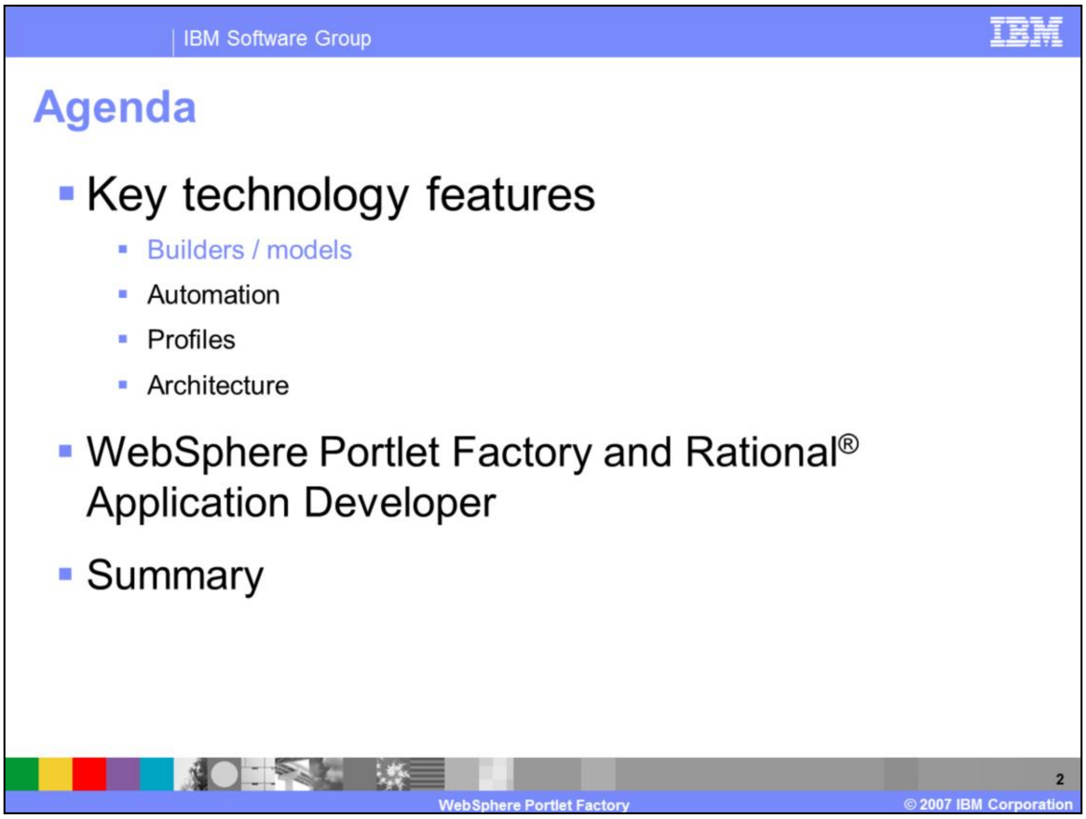
SAP integration workshop

WebSphere Portlet Factory

e business on demand.

Welcome to the WebSphere® Portlet Factory presentation as part of the SAP Integration Workshop. This presentation will give you an introduction to the Portlet Factory product and concepts. The special focus is on the different options on how to integrate SAP with the help of this powerful development environment.


IBM Software Group

IBM

Agenda

- Key technology features
  - Builders / models
  - Automation
  - Profiles
  - Architecture
- WebSphere Portlet Factory and Rational® Application Developer
- Summary

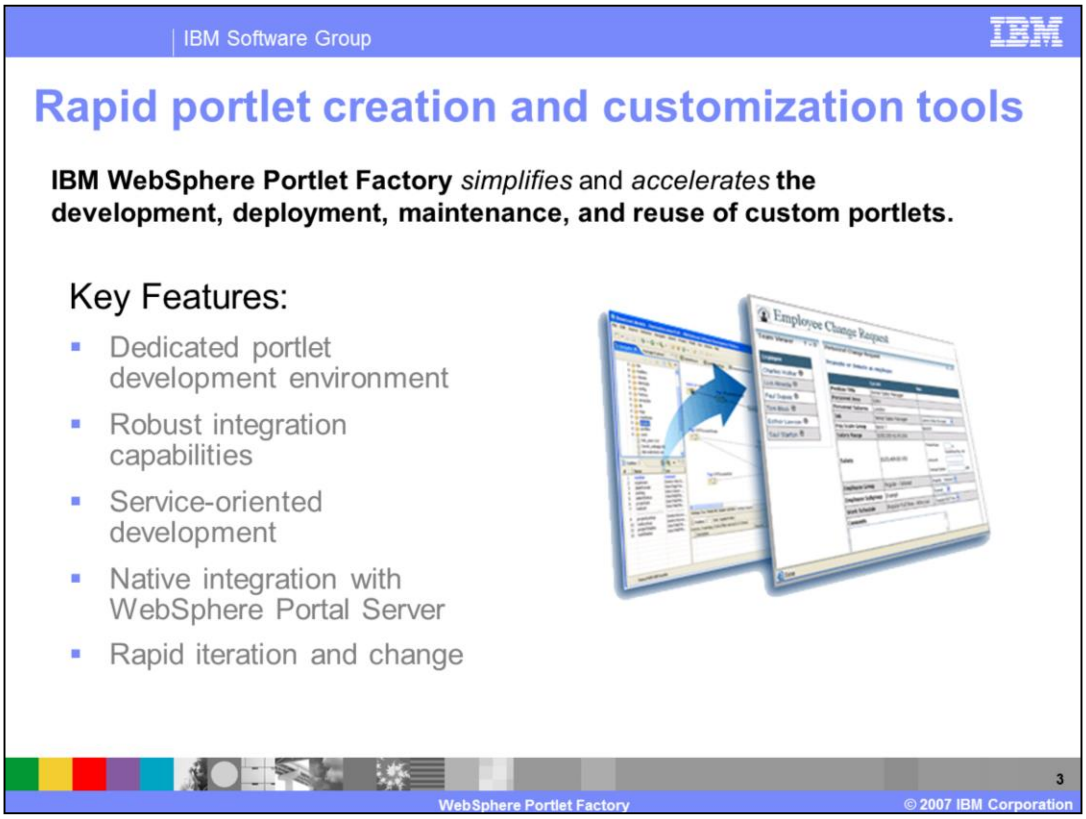
2

WebSphere Portlet Factory

© 2007 IBM Corporation

The presentation will introduce the key technologies and features of the different components and will also give a short positioning and overview about the two tool options from IBM.

Both have their special target space and can work together, as you will see.



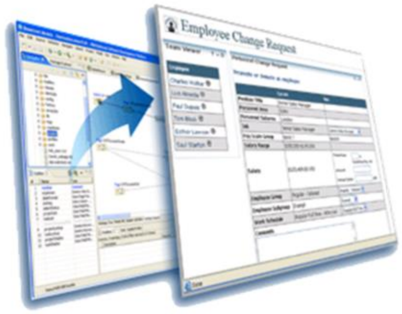
IBM Software Group

Rapid portlet creation and customization tools

IBM WebSphere Portlet Factory *simplifies* and *accelerates* **the development, deployment, maintenance, and reuse of custom portlets.**

Key Features:

- Dedicated portlet development environment
- Robust integration capabilities
- Service-oriented development
- Native integration with WebSphere Portal Server
- Rapid iteration and change



3

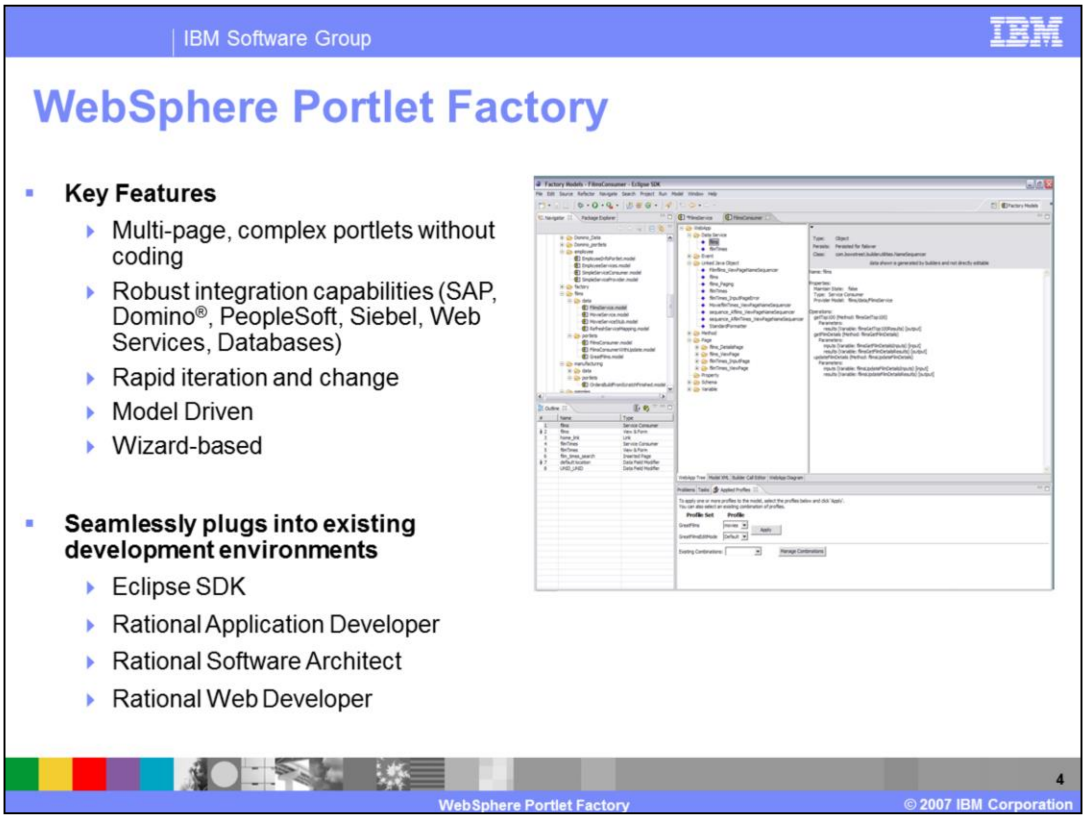
WebSphere Portlet Factory © 2007 IBM Corporation

The Portlet Factory is a rapid development tool.

The tools are focused on very fast creation of portlets and fast change of the created content. Together with the robust integration capabilities that are part of the SOA concept, these tools drive the development in WebSphere environments.

The rapid iteration and change also allows an evolution of the portlets step by step.

But now to the key features of these tools…



IBM Software Group

WebSphere Portlet Factory

- **Key Features**
  - Multi-page, complex portlets without coding
  - Robust integration capabilities (SAP, Domino®, PeopleSoft, Siebel, Web Services, Databases)
  - Rapid iteration and change
  - Model Driven
  - Wizard-based
- **Seamlessly plugs into existing development environments**
  - Eclipse SDK
  - Rational Application Developer
  - Rational Software Architect
  - Rational Web Developer

4

WebSphere Portlet Factory

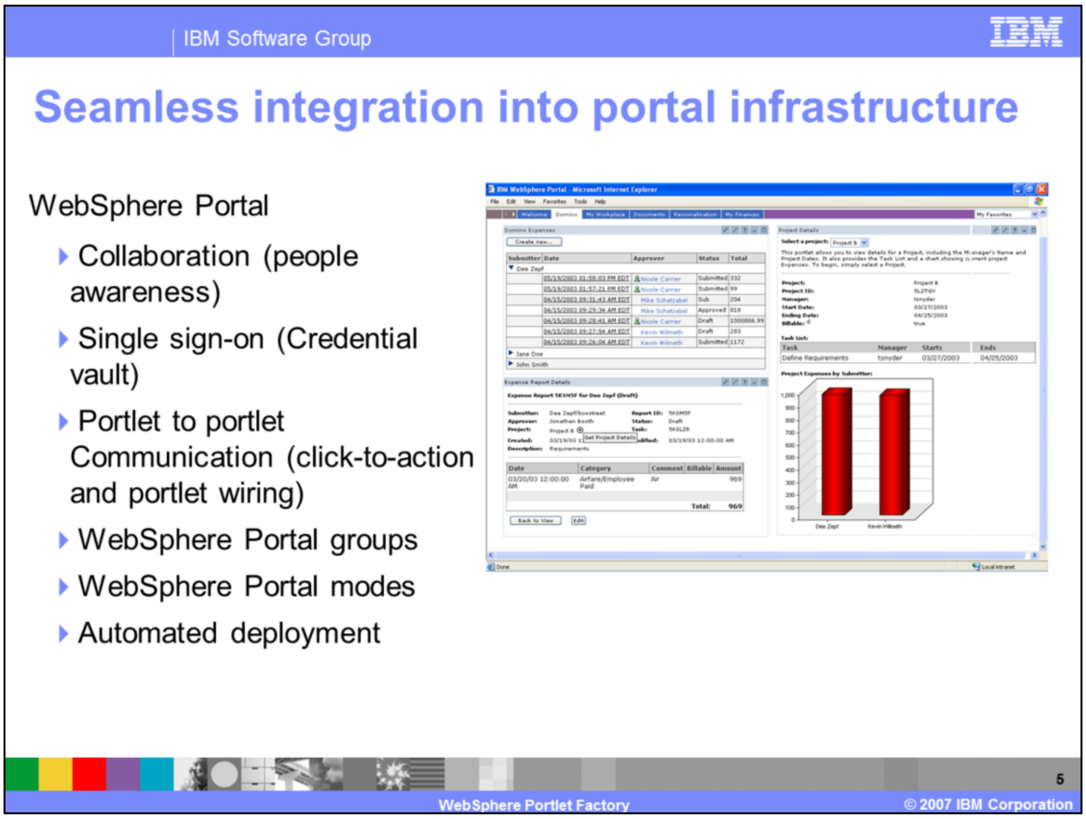
© 2007 IBM Corporation

The Portlet Factory allows a easy creation of multi-page portlets with a rich content out of different back-end systems.

The portlets can leverage the integration capabilities, which helps to integrate complex data with a single form. This form concept is like a wizard based development; you use the predefined markup to define what the content should look like, then the code is generated automatically.

The portlet factory tools are Eclipse based, so they can be installed on any Eclipse environment like Rational Application Developer, Rational Software Architect, WebSphere Integration Developer and a plain Eclipse installation.

But the seamless integration in the WebSphere portfolio is not only into the tools…

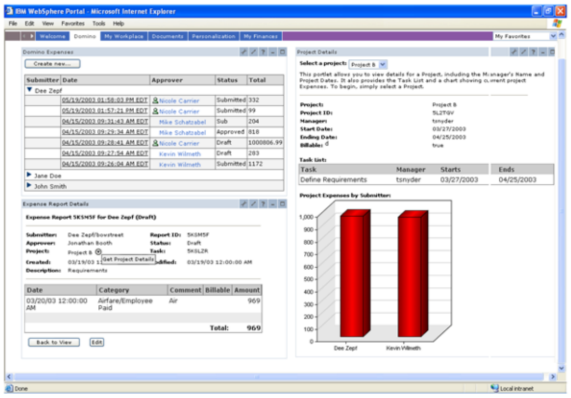


IBM Software Group

Seamless integration into portal infrastructure

WebSphere Portal

- Collaboration (people awareness)
- Single sign-on (Credential vault)
- Portlet to portlet Communication (click-to-action and portlet wiring)
- WebSphere Portal groups
- WebSphere Portal modes
- Automated deployment



Submitter	Date	Approver	Status	Total
Dee Zepf				
	05/19/2003 01:58:03 PM EDT	Nicole Carrier	Submitted	332
	05/19/2003 01:57:21 PM EDT	Nicole Carrier	Submitted	99
	04/15/2003 09:31:43 AM EDT	Mike Schatzabel	Sub	204
	04/15/2003 09:29:34 AM EDT	Mike Schatzabel	Approved	818
	04/15/2003 09:28:41 AM EDT	Nicole Carrier	Draft	1000806.99
	04/15/2003 09:27:54 AM EDT	Kevin Wilmeth	Draft	283
	04/15/2003 09:26:04 AM EDT	Kevin Wilmeth	Submitted	1172
Jane Doe				
John Smith				

Expense Report 5KSM5F for Dee Zepf (Draft)

Submitter:	Dee Zepf/bovstreet	**Report ID:**	5KSM5F
Approver:	Jonathan Booth	**Status:**	Draft
Project:	Project B	**Task:**	5KSLZR
Created:	03/19/03 1	**Modified:**	03/19/03 12:00:00 AM
Description:	Requirements		

Date	Category	Comment	Billable	Amount
03/20/03 12:00:00 AM	Airfare/Employee Paid	Air		969
			Total:	**969**

Select a project: Project B

This portlet allows you to view details for a Project, including the Manager's Name and Project Dates. It also provides the Task List and a chart showing current project Expenses. To begin, simply select a Project.

Project:	Project B
Project ID:	5L2TGV
Manager:	tsnyder
Start Date:	03/27/2003
Ending Date:	04/25/2003
Billable:	true

Task List:

Task	Manager	Starts	Ends
Define Requirements	tsnyder	03/27/2003	04/25/2003

Project Expenses by Submitter:

5

WebSphere Portlet Factory

© 2007 IBM Corporation

The portlet factory also supports the different built-in mechanism of the WebSphere Portal.

The Sametime people-awareness can be added to a portlet with a simple wizard, as can the usage of the credential vault to store user credentials for the login to backend systems.

The portlets can work together using portlet-to-portlet communication – including portlets that are created with other tools.

The profiling of the portlets can react on the WebSphere Portal Server group definitions and the visualization can be different depending on the actual portlet mode – normal or maximized.

And of course the tools leverage the XML interface to deploy the portlets to the test or production systems.

The development options allow you to put a facade on your current working environment with new portlets.

You can use already installed and created applications and set a portlet-based user interface on top of them.

There the portlet factory does not care if the application is SAP, Lotus, PeopleSoft or a lot of other data sources.

WebSphere Portlet Factory key concepts

Builder

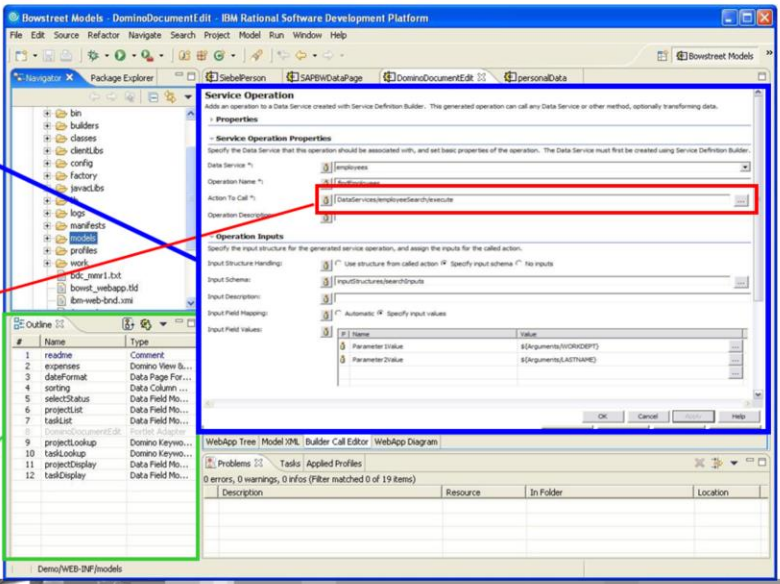
An adaptive, reusable component that generates Java™, XML and JSP for specific portlet functionality

Profile

A set of parameters that vary builder inputs to dynamically generate unique versions of the portlet

Model

A container of builders



#	Name	Type
1	readme	Comment
2	expenses	Domino View &...
3	dateFormat	Data Page For...
4	sorting	Data Column ...
5	selectStatus	Data Field Mo...
6	projectList	Data Field Mo...
7	taskList	Data Field Mo...
8	DominoDocumentEdit	Portlet Adapter
9	projectLookup	Domino Keywo...
10	taskLookup	Domino Keywo...
11	projectDisplay	Data Field Mo...
12	taskDisplay	Data Field Mo...

Service Operation

Adds an operation to a Data Service created with Service Definition Builder. This generated operation can call any Data Service or other method, optionally transforming data.

Properties

Service Operation Properties

Specify the Data Service that this operation should be associated with, and set basic properties of the operation. The Data Service must first be created using Service Definition Builder.

Data Service *: employees

Operation Name *: findEmployees

Action To Call *: DataServices/employeeSearch/execute

Operation Description:

Operation Inputs

Specify the input structure for the generated service operation, and assign the inputs for the called action.

Input Structure Handling: Use structure from called action [✓ Specify input schema] No inputs

Input Schema: inputStructures/searchInputs

Input Description:

Input Field Mapping: Automatic [✓ Specify input values]

Input Field Values:

P	Name	Value
	Parameter1Value	${Arguments/WORKDEPT}
	Parameter2Value	${Arguments/LASTNAME}

0 errors, 0 warnings, 0 infos (Filter matched 0 of 19 items)

Description	Resource	In Folder	Location

6

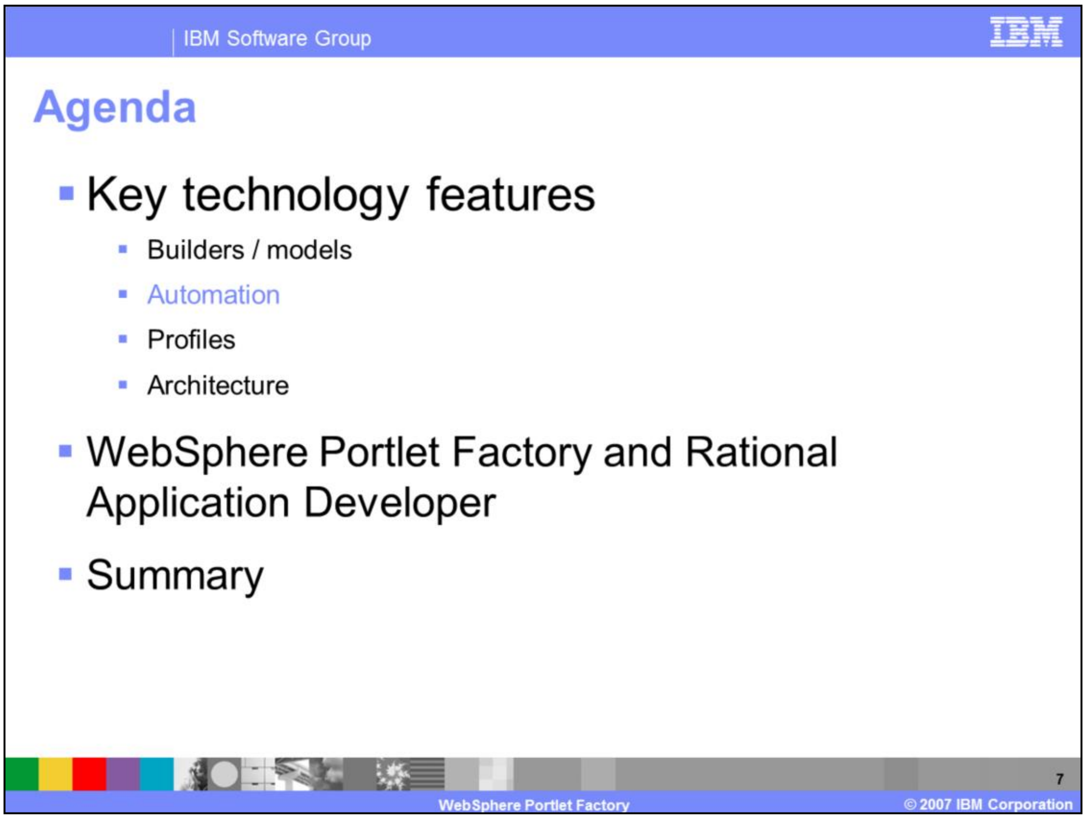
WebSphere Portlet Factory © 2007 IBM Corporation

Here the typical Portlet Factory perspective in Rational Application Developer is shown.

Builders offer easy-to-use, wizard-like user interfaces, which make it fast and easy to develop portlets. The builder generates all of the application code including Java, XML schemas, variables, and so on, based on the inputs given. Builders are also used throughout the entire development process, allowing developers to go back and change a builder's input values and have the entire portlet application update instantly.

A profile contains a set of parameters that vary the way an application behaves. A profile feeds values into builders based on the user's identity or other contextual information such as language, geography or group membership. Then once the profile is automatically applied, the application regenerates itself to adapt to that specific user.

A model is the container that holds the ordered list of builder calls. Typically, developers who want to create a new model will create a new model and then add the appropriate builders to the model.



IBM Software Group

Agenda

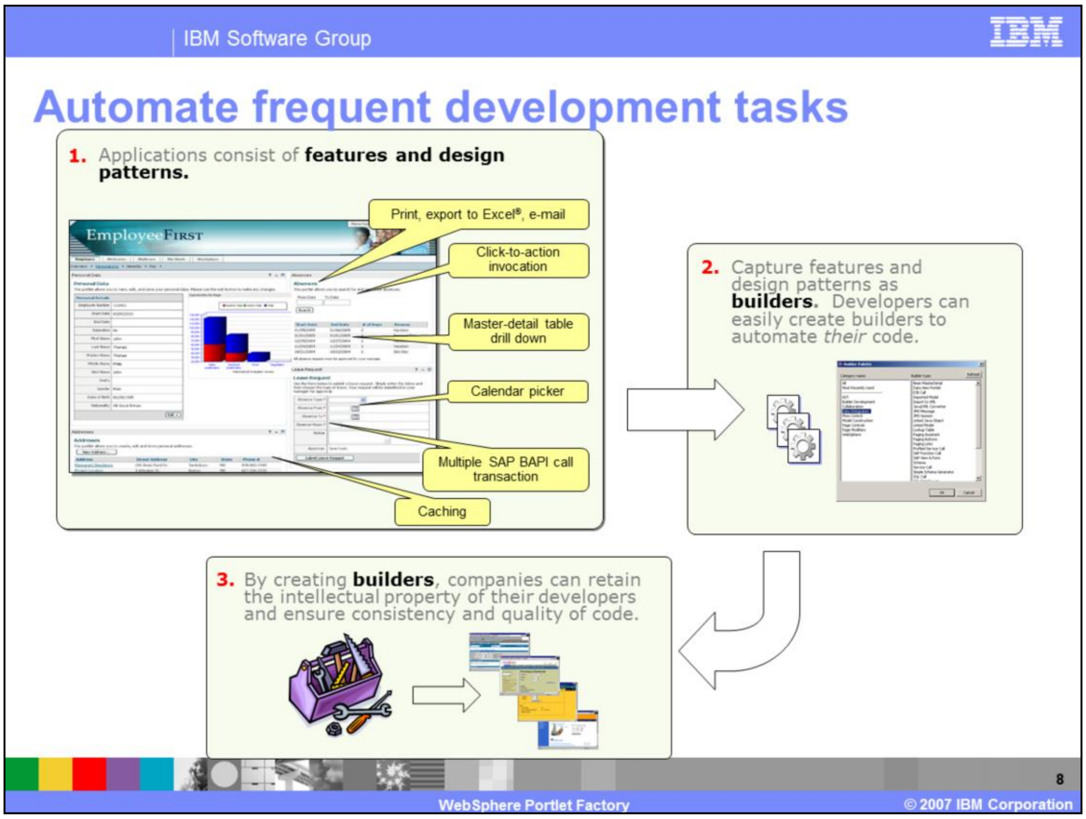
- Key technology features
  - Builders / models
  - Automation
  - Profiles
  - Architecture
- WebSphere Portlet Factory and Rational Application Developer
- Summary

7

WebSphere Portlet Factory

© 2007 IBM Corporation

This section will take a deeper look into the automation and regeneration possibilities.


IBM Software Group

Automate frequent development tasks

1. Applications consist of **features and design patterns.**

Print, export to Excel®, e-mail
Click-to-action invocation
Master-detail table drill down
Calendar picker
Multiple SAP BAPI call transaction
Caching

2. Capture features and design patterns as **builders.** Developers can easily create builders to automate *their* code.

3. By creating **builders**, companies can retain the intellectual property of their developers and ensure consistency and quality of code.

8

WebSphere Portlet Factory

© 2007 IBM Corporation

Often developers use the tools to generate specific portlets. But after a while it becomes clear that the same pattern is needed very often.

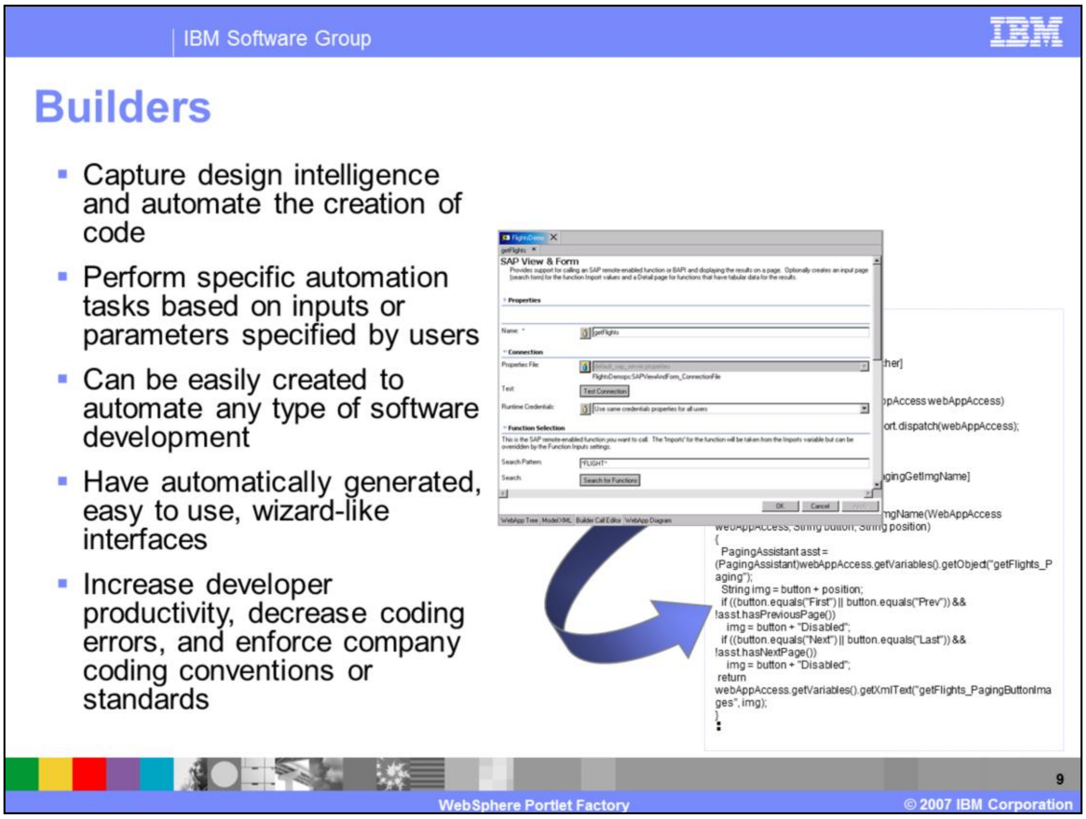
This is the time to generate your own custom builders for the tools to save time, effort and ensure the quality of code.

There are two options to generate your own builder:

The first one is to use a special builder, where you just assemble the builders as usual and define the manual inputs. Everything else is then generated automatically and the new builder appears in the builder list.

The second option is more complex but also more powerful. For example, if you connect to your own back-end system with a special protocol, you can write Java code to do whatever you like. This is well documented in the tools.

The newly created builders can than be reused every time, making the work very easy for all developers.



Builders

- Capture design intelligence and automate the creation of code
- Perform specific automation tasks based on inputs or parameters specified by users
- Can be easily created to automate any type of software development
- Have automatically generated, easy to use, wizard-like interfaces
- Increase developer productivity, decrease coding errors, and enforce company coding conventions or standards

SAP View & Form

Provides support for calling an SAP remote-enabled function or BAPI and displaying the results on a page. Optionally creates an input page (search form) for the function Import values and a Detail page for functions that have tabular data for the results.

Properties

Name: * getFlights

Connection

Properties File: FlightsDemops:SAPViewAndForm_ConnectionFile

Test: Test Connection

Runtime Credentials: Use same credentials properties for all users

Function Selection

This is the SAP remote-enabled function you want to call. The 'Imports' for the function will be taken from the Imports variable but can be overridden by the Function Inputs settings.

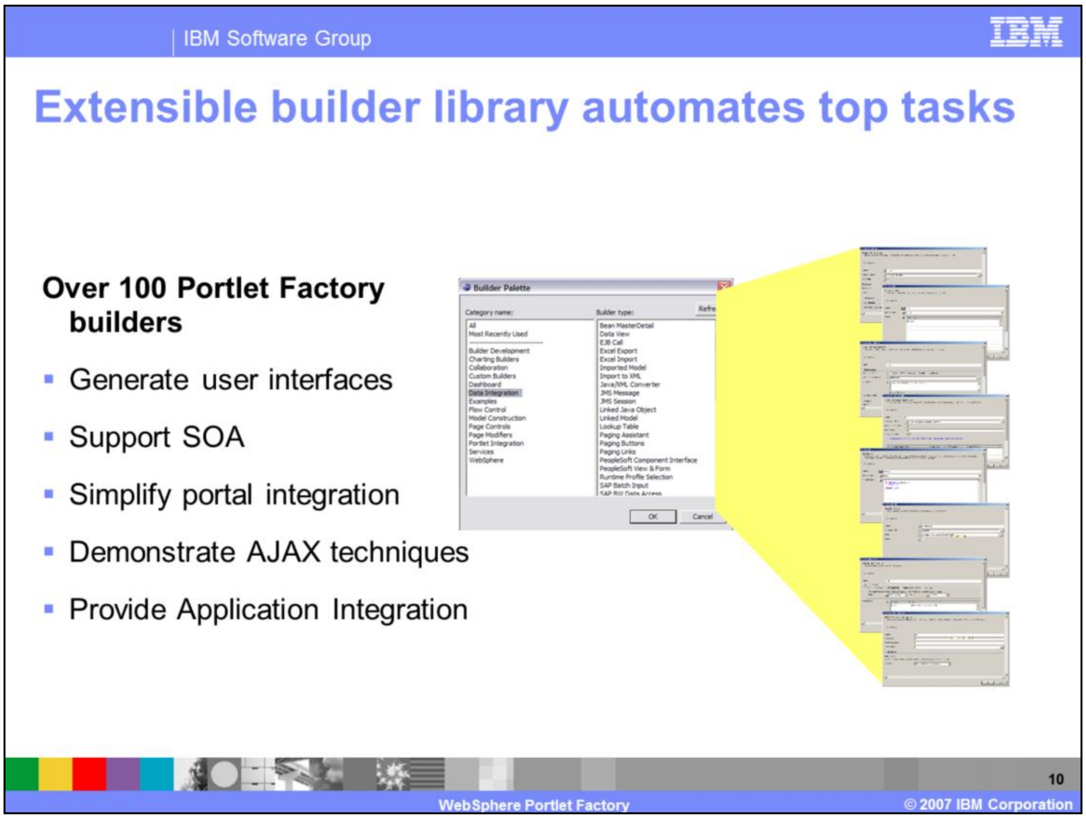
Search Pattern: *FLIGHT*

Search: Search for Functions

```
ther]

pAccess webAppAccess)

ort.dispatch(webAppAccess);

gingGetImgName]

mgName(WebAppAccess
webAppAccess, String button, String position)
{
  PagingAssistant asst =
(PagingAssistant)webAppAccess.getVariables().getObject("getFlights_P
aging");
  String img = button + position;
  if ((button.equals("First") || button.equals("Prev")) &&
!asst.hasPreviousPage())
    img = button + "Disabled";
  if ((button.equals("Next") || button.equals("Last")) &&
!asst.hasNextPage())
    img = button + "Disabled";
 return
webAppAccess.getVariables().getXmlText("getFlights_PagingButtonIma
ges", img);
}
:
```

All builders - no matter if they are shipped with the product or self created – automate the creation of real code.

As you can see, the forms will be stored in the background as xml documents. This meta-data gets compiled in running code when accessed.



IBM Software Group

Extensible builder library automates top tasks

Over 100 Portlet Factory builders

- Generate user interfaces
- Support SOA
- Simplify portal integration
- Demonstrate AJAX techniques
- Provide Application Integration

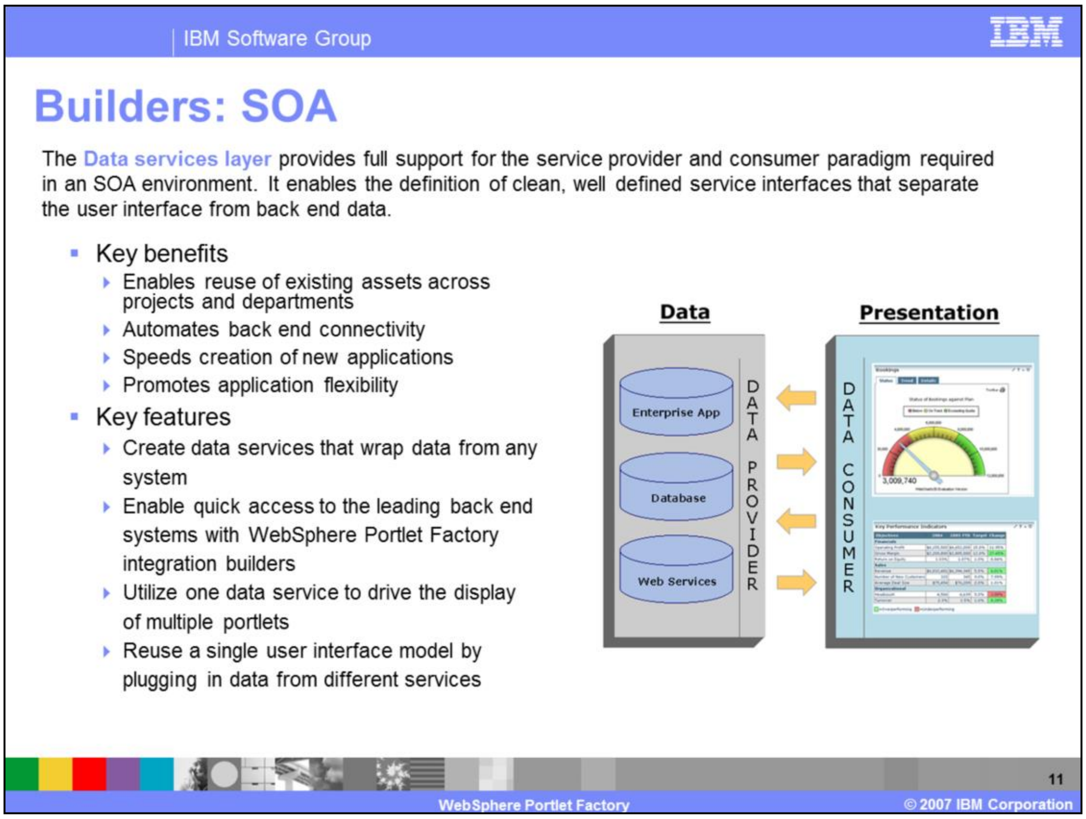
Builder Palette
Category name:
All
Most Recently Used
Builder Development
Charting Builders
Collaboration
Custom Builders
Dashboard
Data Integration
Examples
Flow Control
Model Construction
Page Controls
Page Modifiers
Portlet Integration
Services
WebSphere
Builder type:
Bean MasterDetail
Data View
EJB Call
Excel Export
Excel Import
Imported Model
Import to XML
Java/XML Converter
JMS Message
JMS Session
Linked Java Object
Linked Model
Lookup Table
Paging Assistant
Paging Buttons
Paging Links
PeopleSoft Component Interface
PeopleSoft View & Form
Runtime Profile Selection
SAP Batch Input
OK
Cancel

10

WebSphere Portlet Factory

© 2007 IBM Corporation

There are over a hundred builders that are shipped with the product. They are grouped in the different target topics like portal integration and backend integration.

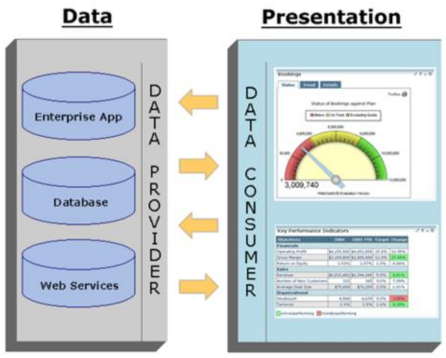


IBM Software Group

Builders: SOA

The Data services layer provides full support for the service provider and consumer paradigm required in an SOA environment. It enables the definition of clean, well defined service interfaces that separate the user interface from back end data.

- Key benefits
  - Enables reuse of existing assets across projects and departments
  - Automates back end connectivity
  - Speeds creation of new applications
  - Promotes application flexibility
- Key features
  - Create data services that wrap data from any system
  - Enable quick access to the leading back end systems with WebSphere Portlet Factory integration builders
  - Utilize one data service to drive the display of multiple portlets
  - Reuse a single user interface model by plugging in data from different services



11

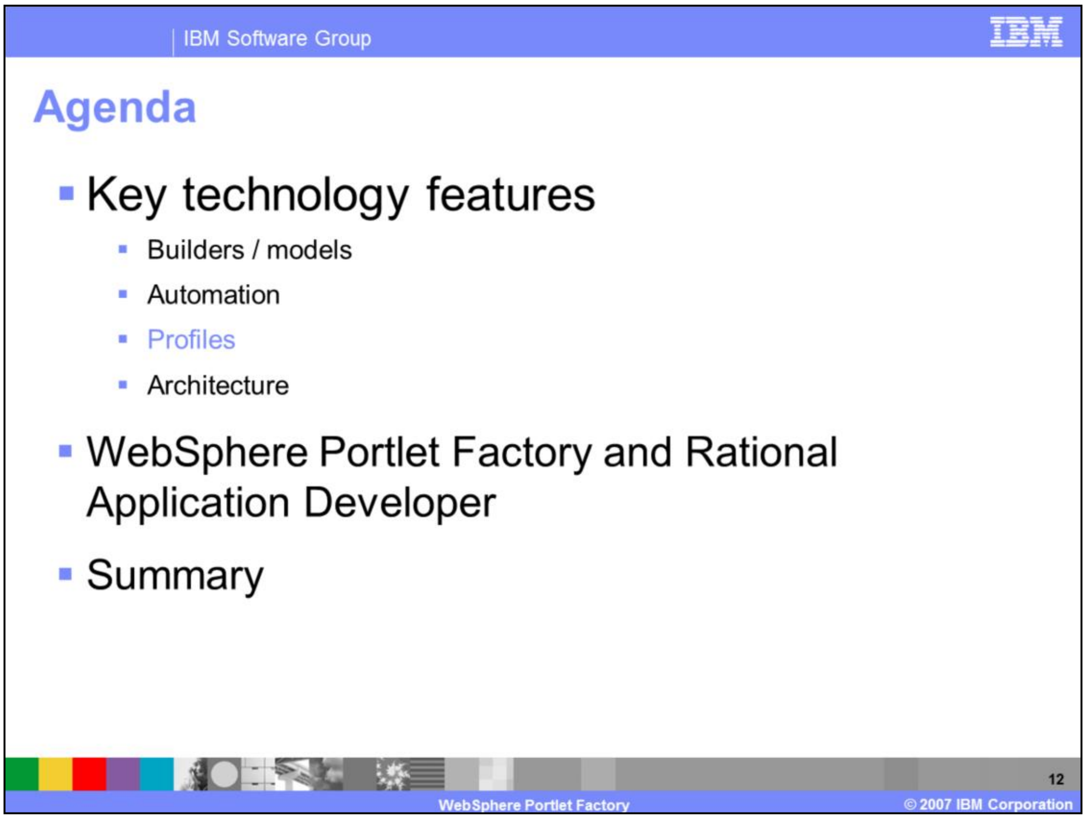
WebSphere Portlet Factory

© 2007 IBM Corporation

The builders drive also includes the Service Oriented Architecture. The data provider can be created independently from the data consumer, so the portlet factory generated data provider can be also used by other consumers and the other way around.

This makes the tools more powerful in providing and consuming the different services with different protocols.

All consumers work with automatic generated user interfaces depending on the data structure.



IBM Software Group

Agenda

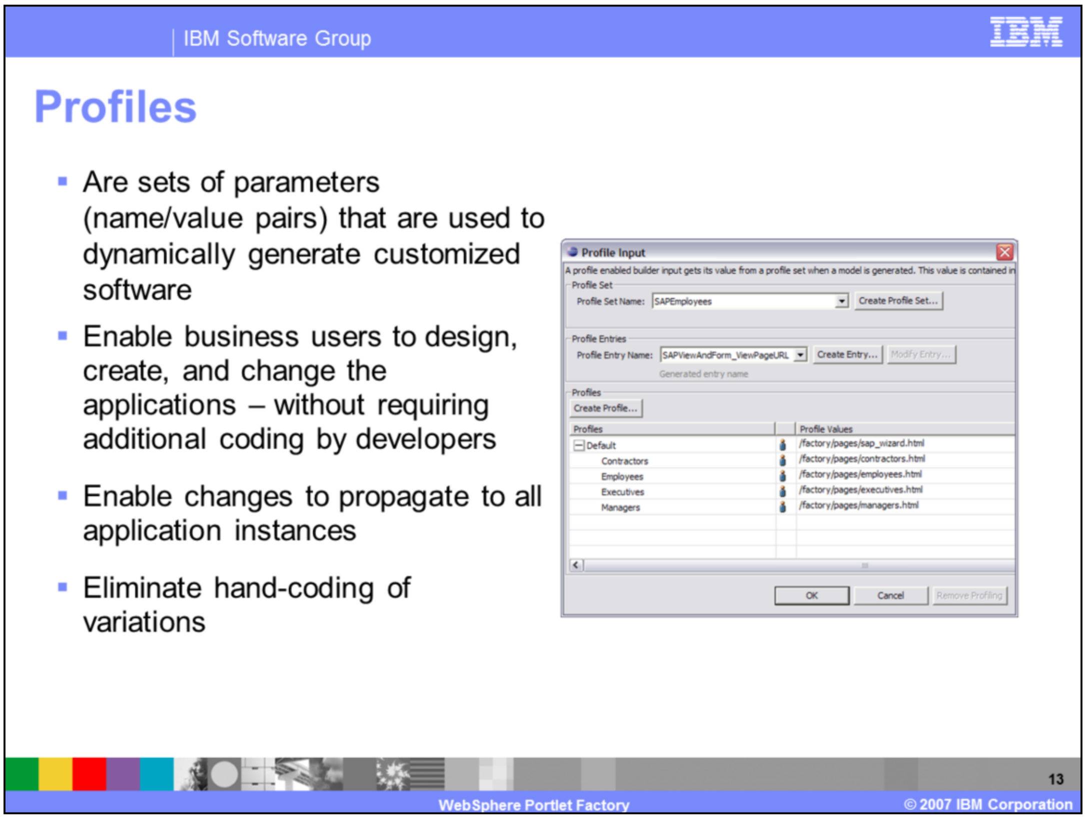
- Key technology features
  - Builders / models
  - Automation
  - Profiles
  - Architecture
- WebSphere Portlet Factory and Rational Application Developer
- Summary

12

WebSphere Portlet Factory

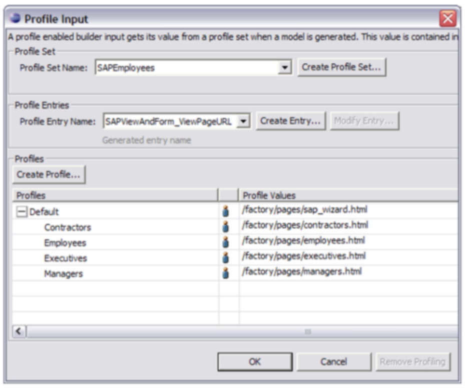
© 2007 IBM Corporation

This section talks about the mentioned profiles and the possibility to tailor the portlets to the actual needs.



Profiles

- Are sets of parameters (name/value pairs) that are used to dynamically generate customized software
- Enable business users to design, create, and change the applications – without requiring additional coding by developers
- Enable changes to propagate to all application instances
- Eliminate hand-coding of variations



Profile Input

A profile enabled builder input gets its value from a profile set when a model is generated. This value is contained in

Profile Set

Profile Set Name: SAPEmployees

Profile Entries

Profile Entry Name: SAPViewAndForm_ViewPageURL

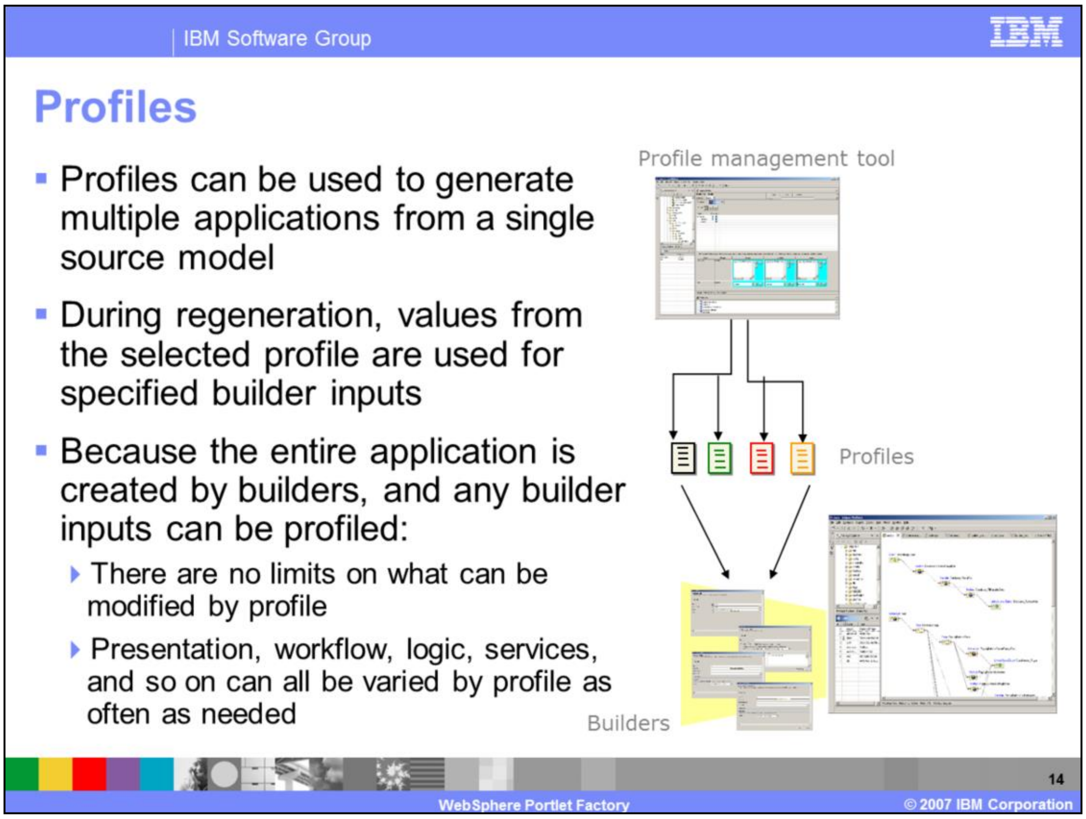
Generated entry name

Profiles

Profiles	Profile Values
Default	/factory/pages/sap_wizard.html
Contractors	/factory/pages/contractors.html
Employees	/factory/pages/employees.html
Executives	/factory/pages/executives.html
Managers	/factory/pages/managers.html

Speeding development the first time is only half the real power of WebSphere Portlet Factory. Profiles allow new versions of existing portlets without new development cycles. In the past if you needed a portlet for one group, say an Order Summary portlet for internal sales reps that was very detailed, then determined that pushing that out to distributors on an extranet would increase efficiency even more – but with less detailed order information, you would do one of two things. You would either add conditional logic to the existing portlet, or you would copy, edit and deploy a new portlet. Both are approaches that will work, but in either case, new development cycles and lengthy testing and deployment cycles are necessary. Profiles eliminate further lengthy development, test and deployment cycles making it even quicker to build portlets for subsequent requirements.

That is because developers do not create new portlets each time a new portlet is required, they just profile a single instance - essentially, a single code base - and then apply profiles that allow it to be called by different portals at run-time for use in different contexts. This regeneration capability, based on Profiles, significantly eases ongoing maintenance by propagating changes on-demand to all portlet instances. Additionally, profiles can be exposed through the portal’s standard edit, configure and administration features – a powerful capability that allows portlets and variations to be created and customized by any delegated administrator or user, with no limits. IT and developers are no longer the bottleneck for much of the varied portlets required as the needs of the business evolve.



IBM Software Group

Profiles

- Profiles can be used to generate multiple applications from a single source model
- During regeneration, values from the selected profile are used for specified builder inputs
- Because the entire application is created by builders, and any builder inputs can be profiled:
  - There are no limits on what can be modified by profile
  - Presentation, workflow, logic, services, and so on can all be varied by profile as often as needed

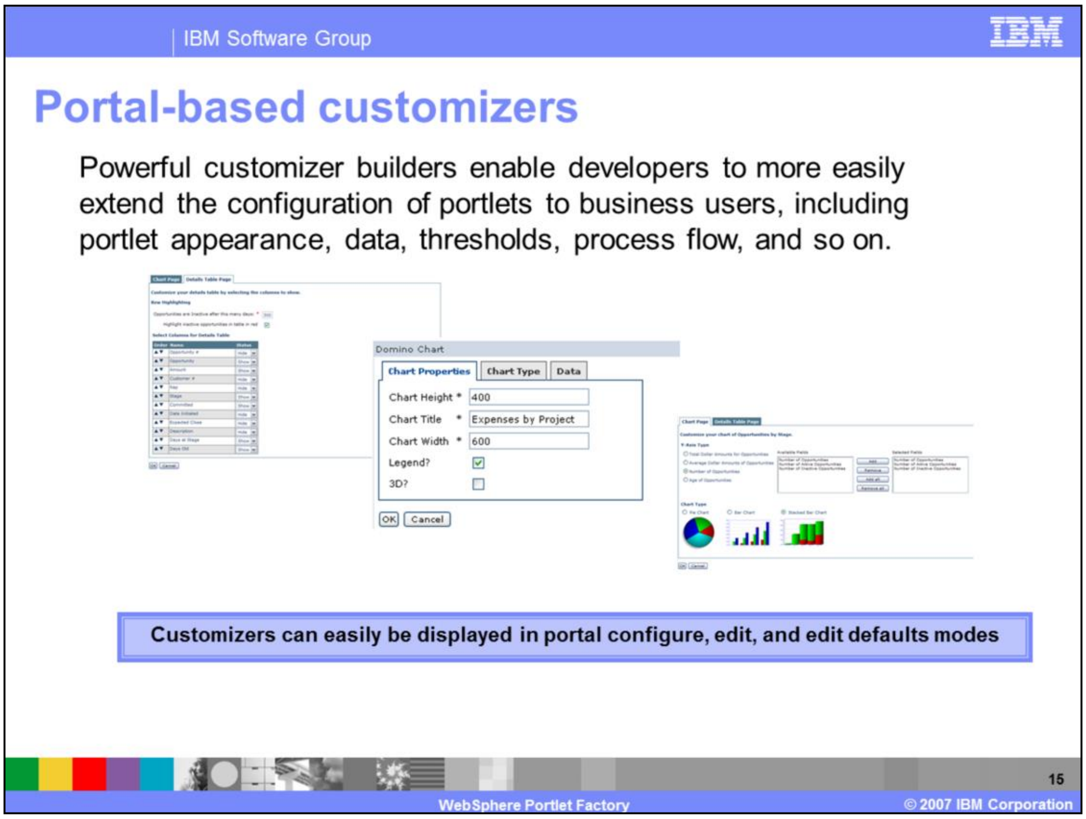
14

WebSphere Portlet Factory © 2007 IBM Corporation

To organize your profiles you will use the built-in profile management tool where you have one point of access to all profiles.

These profiles will then get used in the builders to generate your customized portlet for every profile.

This is possible because the entire application is created by builders and the real code is always generated on demand.



IBM Software Group

Portal-based customizers

Powerful customizer builders enable developers to more easily extend the configuration of portlets to business users, including portlet appearance, data, thresholds, process flow, and so on.

Domino Chart

Chart Properties | Chart Type | Data

Chart Height *	400
Chart Title *	Expenses by Project
Chart Width *	600
Legend?	✓
3D?	

OK | Cancel

Customizers can easily be displayed in portal configure, edit, and edit defaults modes

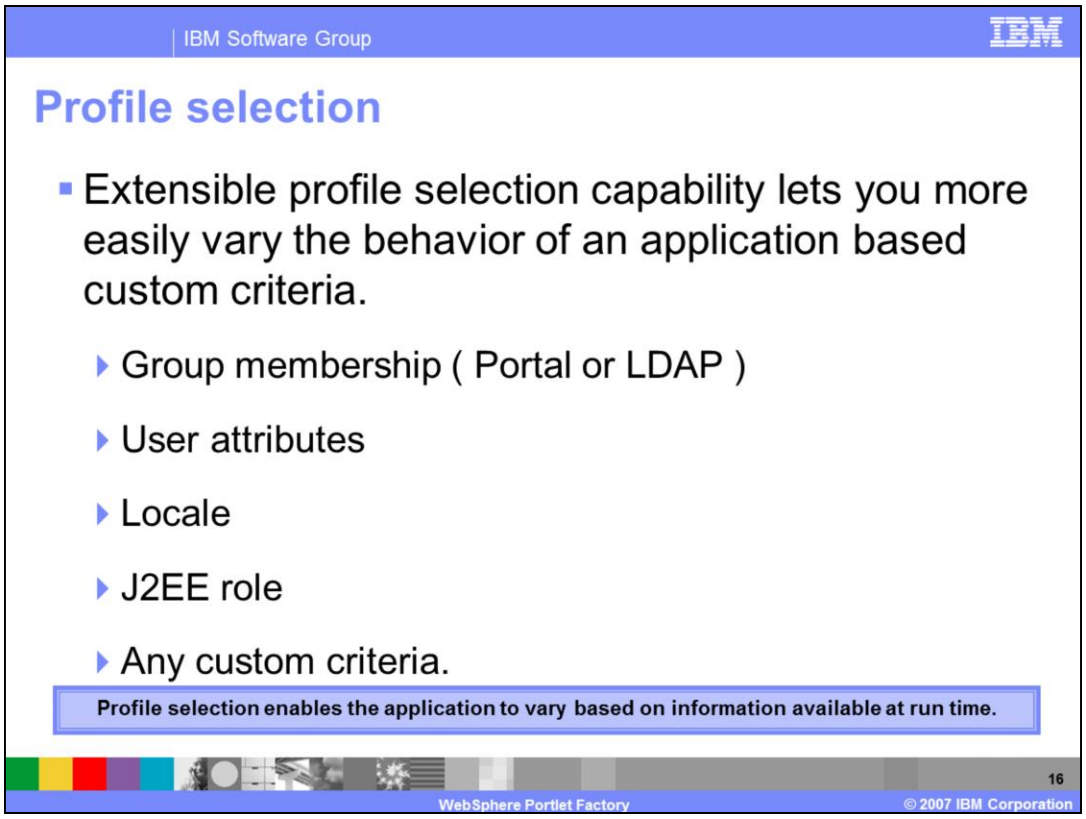
15

WebSphere Portlet Factory © 2007 IBM Corporation

Another feature included in the Dashboard Framework is the ability to very easily create robust, browser-based configuration wizards that enable non-technical users to dynamically personalize and configure their dashboard portlets –without having to involve IT. Users can change really any aspect of the portlet exposed by IT, which can include the appearance of the portlet, the functionality, application flow, and even connectivity to the back end system.

Basic support for creating these browser-based configuration screens is included as part of the WebSphere Portlet Factory, but the framework adds some features that makes this even easier. For example, the Table Customizer builder makes it very straightforward to customize a table, including features such as the ability to reorder columns, rename columns, change column widths and alignments, add sorting, and so on.

With this capability to extend portlet configuration to business users, companies can increase the flexibility of their dashboards and reduce IT maintenance costs.



IBM Software Group

Profile selection

- Extensible profile selection capability lets you more easily vary the behavior of an application based custom criteria.
  - Group membership (Portal or LDAP)
  - User attributes
  - Locale
  - J2EE role
  - Any custom criteria.

Profile selection enables the application to vary based on information available at run time.

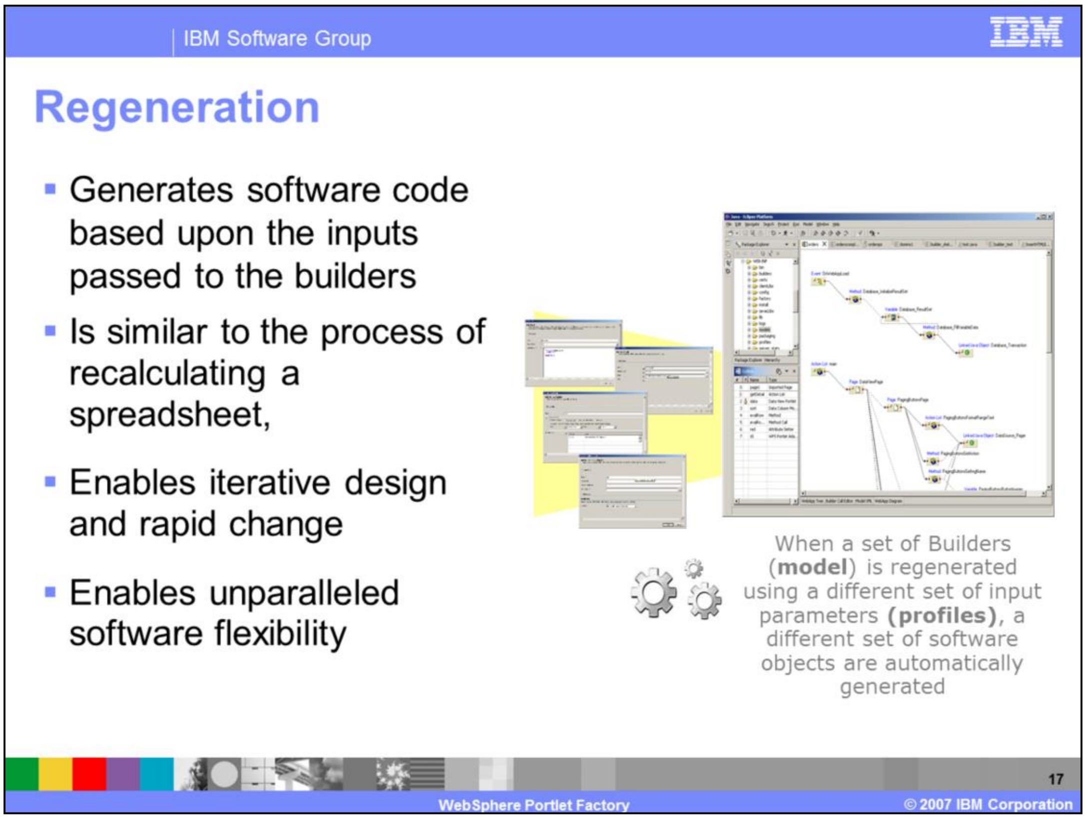
16

WebSphere Portlet Factory © 2007 IBM Corporation

The profiles are more powerful then simple “if clauses”; the behavior of the portlet can be different depending on different options.

Through the tight integration with WebSphere Portal, the group membership can be used.

Different user attributes, language settings or any custom criteria can also be used for profiling.



Regeneration

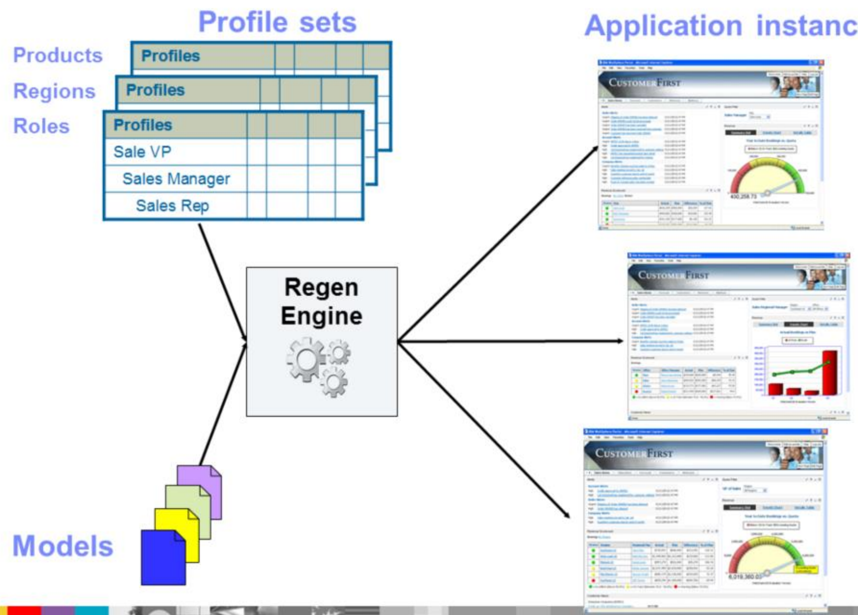
- Generates software code based upon the inputs passed to the builders
- Is similar to the process of recalculating a spreadsheet,
- Enables iterative design and rapid change
- Enables unparalleled software flexibility

When a set of Builders (**model**) is regenerated using a different set of input parameters **(profiles)**, a different set of software objects are automatically generated

The regeneration approach has many positive aspects. One is of course the possibility of iterative design and rapid change. Because you can change the model and the code is automatically regenerated you see what you developed – without additional manual compile steps.

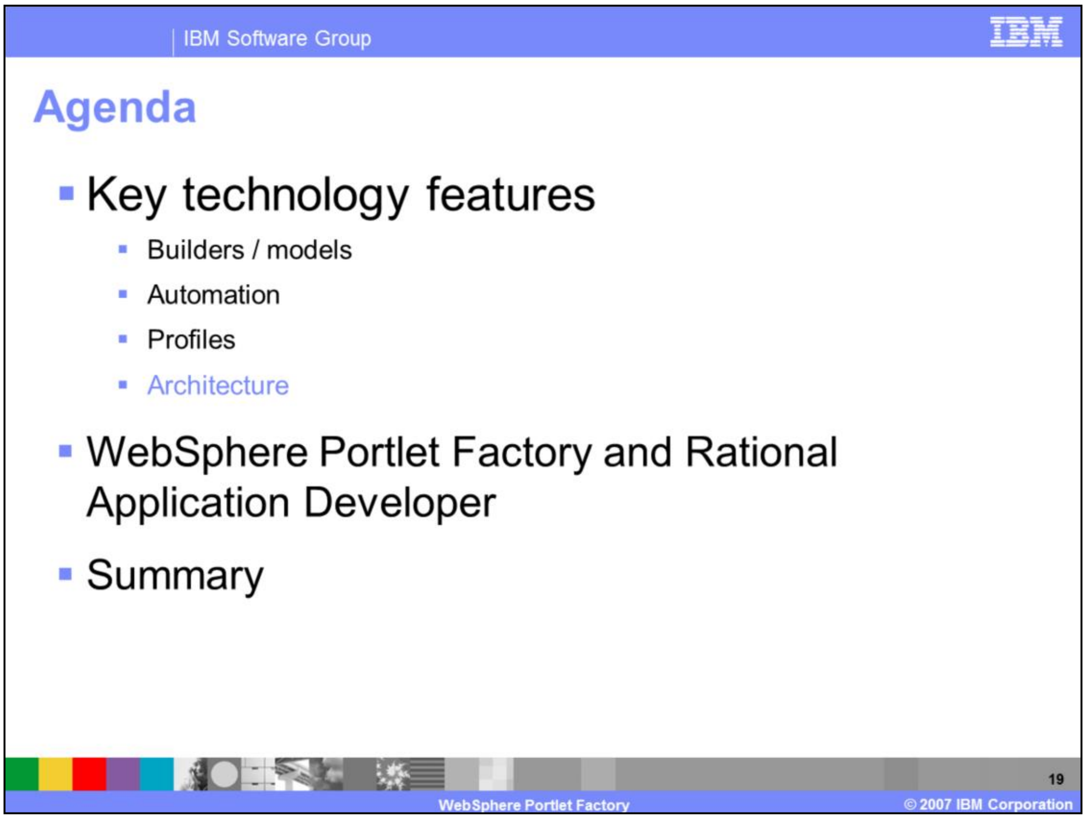
Another main factor is the reduction of maintenance costs because there is only one deployed portlet and one code base to maintain.

Putting it all together



The regeneration engine is responsible for building the portlets for the specific needs at the moment.

For that, the models get combined with the different profile sets to build different application instances of one application model.

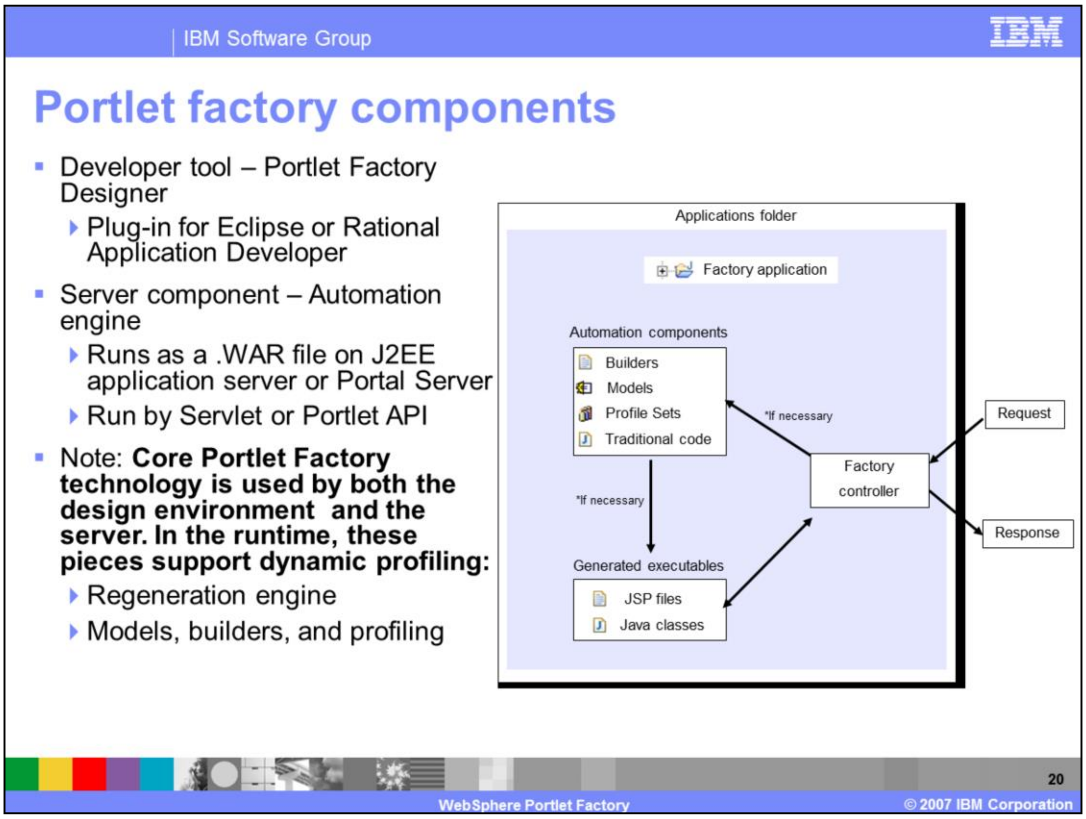


Agenda

- Key technology features
  - Builders / models
  - Automation
  - Profiles
  - Architecture
- WebSphere Portlet Factory and Rational Application Developer
- Summary

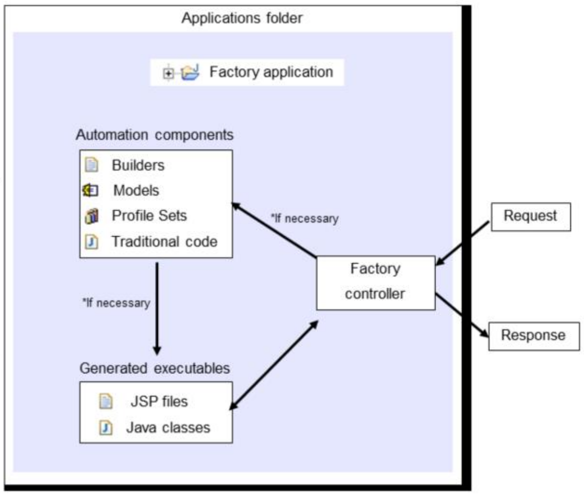
This section will take a short view in the architecture of the portlet factory tools.

To illustrate this you will see a flow from call to serve, and the placement of the portlet factory modules in the installed software stack.



Portlet factory components

- Developer tool – Portlet Factory Designer
  - Plug-in for Eclipse or Rational Application Developer
- Server component – Automation engine
  - Runs as a .WAR file on J2EE application server or Portal Server
  - Run by Servlet or Portlet API
- Note: **Core Portlet Factory technology is used by both the design environment and the server. In the runtime, these pieces support dynamic profiling:**
  - Regeneration engine
  - Models, builders, and profiling

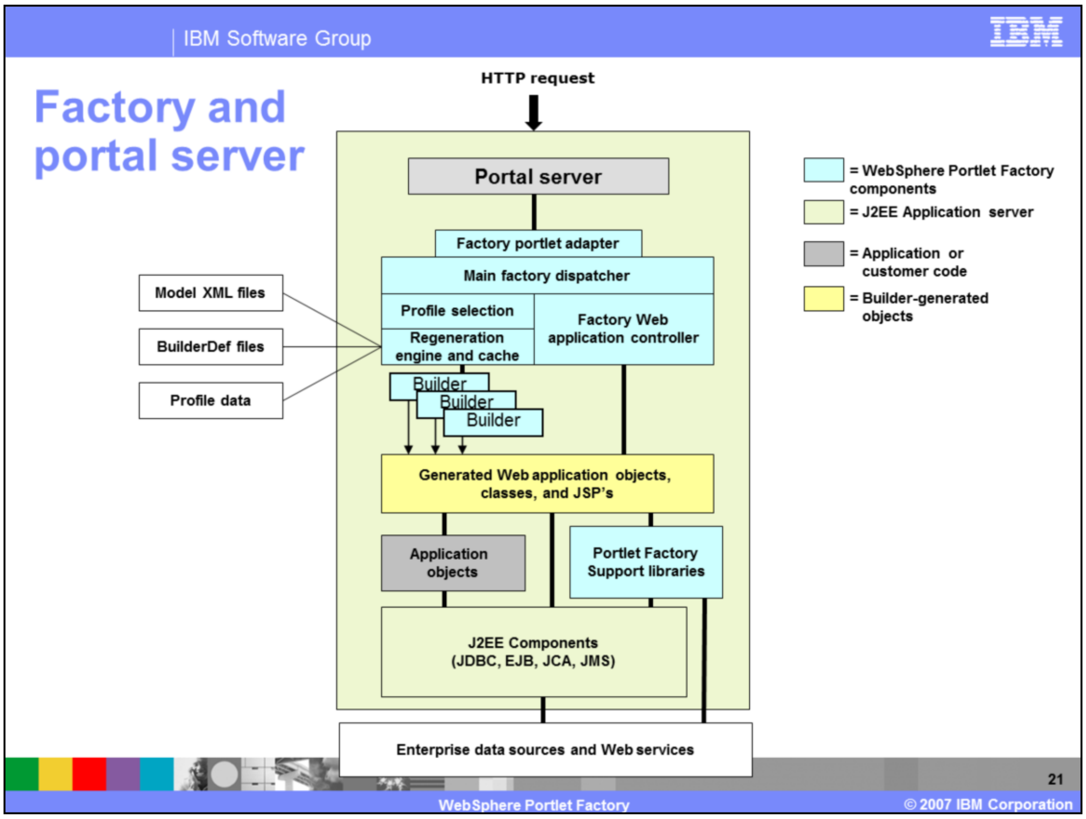


This shows the flow after the model is already developed and installed in the running environment.

The call enters the ‘Factory controller’ running in the application.

The controller checks if the current portlet is already compiled. If not, it accesses the application components and starts the generation.

These generated modules are then used to deliver the correct response.

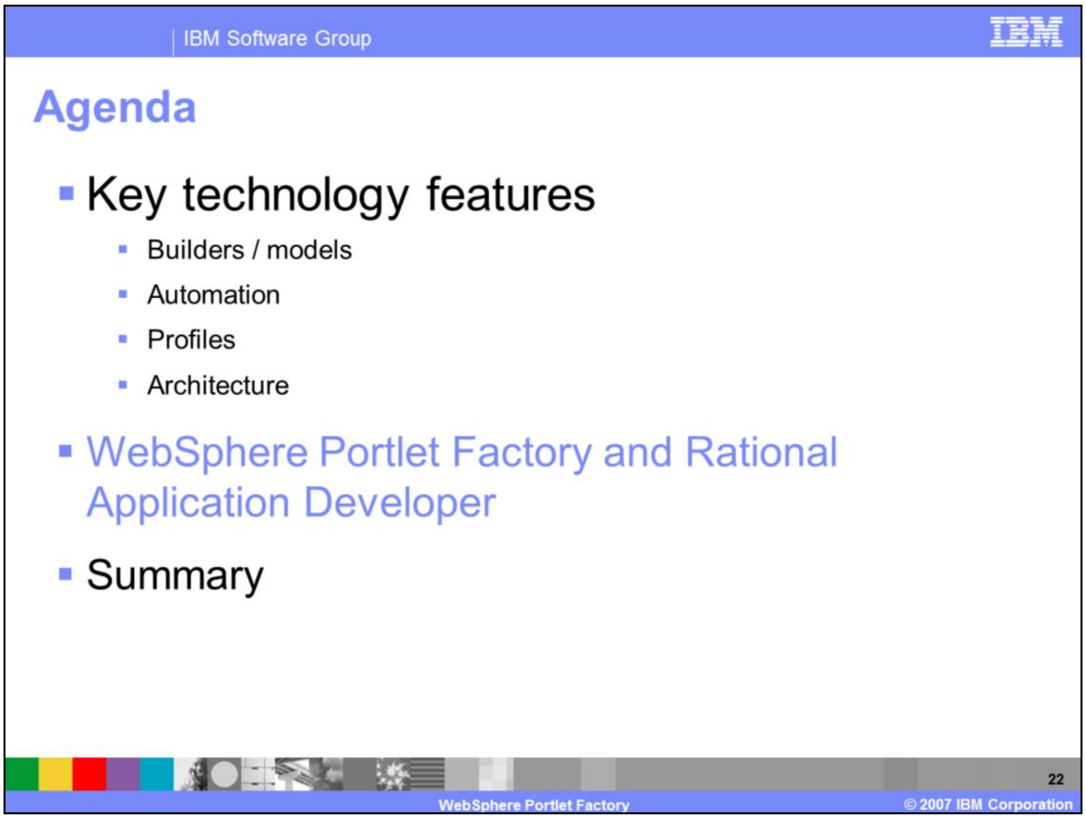

IBM Software Group
IBM
Factory and portal server
HTTP request
Portal server
Factory portlet adapter
Main factory dispatcher
Profile selection
Regeneration engine and cache
Factory Web application controller
Model XML files
BuilderDef files
Profile data
Builder
Builder
Builder
Generated Web application objects, classes, and JSP's
Application objects
Portlet Factory Support libraries
J2EE Components (JDBC, EJB, JCA, JMS)
Enterprise data sources and Web services
= WebSphere Portlet Factory components
= J2EE Application server
= Application or customer code
= Builder-generated objects
21
WebSphere Portlet Factory
© 2007 IBM Corporation

In the portal server the portlet factory components are placed in several places.

The portal server is the front end to the user, and on request it calls the factory portlet adapter. The portlet adapter is the capsulation containing all portlet specific values and sits on the dispatcher.

The tool then decides if the objects are already accessible or if they need to be built.

Also some special support libraries are shipped with the product and included in the classpath.



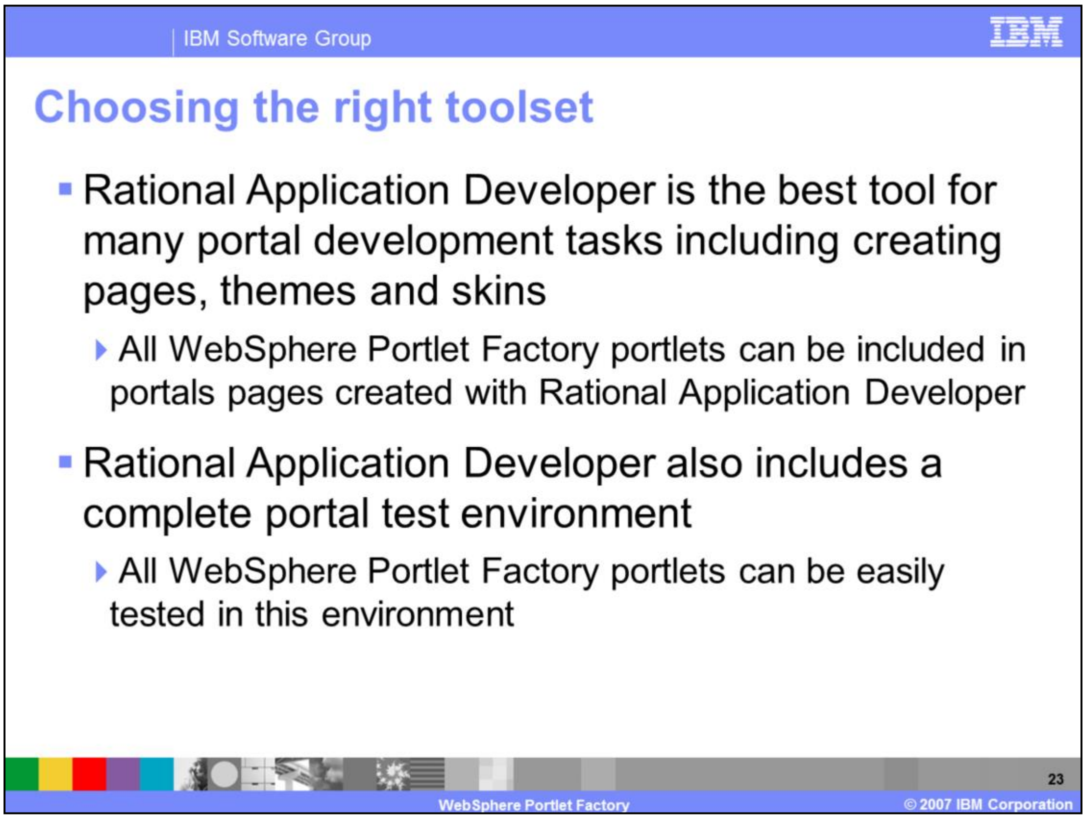
IBM

Agenda

- Key technology features
  - Builders / models
  - Automation
  - Profiles
  - Architecture
- WebSphere Portlet Factory and Rational Application Developer
- Summary

The Rational Application Developer and the WebSphere Portlet Factory are both Eclipse-based IBM development tools to create portlets.

This section will cover the difference between these two tools.


IBM Software Group

Choosing the right toolset

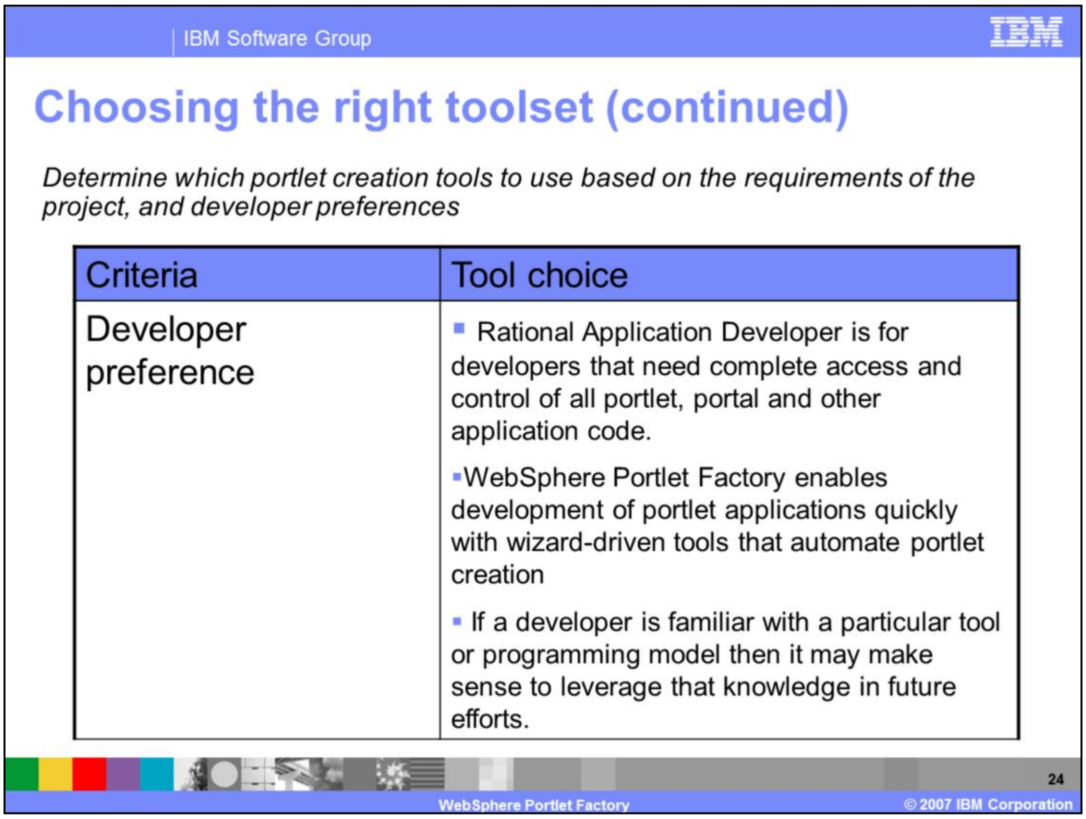
- Rational Application Developer is the best tool for many portal development tasks including creating pages, themes and skins
  - All WebSphere Portlet Factory portlets can be included in portals pages created with Rational Application Developer
- Rational Application Developer also includes a complete portal test environment
  - All WebSphere Portlet Factory portlets can be easily tested in this environment

23

WebSphere Portlet Factory © 2007 IBM Corporation

From a skills perspective, the tools are comparable. Neither requires you to have deep technical knowledge of portlet APIs or J2EE application development in order to be productive. Both enable you to integrate back-end systems such as SAP, Siebel, Domino, or PeopleSoft into portlets with no programming required. However, due to its highly-abstracted model-driven approach, the WebSphere Portlet Factory is less technical to use for portlet generation. Rational Application Developer provides a visual environment for portal and portlet development but by virtue of the fact that you are hand-building portlets and portal applications. Rational Application Developer requires some knowledge of portlet and portal application construction and typically requires writing some Java code when building more complex portlets.

A common scenario is to use Rational Application Developer and WebSphere Portlet Factory together to form comprehensive portal solutions. Portlets created in Rational Application Developer can be combined with portlets created with WebSphere Portlet Factory on portal pages that are created, tested, and deployed using Rational Application Developer. Since the tools plug into Rational Application Developer, the development environment is consistent when working in either tool.


IBM Software Group

Choosing the right toolset (continued)

Determine which portlet creation tools to use based on the requirements of the project, and developer preferences

Criteria	Tool choice
Developer preference	▪ Rational Application Developer is for developers that need complete access and control of all portlet, portal and other application code. ▪ WebSphere Portlet Factory enables development of portlet applications quickly with wizard-driven tools that automate portlet creation ▪ If a developer is familiar with a particular tool or programming model then it may make sense to leverage that knowledge in future efforts.

24

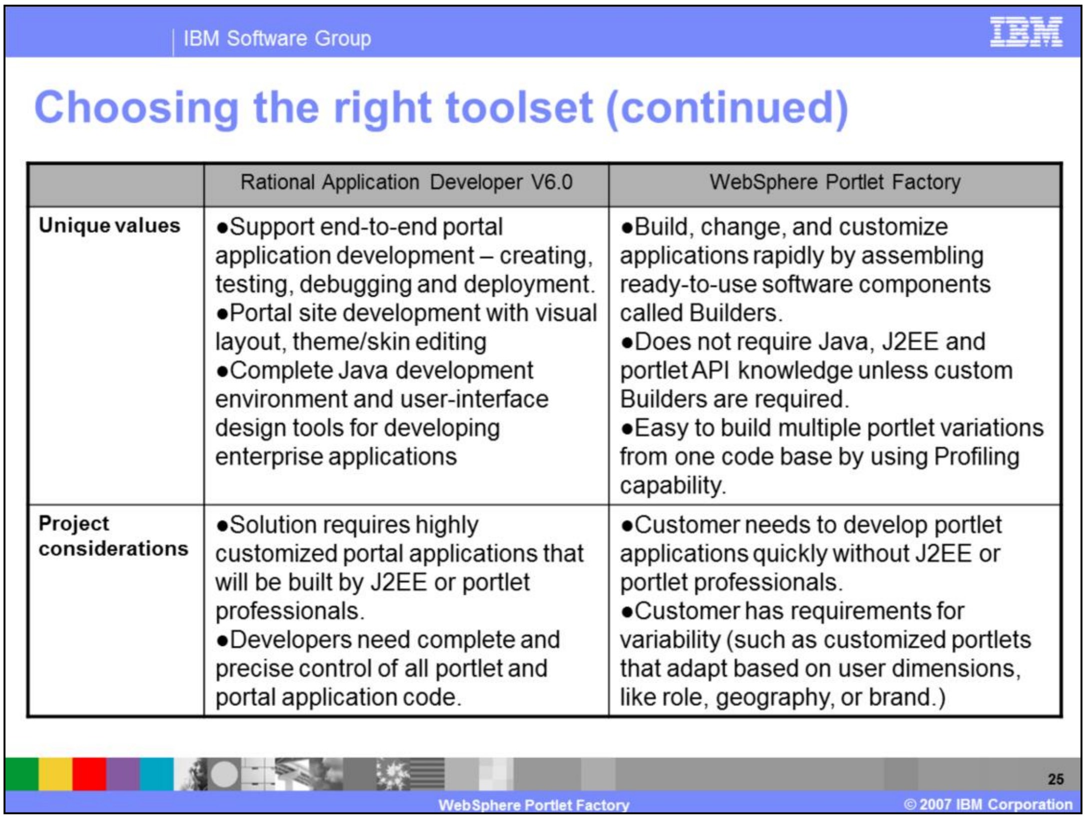
WebSphere Portlet Factory © 2007 IBM Corporation

To choose the right toolset for your needs, developer preference is one decision component.

The Rational Application Developer is for developers who need complete access and control of all portlet, portal and other application code.

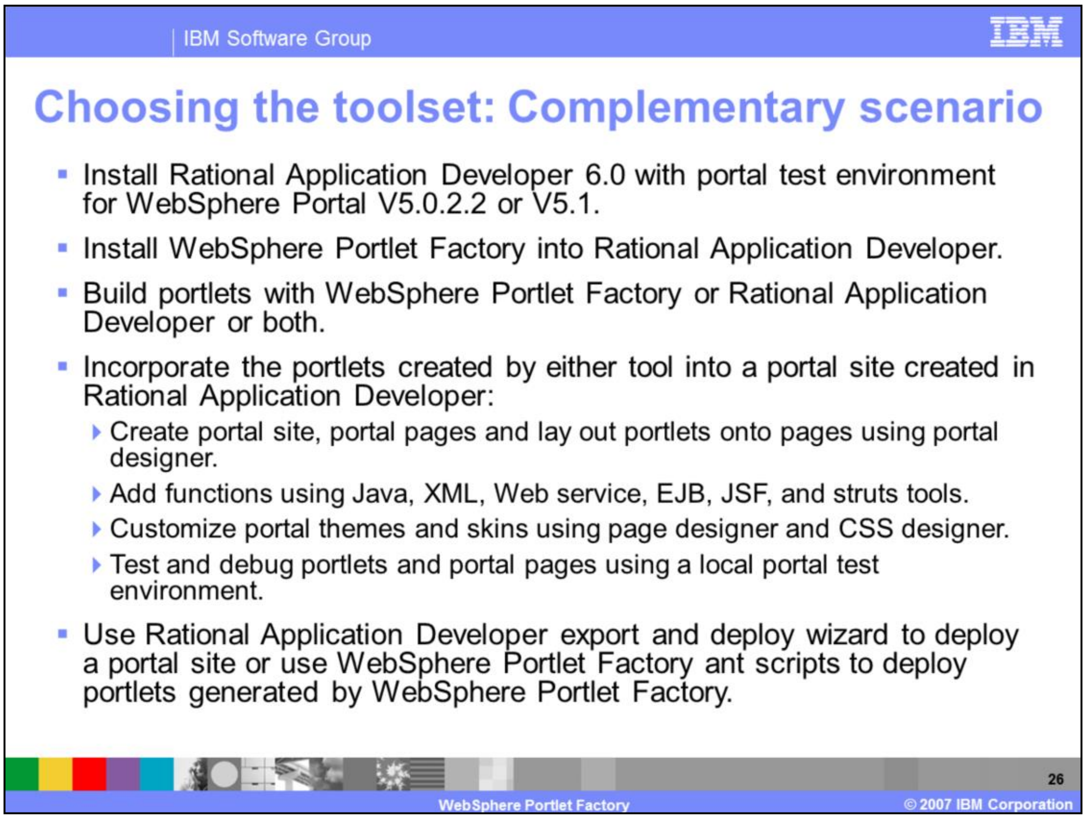
The WebSphere Portlet Factory enables development of portlet applications quickly with wizard-driven tools that automate portlet creation.

Of course the Rational Application Developer also provides several wizards to make the portlet development more easy, but the focus is different.



Choosing the right toolset (continued)

	Rational Application Developer V6.0	WebSphere Portlet Factory
Unique values	●Support end-to-end portal application development – creating, testing, debugging and deployment. ●Portal site development with visual layout, theme/skin editing ●Complete Java development environment and user-interface design tools for developing enterprise applications	●Build, change, and customize applications rapidly by assembling ready-to-use software components called Builders. ●Does not require Java, J2EE and portlet API knowledge unless custom Builders are required. ●Easy to build multiple portlet variations from one code base by using Profiling capability.
Project considerations	●Solution requires highly customized portal applications that will be built by J2EE or portlet professionals. ●Developers need complete and precise control of all portlet and portal application code.	●Customer needs to develop portlet applications quickly without J2EE or portlet professionals. ●Customer has requirements for variability (such as customized portlets that adapt based on user dimensions, like role, geography, or brand.)


IBM Software Group

Choosing the toolset: Complementary scenario

- Install Rational Application Developer 6.0 with portal test environment for WebSphere Portal V5.0.2.2 or V5.1.
- Install WebSphere Portlet Factory into Rational Application Developer.
- Build portlets with WebSphere Portlet Factory or Rational Application Developer or both.
- Incorporate the portlets created by either tool into a portal site created in Rational Application Developer:
  - Create portal site, portal pages and lay out portlets onto pages using portal designer.
  - Add functions using Java, XML, Web service, EJB, JSF, and struts tools.
  - Customize portal themes and skins using page designer and CSS designer.
  - Test and debug portlets and portal pages using a local portal test environment.
- Use Rational Application Developer export and deploy wizard to deploy a portal site or use WebSphere Portlet Factory ant scripts to deploy portlets generated by WebSphere Portlet Factory.

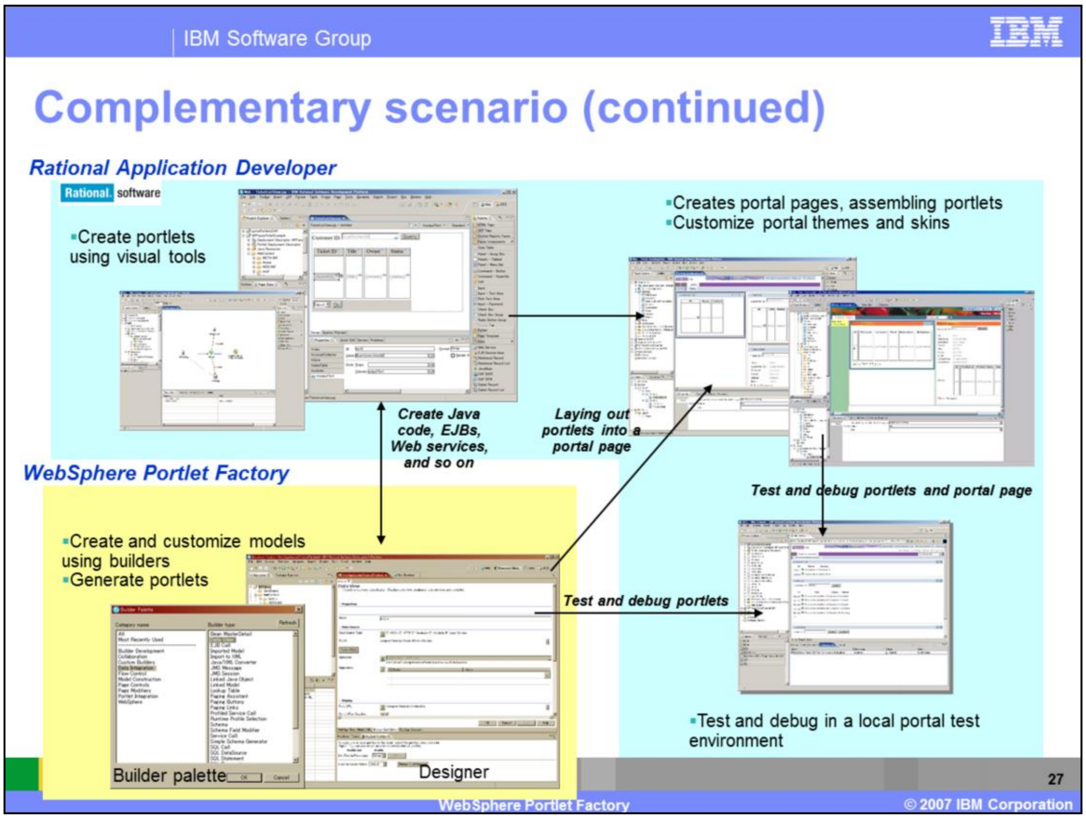
26

WebSphere Portlet Factory

© 2007 IBM Corporation

Remember that the portlet factory and the Rational Application Developer can be installed on the same Eclipse base.

Here you can see the steps necessary for this installation option.

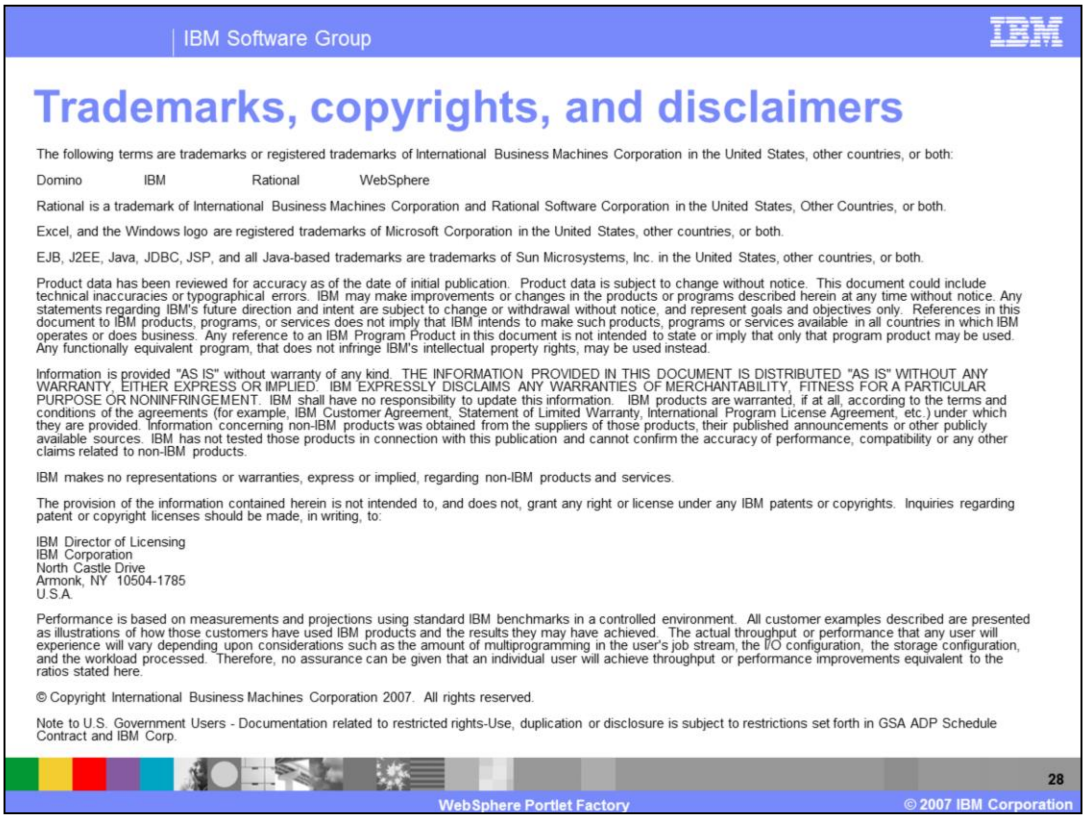

IBM Software Group

Complementary scenario (continued)

Rational Application Developer
Rational. software
Create portlets using visual tools
Creates portal pages, assembling portlets
Customize portal themes and skins
Create Java code, EJBs, Web services, and so on
Laying out portlets into a portal page
WebSphere Portlet Factory
Test and debug portlets and portal page
Create and customize models using builders
Generate portlets
Test and debug portlets
Builder palette
Designer
Test and debug in a local portal test environment
27
WebSphere Portlet Factory
© 2007 IBM Corporation

So the Rational Application Developer and the WebSphere Portlet Factory can be used in the same environment for their main competencies.

And not only the target system can be the same, the different portlets can also cooperate using the portlet-to-portlet communication standard.



Trademarks, copyrights, and disclaimers