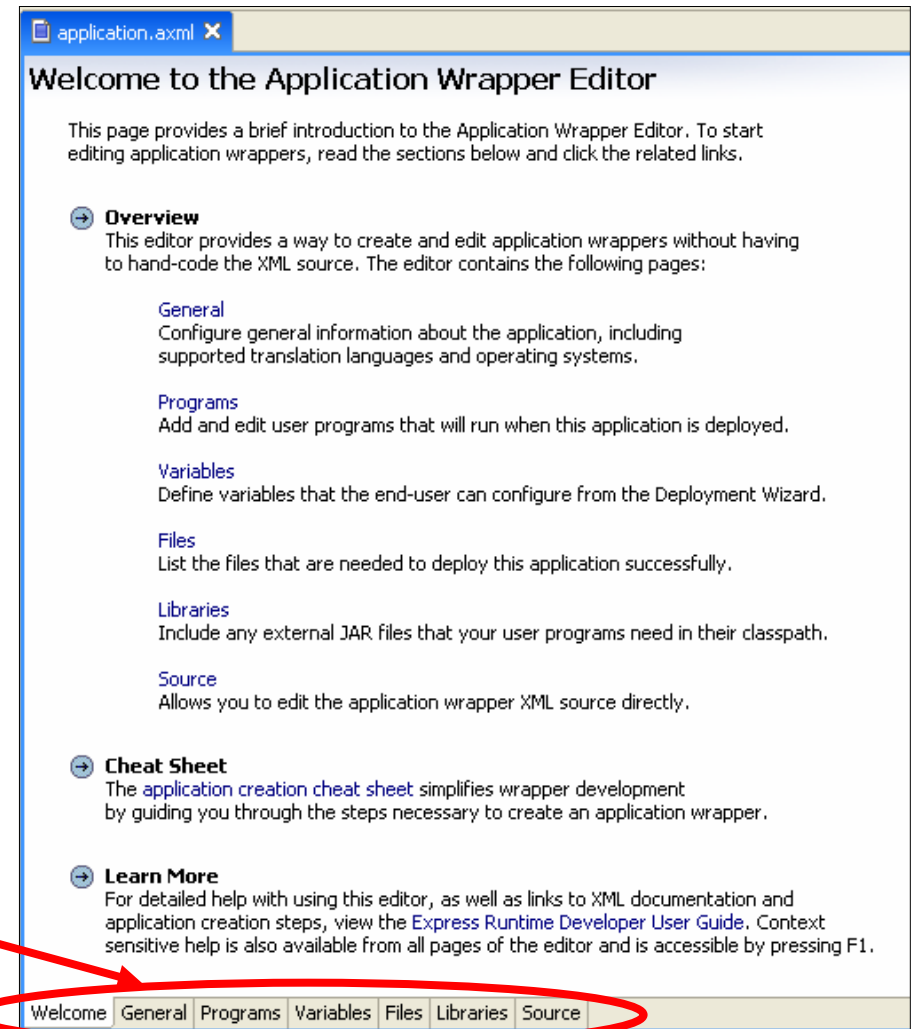
The screenshot shows a 'New' wizard dialog box titled 'Express Runtime application project'. The dialog contains the following fields and options:

- Application ID:** TradeWin
- Version:** 1.0
- Installation time (minutes):** 20
- Operating system:** Windows (dropdown menu)
- Default language:** English (dropdown menu)
- Wrapper file name:** application.xml

At the bottom of the dialog, there are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. The 'Finish' button is highlighted with a mouse cursor.

Trade application XML Editor

- The Editor allows you to edit the application wrapper using wizards and graphical input.
 - You do not have to edit the raw XML
- The Welcome page has useful links.
- The Cheat Sheet guides you through creating an application.
- Use the tabs to configure different pieces of the wrapper



Trade application XML Editor: General tab

General Application Information

Basic Application Configuration

*ID: TradeWin ([Click to change](#))

*Application name: TradeWin

*Version: 1.0

Installation time (minutes): 20

Provider name: IBM

License text:

Advanced Application Configuration

Deployment package protected:

Deployment package name:

Configuration instructions:

Supported Translation Languages

Select the translation languages for this wrapper. For each selected language, you must provide an XML file containing translated strings.

*Default: English

*Languages:

- Chinese
- Chinese (Taiwan)
- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazil)

Select All
Deselect All

Supported Operating Systems

Select the operating systems supported by this application.

- Linux
- Linux on POWER
- OS/400 (i5/OS)
- Windows

Select All
Deselect All

Welcome **General** Programs Variables Files Libraries Source

Trade application XML Editor: Programs tab

- This is where user programs are defined.
- A main user program is required, but no program is associated by default.
- Define the user programs for Trade from this tab.

The screenshot shows the 'User Programs Information' section of the XML Editor. It includes a table for 'User Programs' with one entry, 'Main Program'. Below this are several configuration sections: 'Basic Program Configuration' (Program type, Program), 'Custom Program Options' (System command), 'Java Program Options' (Additional classpath), 'Advanced Program Configuration' (Timeout, Wait for completion, Program reboots, Force reboot, Environment variables), 'Program Success Type' (Success type, Search strings), and 'Program Arguments' (Response file, Log file name, Arguments). The 'Programs' tab in the bottom navigation bar is circled in red.

User programs for Trade

- Trade uses two user programs
 - Predeployment Checker
 - Main: Installation program
- PDC performs predeployment checks
 - Is WebSphere Application Server - Express is installed, is DB2 installed? Is the IBM HTTP Server installed?
 - Is Trade already installed?
- Main deploys the Trade application
 - Create DB2 table
 - Invoke wsadmin scripts to configure WebSphere Application Server - Express
 - Configure resources and install the EAR file

Define the Trade main user program

5. Now add a PDC program

1. Select the Program type and enter the Program Class name

User Programs
Add, remove or edit the user programs defined in the application. The main program is required.

Main Program	Add...	Remove
--------------	--------	--------

Basic Program Configuration

*Program type:
 *Program:

Custom Program Options

System command:

Java Program Options

Additional classpath:

Advanced Program Configuration

Timeout (minutes):
 Wait for complete:
 Program reboot:
 Force reboot:
 Environment variables:

Program Success Type
Specify how to determine if the program ran successfully.

Success type:
 Search strings:

Program Arguments

Response file:
 Log file name:
 Arguments:

2. Enter the Response file name. In this example, it's part of the project, no need for fully-qualified path.

3. Enter the Log file name

4. Add the Response file as an argument to the program

Define the Trade PDC

Add User Program
Add a user program to the application
The predeployment checker is run before the deployment package is transferred to the inst

Add program:
 Predeployment Checker
 Entry Program
 Exit Program

Add User Program
Specify the user program type
Run a Java program from a .class file.

Program type:
 Custom program
 InstallShield executable
 Java program

Main class: `com.trade.TradeWinPDC`
Enter the PDC Class name

Predeployment Checker
Main Program

Basic Program Configuration
 *Program type: Java program
 *Program: com.trade.TradeWinPDC

Program Success Type
 Specify how to determine if the program
 Success type: Check return code
 Search strings:

Custom Program Options
 System command:

Java Program Options
 Additional cla

Advanced Pro
 Timeout (minutes):
 Wait for completion:
 Program reboot
 Force reboot:

Program Arguments
 Response file: Trade.prop
 Log file name: TradeWinPDC.log
 Arguments: <response file>

Enter a Response file name

Enter a Log file name

Add a Response file as a program argument

Create the PDC userProgram class

The image shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure. The package `src/TradeWin/userPrograms` is selected, and the context menu is open with the `Class` option highlighted. A red arrow points from this menu item to the `New Java Class` dialog box on the right.

The `New Java Class` dialog box is configured as follows:

- Source Folder:** `TradeWin/src/TradeWin/userPrograms`
- Package:** `com.trade`
- Enclosing type:** (unchecked)
- Name:** `TradeWinPDC`
- Modifiers:** `public` (selected), `default`, `private`, `protected`, `abstract`, `final`, `static` (unchecked)
- Superclass:** `com.ibm.jsdt.support.SupportWindowsBase`
- Interfaces:** (empty)
- Which method stubs would you like to create?**
 - `public static void main(String[] args);`
 - Constructors from superclass
 - Inherited abstract methods

Buttons for `Browse...`, `Finish`, and `Cancel` are visible.

Trade PDC (com.trade.TradeWinPDC)

- Create or paste code that:
 - Performs simple prerequisite checks
 - Determine if WebSphere Application Server - Express is installed
 - Determine if DB2 UDB Express is installed
 - Verify that Trade is not installed already
 - Returns zero return code if successful

Create the Main userProgram class

The image shows the Eclipse IDE interface. On the left, the Project Explorer displays the 'TradeWin' project structure. The 'src/TradeWin/userPrograms' folder is selected and circled in red. A context menu is open over this folder, with the 'Class' option also circled in red. A red arrow points from this menu to the 'New Java Class' dialog box on the right.

The 'New Java Class' dialog box is configured as follows:

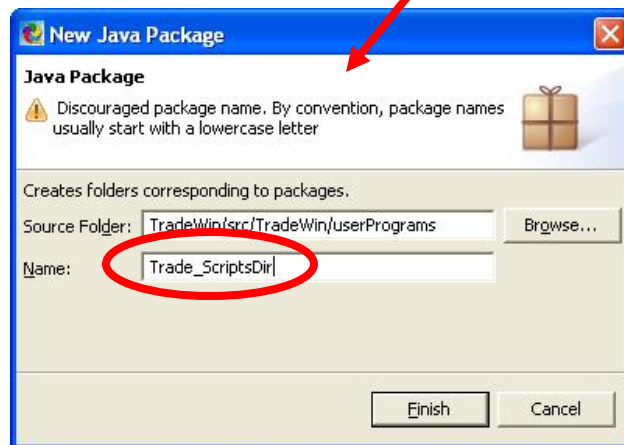
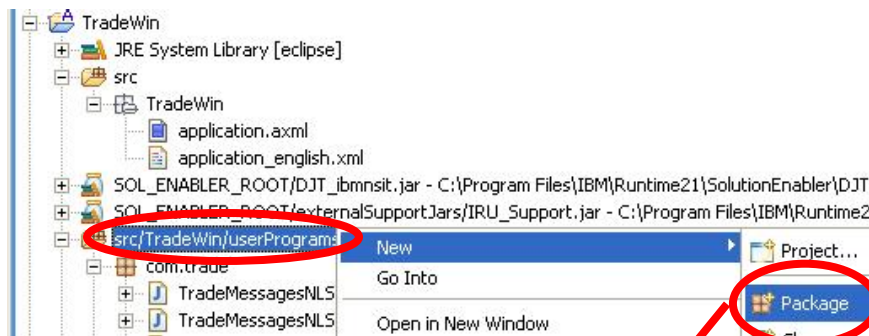
- Source Folder:** TradeWin/src/TradeWin/userPrograms
- Package:** com.trade
- Name:** TradeWinMain
- Modifiers:** public (selected), default, private, protected
- Superclass:** com.ibm.jsdt.support.SupportWindowsBase
- Which method stubs would you like to create?:**
 - public static void main(String[] args)
 - Constructors from superclass
 - Inherited abstract methods

Trade Main (com.trade.TradeWinMain)

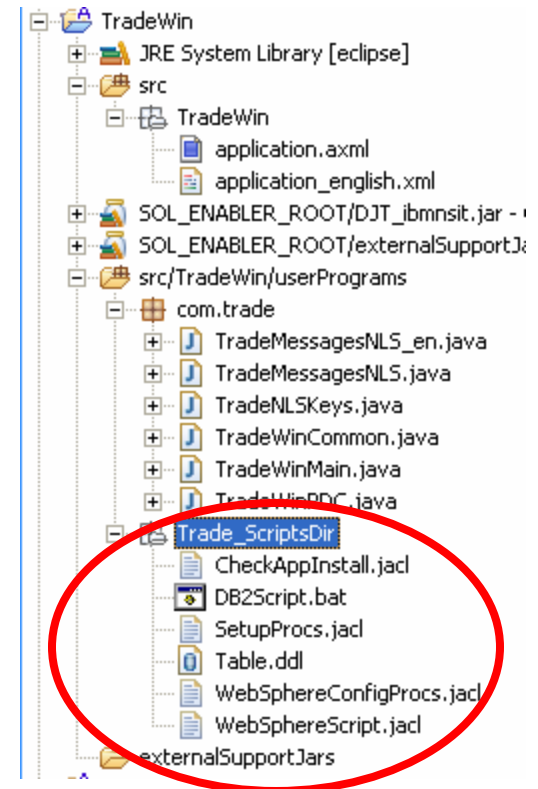
- Create or paste code that:
 - Takes values from the initial response file
 - Takes values from the Deployment Wizard
 - Determines the installation location of WebSphere Application Server - Express and DB2 UDB Express
 - Copies the **Trade.ear** into the **installableApps** directory
 - Invokes the DB2 script
 - Uses the **db2cmd** command with script to create Trade database
 - Invokes the WebSphere Application Server - Express script
 - Configures WebSphere Application Server - Express and deploys the Trade application
 - Uses the wsadmin script interface to WebSphere Application Server - Express with JACL script

Add scripts that are used by userPrograms

- Create a Scripts directory inside userPrograms



- Copy the previously developed scripts for your application into this directory



Variables tab

- Add variables here
- Specify the validation rules for the variable's value

Application Variables

String Variable: DB2UserId

Add...
Remove
Up
Down

Basic Variable Configuration

Specify label and help text for the selected variable.

*Name: DB2UserId
*Label text: DB2 Administrator UserID
Help text:
Type: Typical variable

Variable Associations Configuration

Add, remove or edit the associations for the selected variable.

Property Association: DB2UserId

Variable Validation Configuration

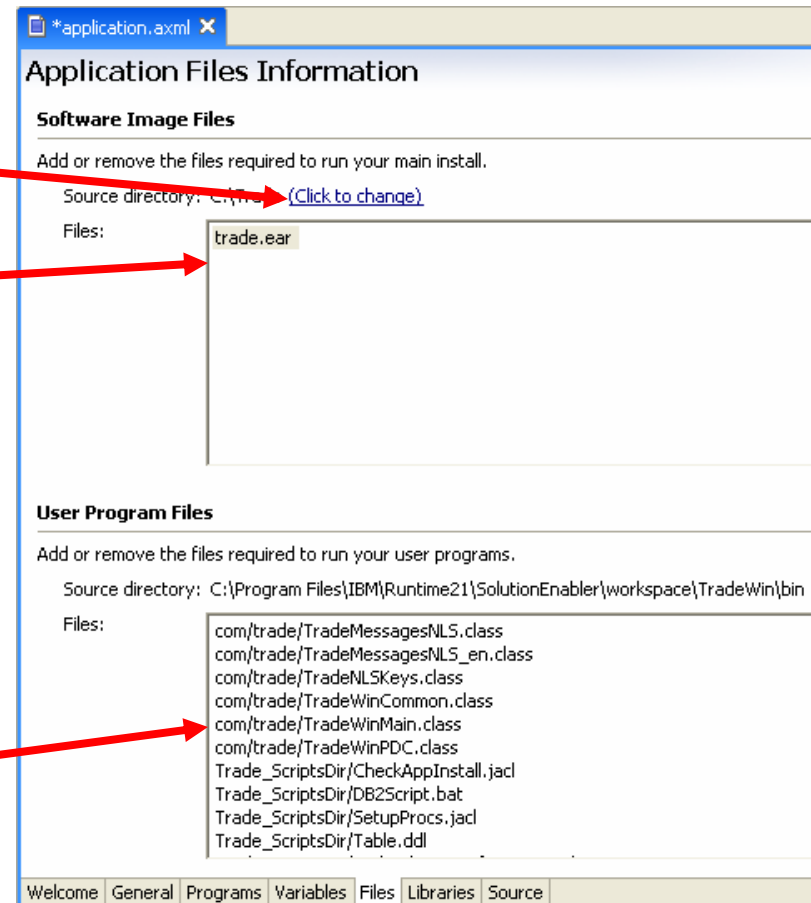
Specify validation requirements for the selected variable value.

Default value:
Required:
Make uppercase:
Make lowercase:
Minimum length: 2
Maximum length: 30
Validation rules:
Valid characters: "@#\$_abcde
Invalid value: "ADMINS"
Invalid value: "GUESTS"
Invalid value: "USERS"
Invalid value: "PUBLIC"
Invalid value: "LOCAL"

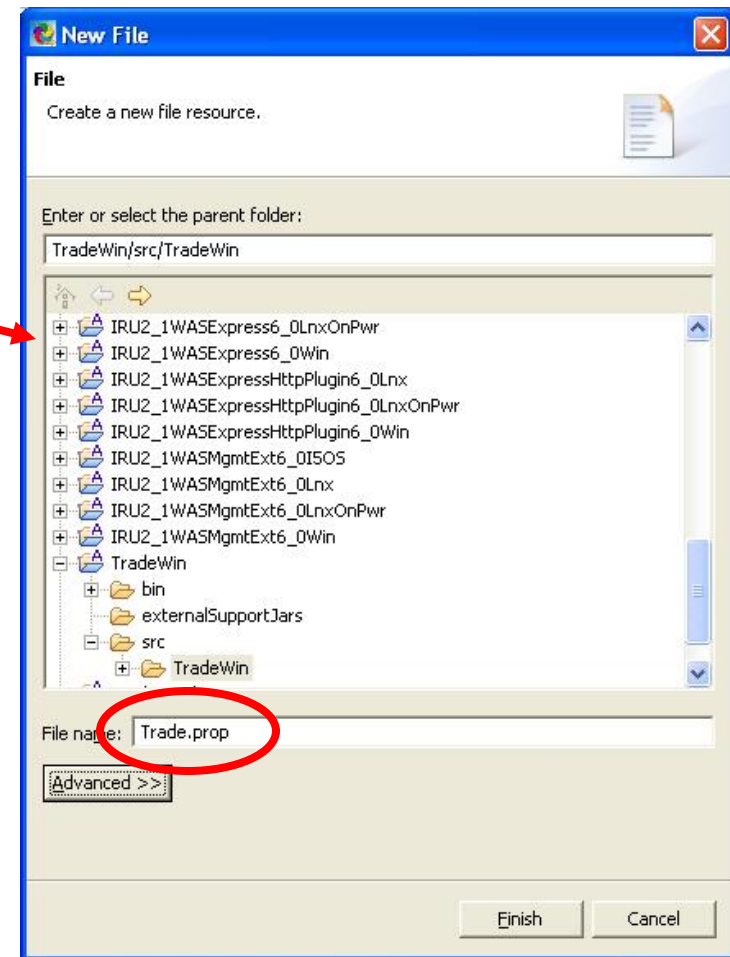
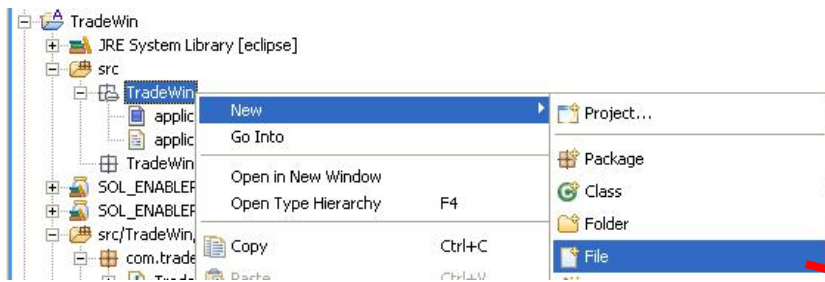
Validation rules

Files tab

- Copy the `trade.ear` file into a directory, for example `c:\Trade`.
- Set **Source directory** by clicking the link
- Add the application file (EAR in our example) to the Software Image files.
- Add all compiled user programs and any script files required by them.



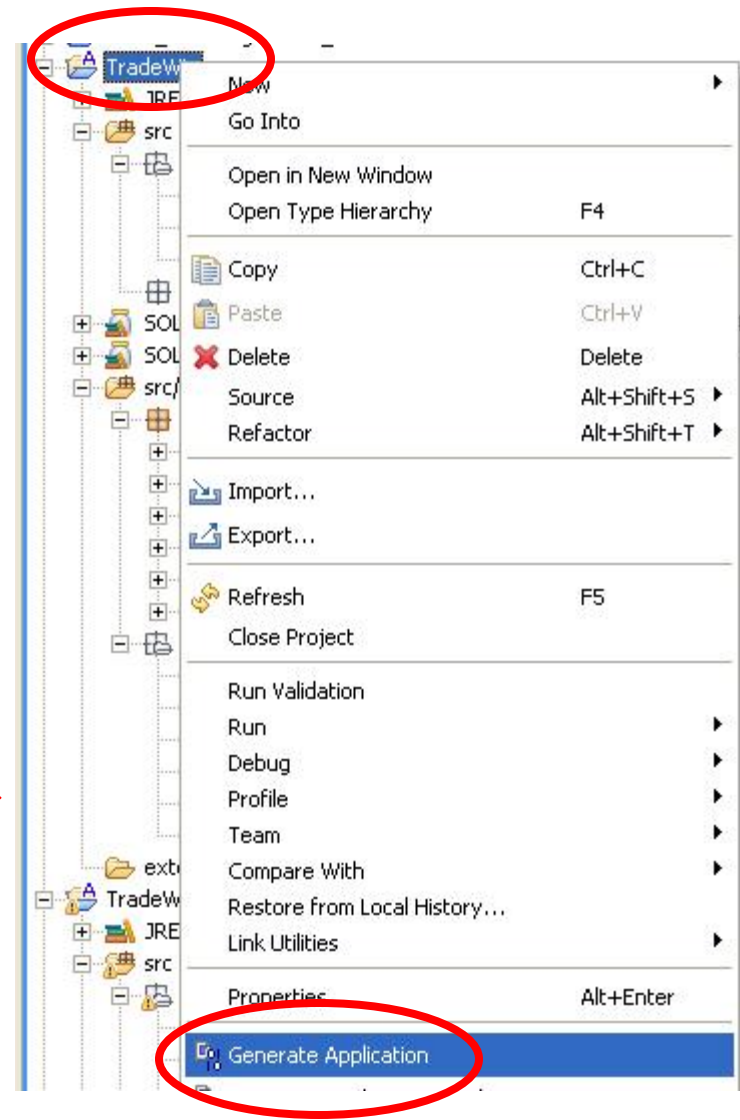
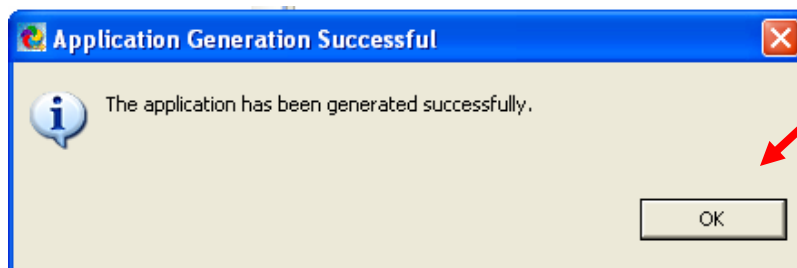
Create the response file: Trade.prop



- Add the appropriate response values to [Trade.prop](#).

Generate the application

- Save all files first.
- Select **Generate Application** from the context menu.



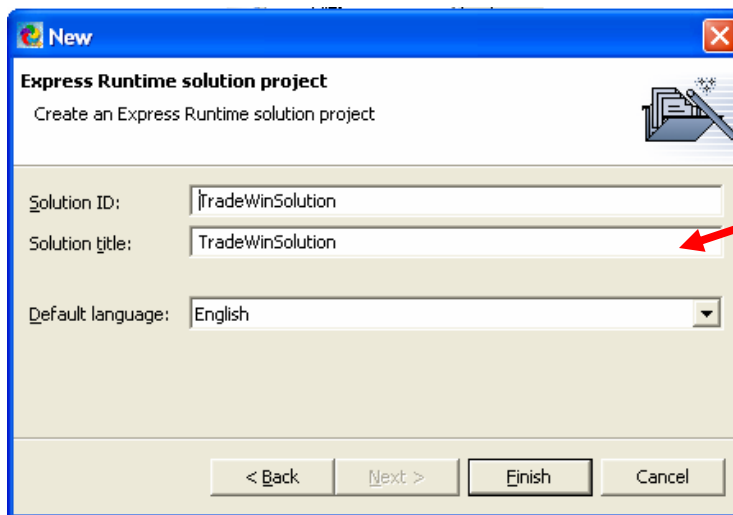
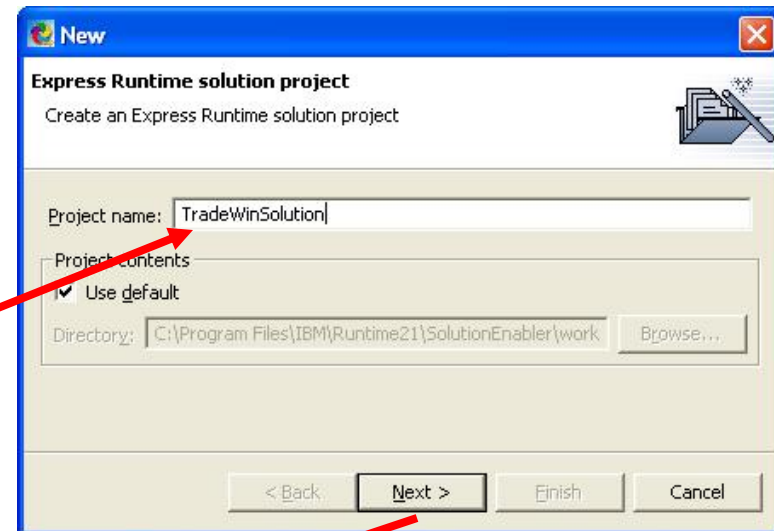
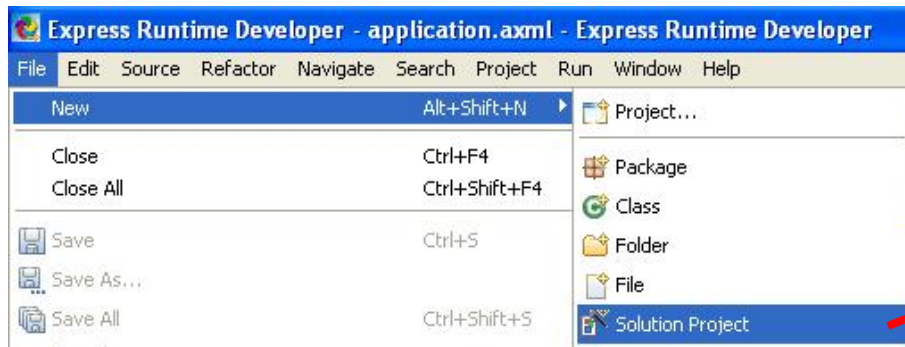
Generate the deployment packages

- Select **Generate Deployment Packages** from the application project context menu.



Creating the Trade solution project

Create the Trade solution project: Wizard



Trade Solution Wrapper XML Editor

- **Access the configuration details by switching between tabs**

solution.xml

Welcome to the Solution Wrapper Editor

This page provides a brief introduction to the Solution Wrapper Editor. To start editing solution wrappers, read the sections below and click the related links.

- ➔ **Overview**
This editor provides a way to create and edit solution wrappers without having to hand-code the XML source. The editor contains the following pages:
 - [General](#)
Configure general information about the solution, including supported translation languages.
 - [Tasks](#)
Add and edit tasks and applications that will run when this solution is deployed.
 - [Validation](#)
Allows you to define validation information for [shared variables](#), if your solution uses them.
 - [Source](#)
Allows you to edit the application wrapper XML source directly.
- ➔ **Cheat Sheet**
The [solution creation cheat sheet](#) simplifies wrapper development by guiding you through the steps necessary to create a solution wrapper.
- ➔ **Learn More**
For detailed help with using this editor, as well as links to XML documentation and solution creation steps, view the [Express Runtime Developer User Guide](#). Context sensitive help is also available from all pages of the editor and is accessible by pressing F1.

Welcome | General | Tasks | Validation | Source

General tab

- Provide general information about this solution, such as the name, images, and so on.

The screenshot shows the 'General Solution Information' configuration window. Red circles highlight the following fields:

- *ID:** TradeWinSolution
- *Title:** TradeWinSolution
- Version:** 6.0
- About screen text:** `<![CDATA[<html><p align="left"><div style="margin-left:25px">
Trade Solution
</style></align></div></td></tr></table>]]>`
- Welcome screen title:** Trade V6 Solution
- Welcome screen text:** The solution to deploy all middleware, the Trade application, and Console for Express Runtime.
- Task group selection prompt:** Select which tasks you would like to deploy
- Deployment package path:** (empty)

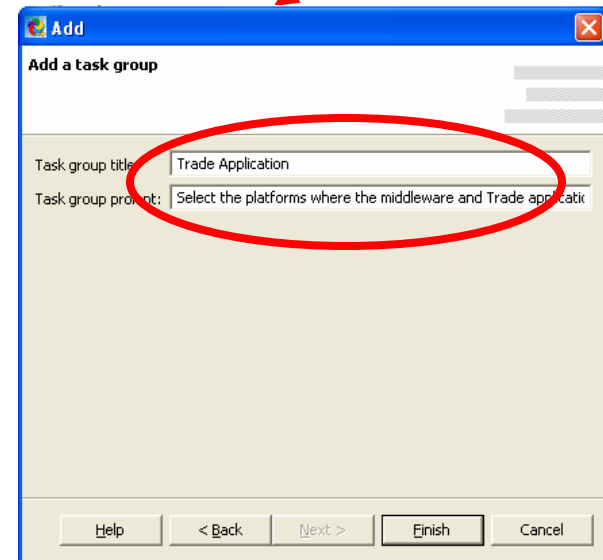
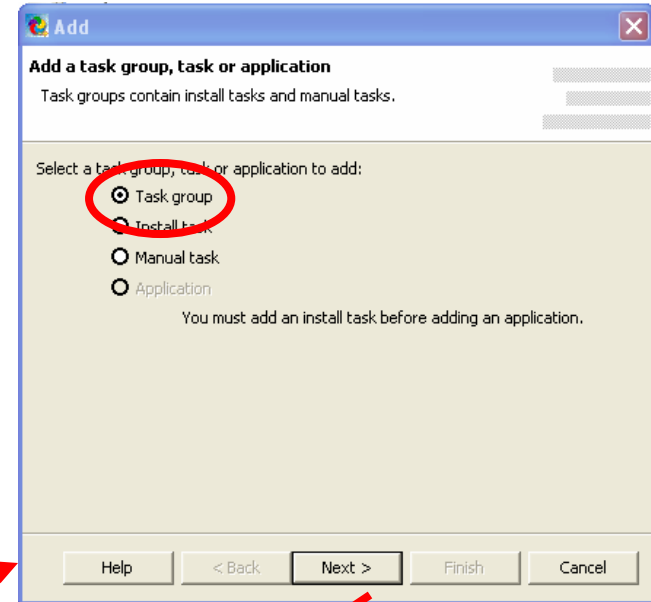
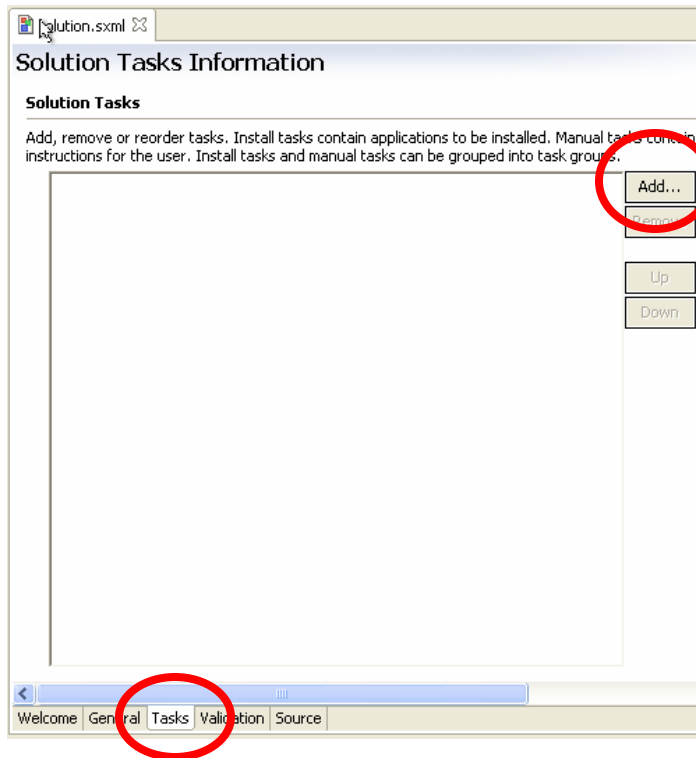
The 'Supported Translation Languages' section on the right shows the following configuration:

- *Default:** English
- *Languages:** English, Chinese, Chinese (Taiwan), French, German, Italian, Japanese, Korean, Portuguese (Brazil), Spanish

The bottom navigation bar includes: Welcome | **General** | Tasks | Validation | Source

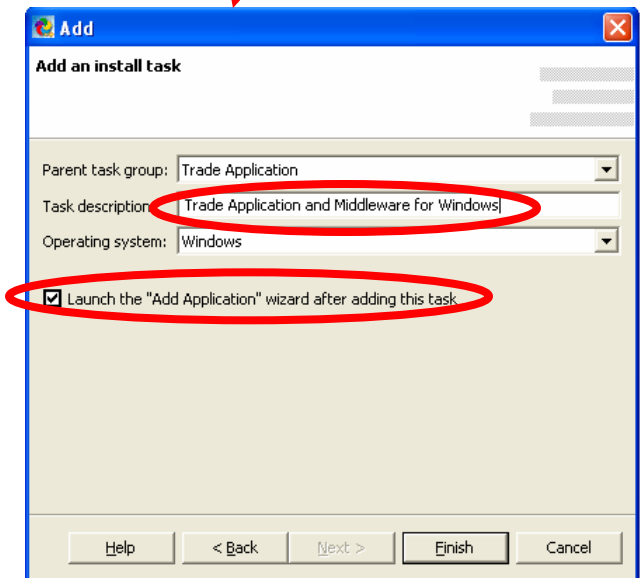
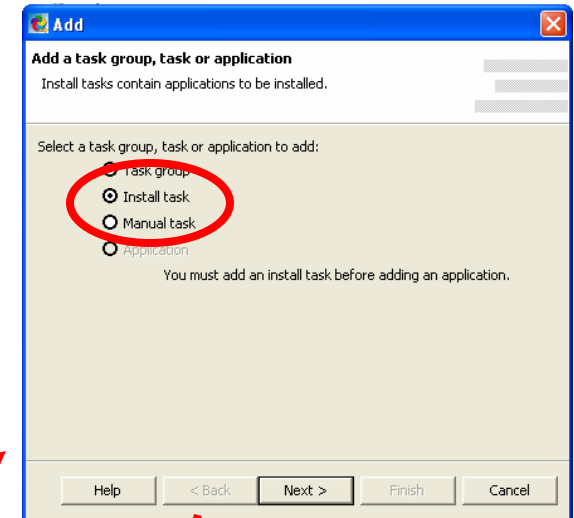
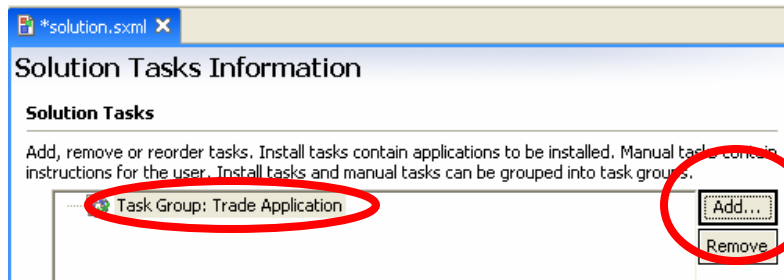
Tasks tab

- Create the **Task group** to group your applications in the solution.



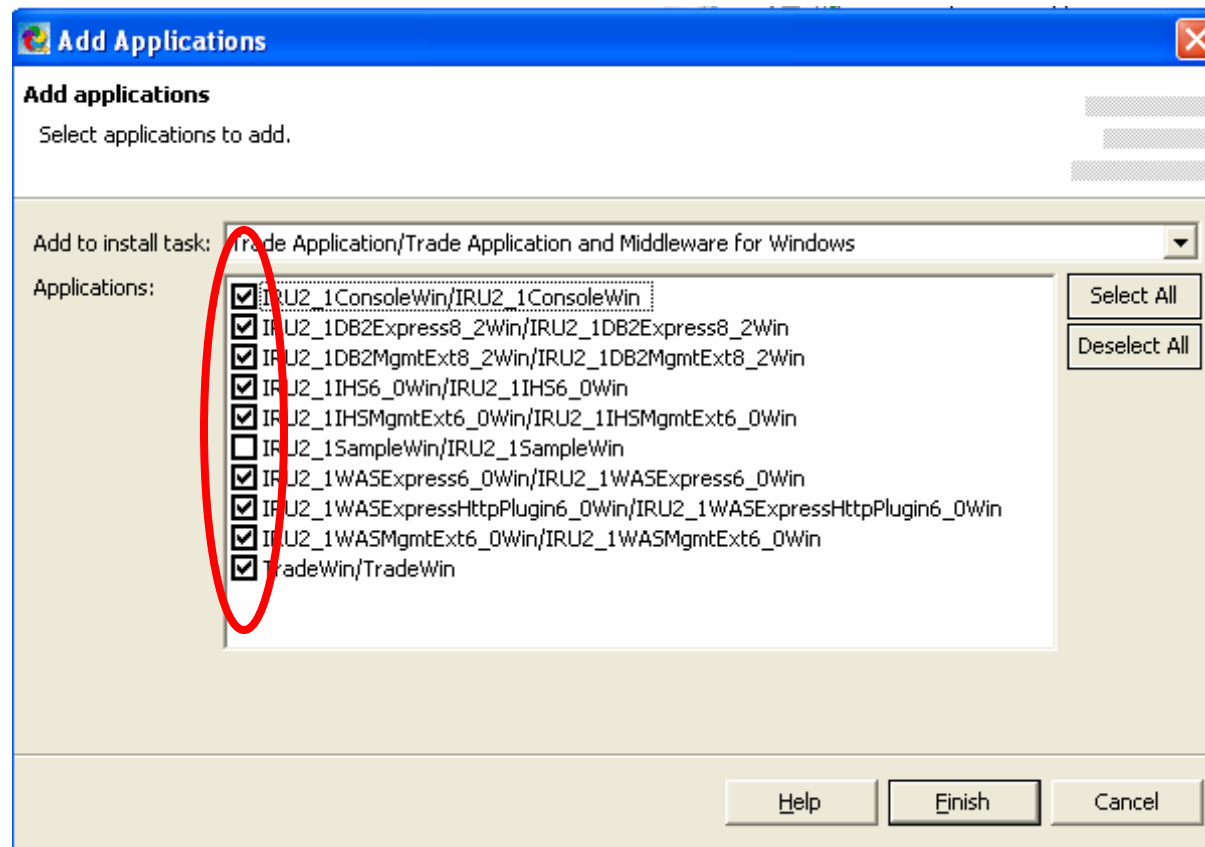
Tasks tab

- Create **Install task** under **Task group** to define the installation process for your applications.
- You may specify several installation tasks.



Add applications to the Install Task

- Add all middleware for Windows and TradeWin applications to your installation task.

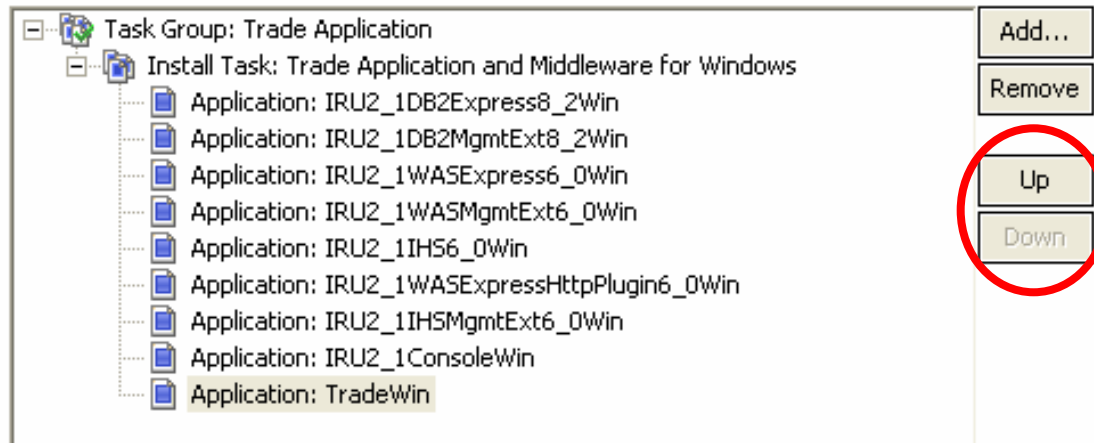


Order applications correctly

- **Make sure you place applications in the correct order. For example, you can't install TradeWin application without installing WebSphere Application Server - Express first.**

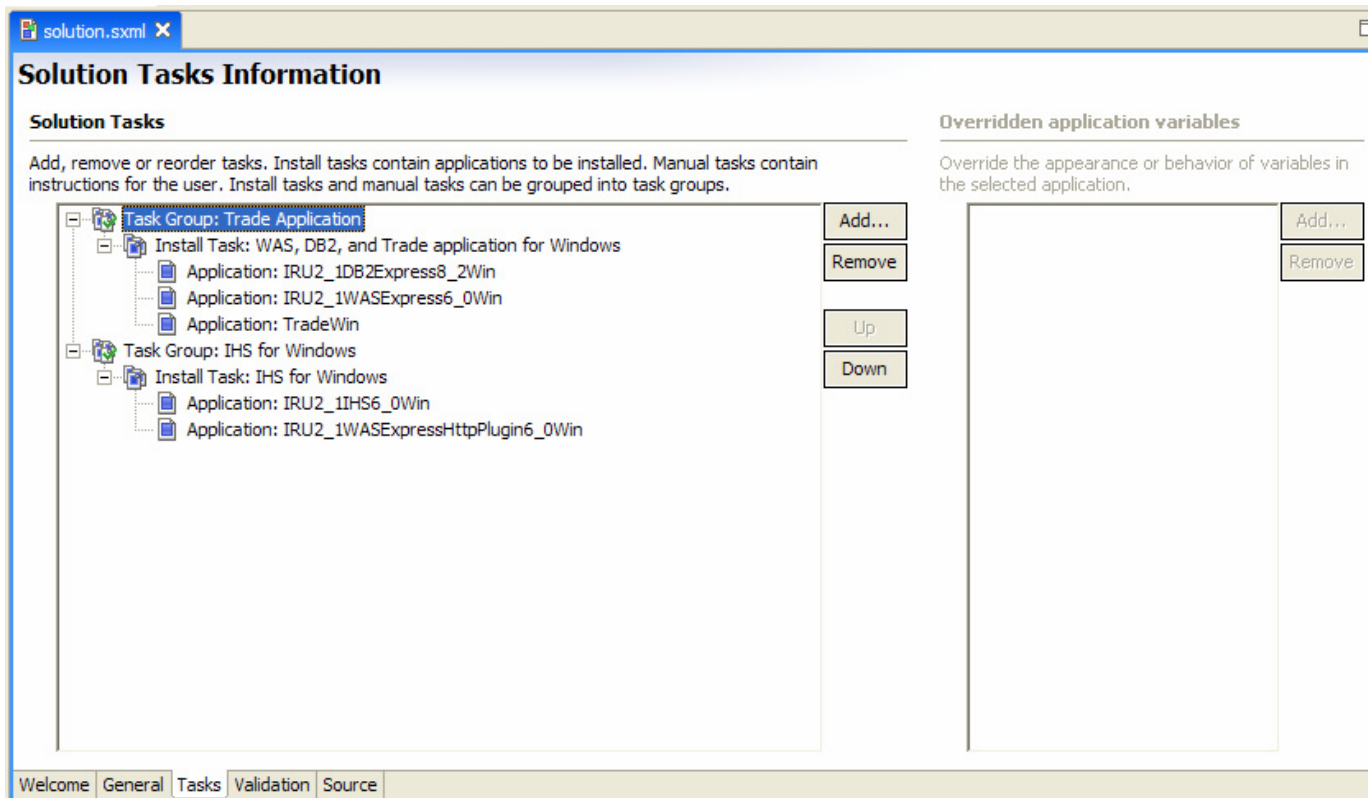
Solution Tasks

Add, remove or reorder tasks. Install tasks contain applications to be installed. Manual tasks contain instructions for the user. Install tasks and manual tasks can be grouped into task groups.



Add applications to multiple Install Task: Optional step

- If you want to support deployment to the multiple systems, you need to separate all applications into several installation tasks. In this example we can deploy WebSphere, DB2, and TradeWin applications on one system and IBM HTTP Server on another system.



Add variable sharing

- Sharing a variable enables 2 or more fields to share the same value, usually across other application wrappers. Shared variables have their own validation rules.

The screenshot illustrates the process of adding variable sharing in the IBM software configuration interface. It is divided into three main sections:

- Installation Tasks:** A tree view on the left shows a task group named "Trade Application" containing an "Install Task: Trade Application and Middleware for Windows". Under this task, several application wrappers are listed, including "Application: IRU2_1DB2Express8_2Win", which is circled in red.
- Overridden application variables:** A central panel with an "Add..." button (circled in red) allows users to select variables to override. A red arrow points from this button to the "Override Application Variables" dialog box.
- Override Application Variables dialog:** A modal dialog box titled "Override Application Variables" with the instruction "Select application variables to override." It contains a list of variables with checkboxes. The "username" and "password" variables are checked and circled in red. Other variables listed include destination, BR, CN, DE, FR, ES, IT, JP, KR, TW, DK, FI, NO, PL, RU, and SE. Buttons for "Select All", "Deselect All", "Help", "< Back", "Next >", "Finish", and "Cancel" are also visible.

Example: Variable overrides

- You create an alias for a shared *value*. Specify alias in the **Shared as** box.

The screenshot shows the 'Override Application Variables' dialog for 'username'. The 'Behavior' dropdown is set to 'Share the value of username with other variables' and the 'Shared as' dropdown is set to 'DB2AdminUserIDWin'. Both dropdowns are circled in red.

Override username
Override the appearance or behavior of username.

Appearance: Editable (default)

Behavior: Share the value of username with other variables

Default value:

Shared as: DB2AdminUserIDWin

Help < Back Next > Finish Cancel

The screenshot shows the 'Override Application Variables' dialog for 'password'. The 'Behavior' dropdown is set to 'Share the value of password with other variables' and the 'Shared as' dropdown is set to 'DB2AdminPasswordWin'. Both dropdowns are circled in red.

Override password
Override the appearance or behavior of password.

Appearance: Editable (default)

Behavior: Share the value of password with other variables

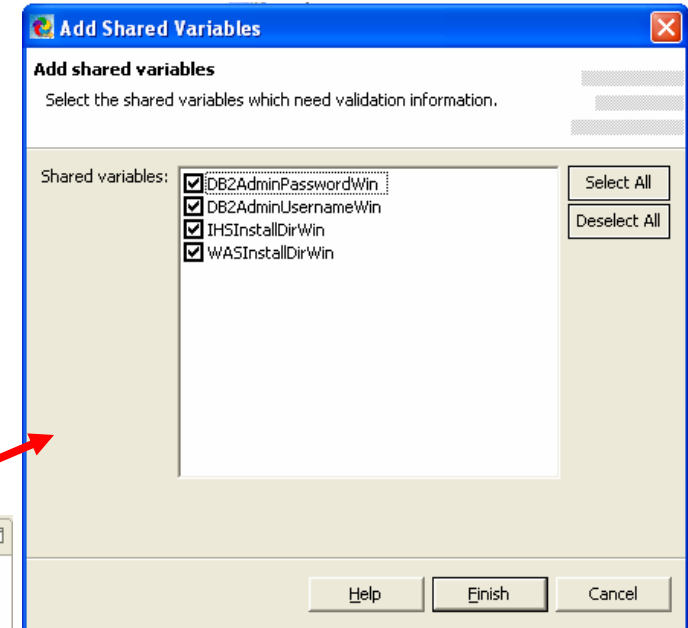
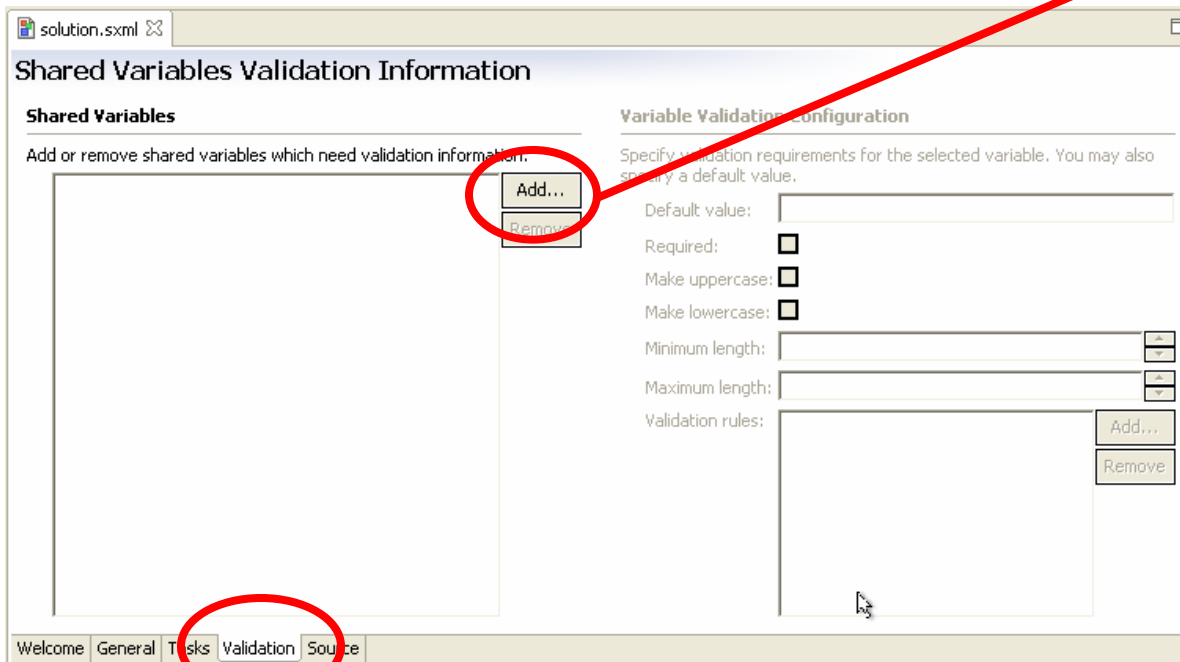
Default value:

Shared as: DB2AdminPasswordWin

Help < Back Next > Finish Cancel

Validation tab

- Add all variables that have to be validated by the Deployment Wizard.



Add appropriate validation rules

Shared Variables Validation Information

Shared Variables
Add or remove shared variables which need validation information.

- DB2AdminPasswordWin
- DB2AdminUsernameWin**
- IBMInstallDirWin
- WASInstallDirWin

Variable Validation Configuration
Specify validation requirements for the selected variable. You may also specify a default value.

Default value: **db2admin**

Required:

Make uppercase:

Make lowercase:

Minimum length: 1

Maximum length: 30

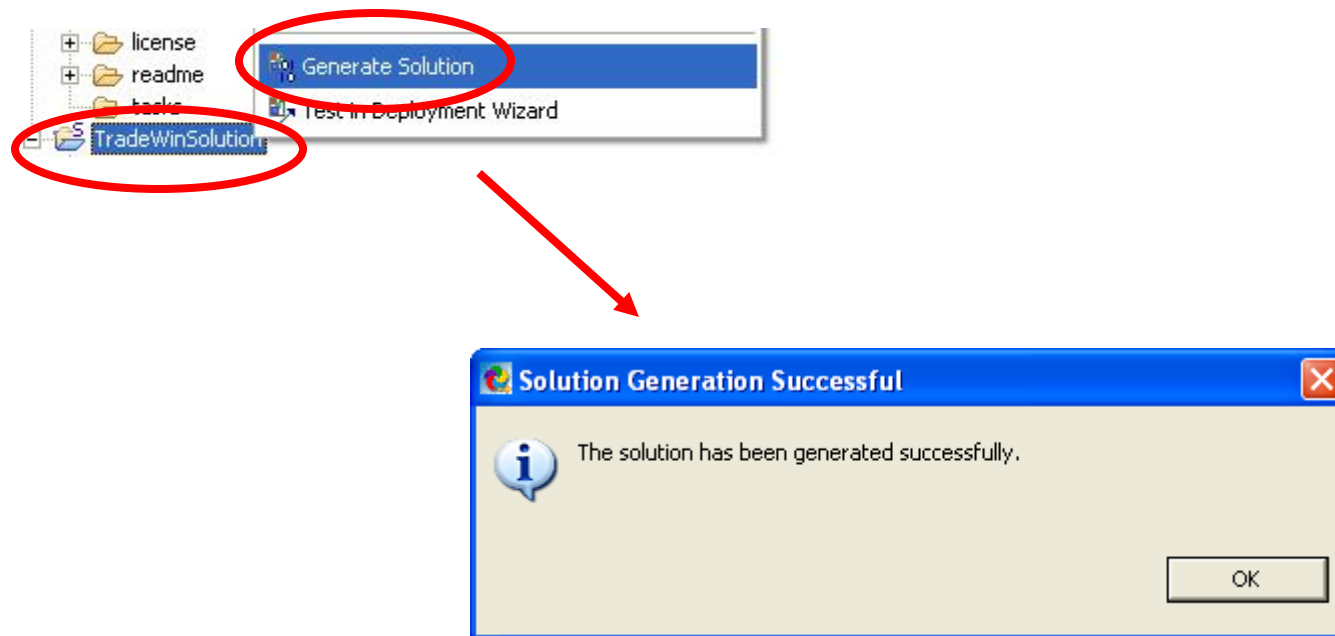
Validation rules:

- Valid characters: "@#\$_abcdefghijklmnopqrstuvwxyz" Add...
- Invalid value: "ADMINS" Remove
- Invalid value: "GUESTS"
- Invalid value: "USERS"
- Invalid value: "PUBLIC"
- Invalid value: "LOCAL"
- Invalid prefix: "IBM"
- Invalid prefix: "SQL"

Welcome | General | Tasks | Validation | Source

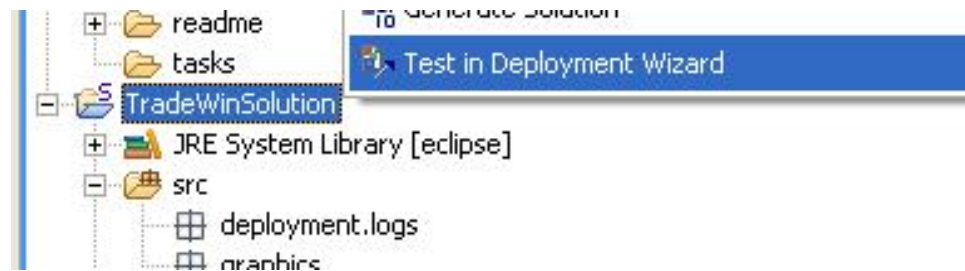
Generate Solution

- Select **Generate Solution** from the solution context menu.



Test solution

- Select **Test in Deployment Wizard** from the solution context menu to test your solution.



End of the module