IBM Software Group

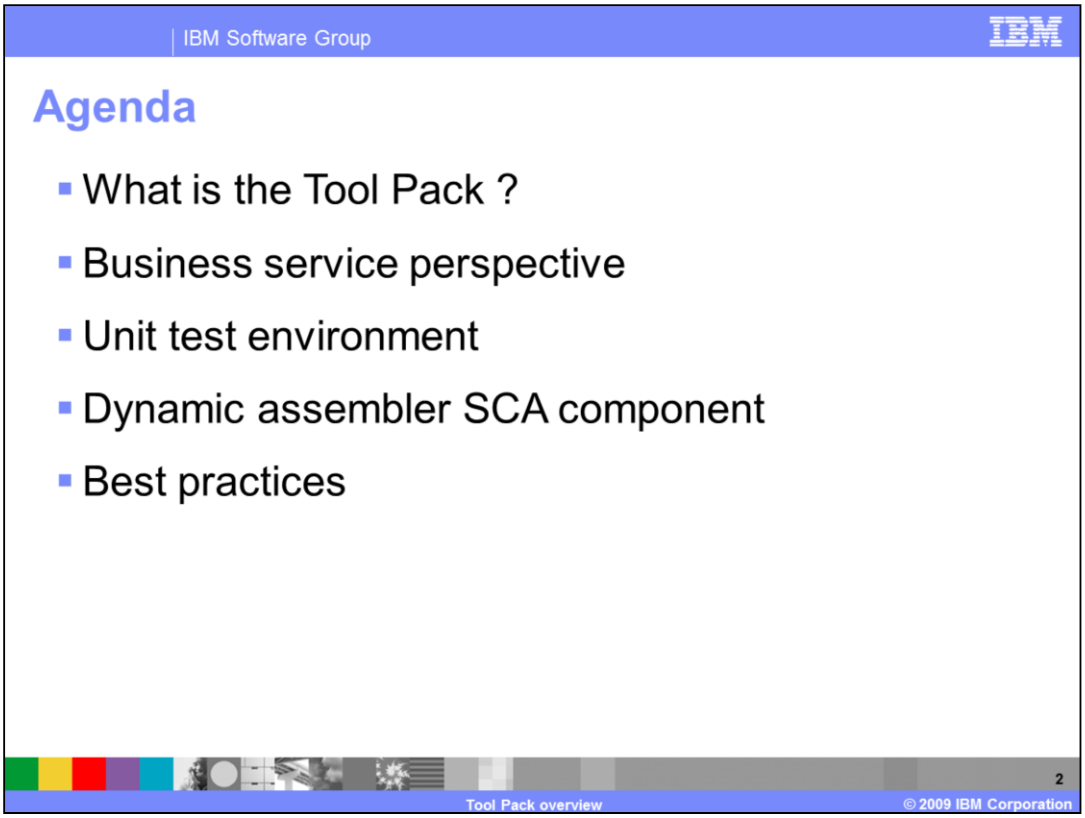
WebSphere Business Services Fabric

Tool Pack overview

@business on demand.

Converted to video June 29, 2015

This presentation provides an overview about WebSphere® Business Services Fabric Tool Pack.



IBM Software Group

IBM

Agenda

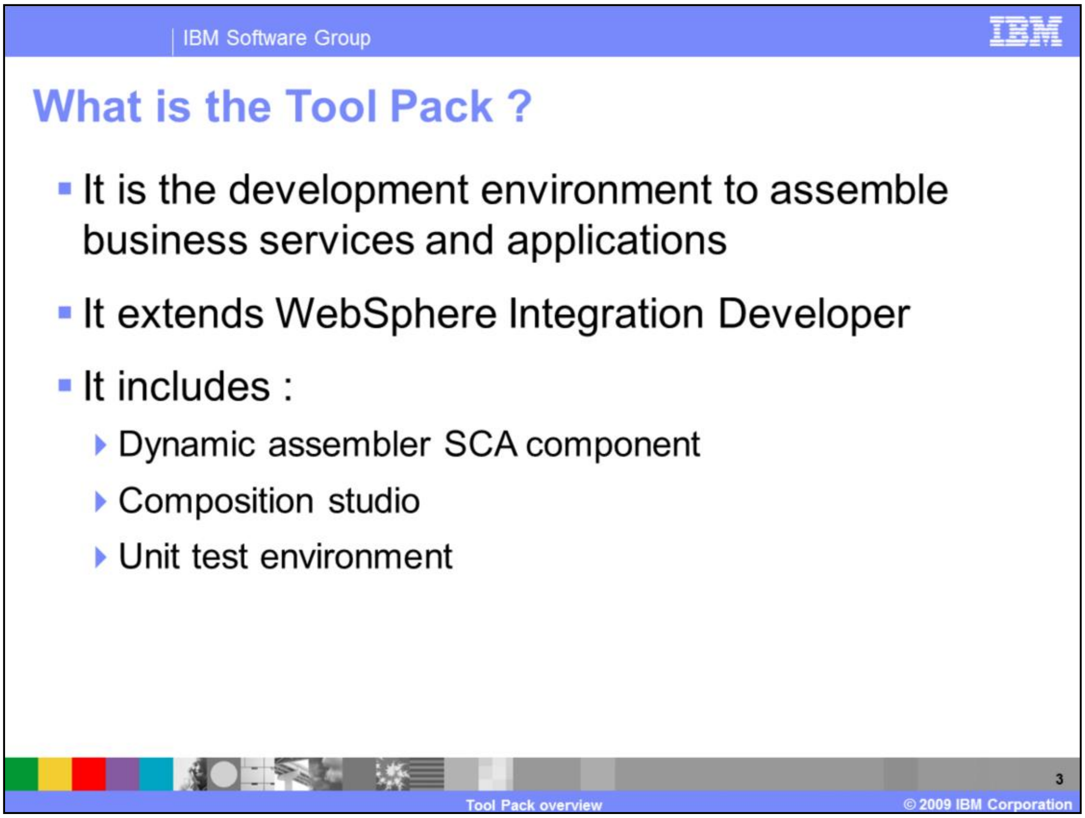
- What is the Tool Pack ?
- Business service perspective
- Unit test environment
- Dynamic assembler SCA component
- Best practices

2

Tool Pack overview

© 2009 IBM Corporation

This education module will cover details about what is a tool pack ? Then talk about the business service perspective, the unit test environment, the dynamic assembler and finally provide some best practices.



IBM Software Group

What is the Tool Pack ?

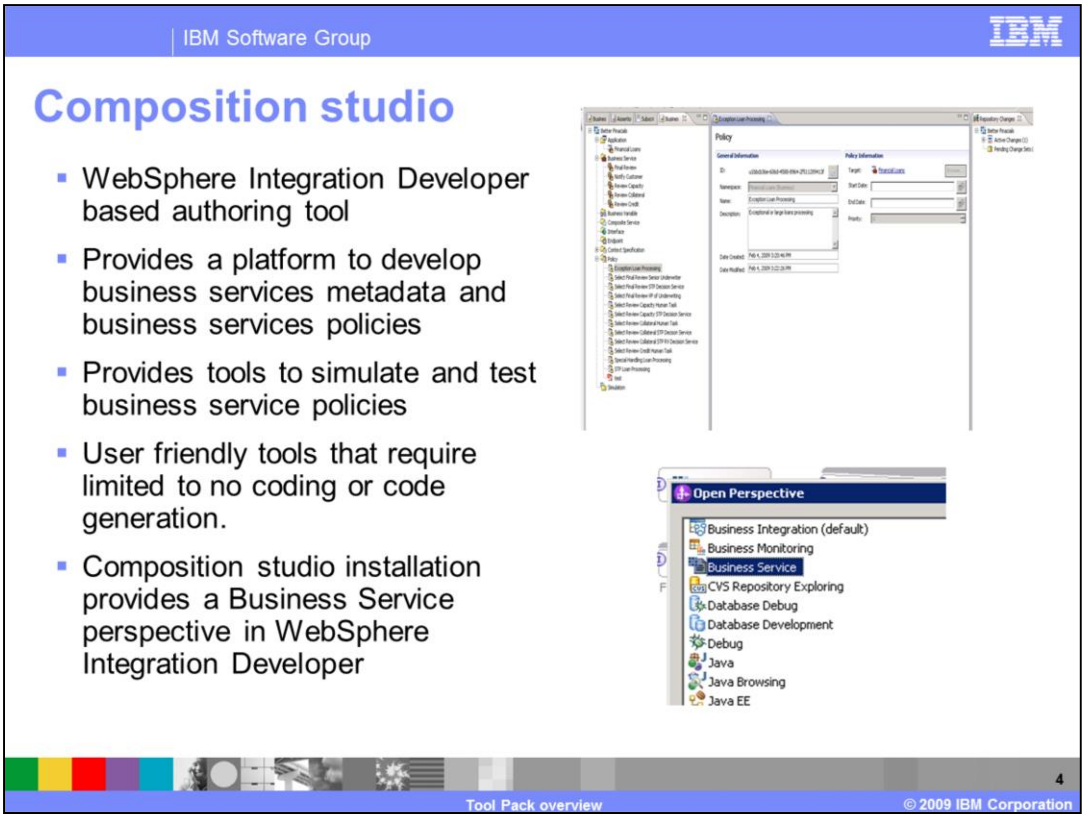
- It is the development environment to assemble business services and applications
- It extends WebSphere Integration Developer
- It includes :
  - Dynamic assembler SCA component
  - Composition studio
  - Unit test environment

3

Tool Pack overview

© 2009 IBM Corporation

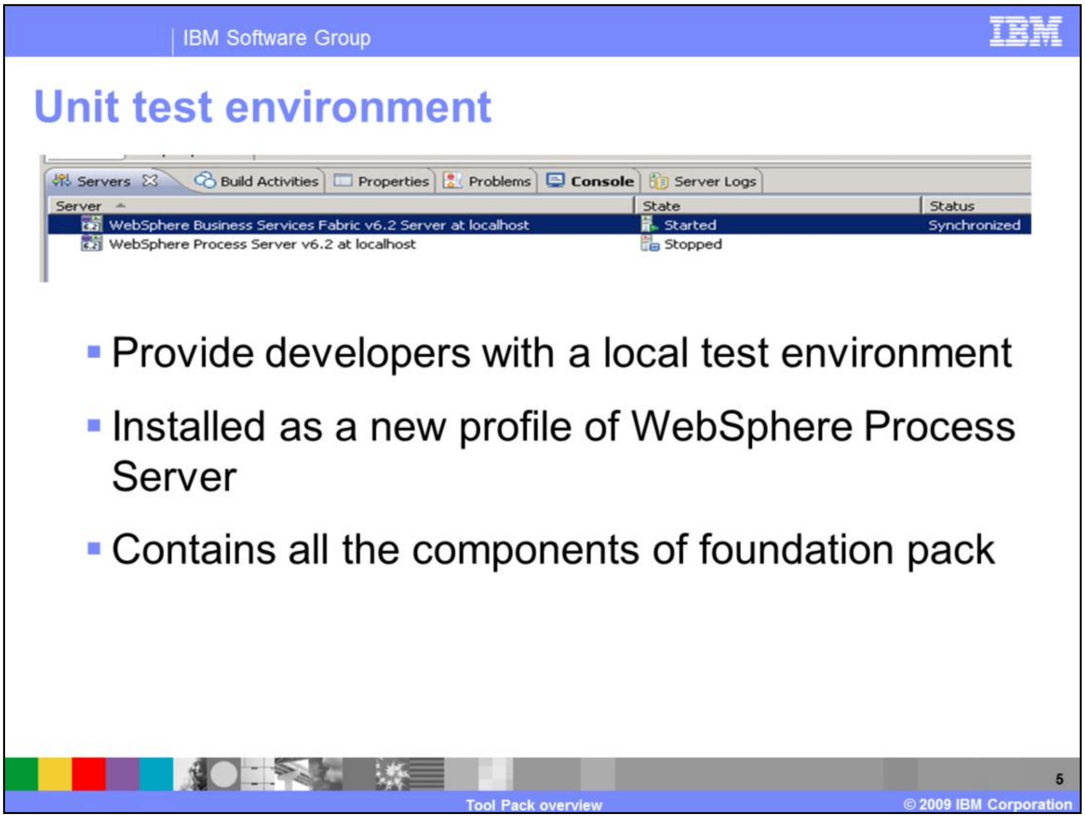
The WebSphere Business Services Tool Pack is the development environment for business services. The Tool Pack extends the WebSphere Integration Developer with the tools needed to create and manage business services metadata. Business services related tools are packaged as an eclipse plug-in that adds an additional perspective to WebSphere Integration Developer called Business Service. The Tool Pack provides developers with first the SCA capabilities for developers to develop dynamic business process Second the Composition Studio, that is a WebSphere Integration Developer plug-in. This enables you to assemble business services, create business policies, and simulate dynamic assembly of business services. The Composition Studio is packaged as an eclipse plug-in and adds an additional perspective to WebSphere Integration Developer called Business Services. And third the unit test environment that allows you to build, run and test business services in your local development environment. The Fabric unit test environment contains all the foundation pack component



Composition studio

- WebSphere Integration Developer based authoring tool
- Provides a platform to develop business services metadata and business services policies
- Provides tools to simulate and test business service policies
- User friendly tools that require limited to no coding or code generation.
- Composition studio installation provides a Business Service perspective in WebSphere Integration Developer

Now you will go through business services perspective, unit test environment and the dynamic assembler SCA component. First you will look into Composition Studio, which is a WebSphere Integration Developer-based authoring tool. Business analysts and integration developers use Composition Studio to model and publish business service models, interfaces, policies, and service portfolios into service oriented business processes and composite business applications. The tools provided by Composition Studio also provide the ability to simulate and test business policies. These tools are user friendly and require limited to no coding or code generation. Therefore they allow for simpler change management and do not require extensive coding skills to use. It is installed in WebSphere Integration Developer as a set of plug-in and add a new perspective called Business Services. Refer the composition studio overview for more details on this topic.



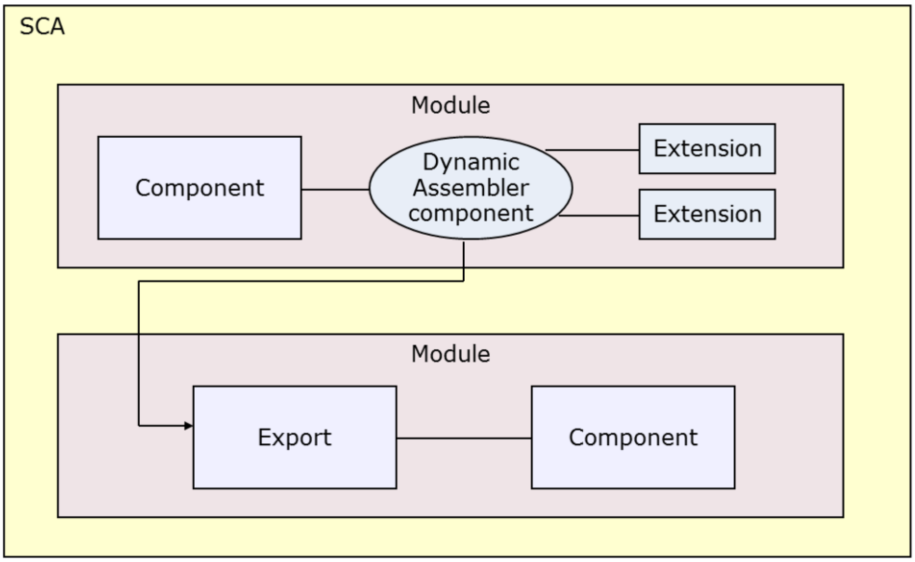
Unit test environment

Server	State	Status
WebSphere Business Services Fabric v6.2 Server at localhost	Started	Synchronized
WebSphere Process Server v6.2 at localhost	Stopped	

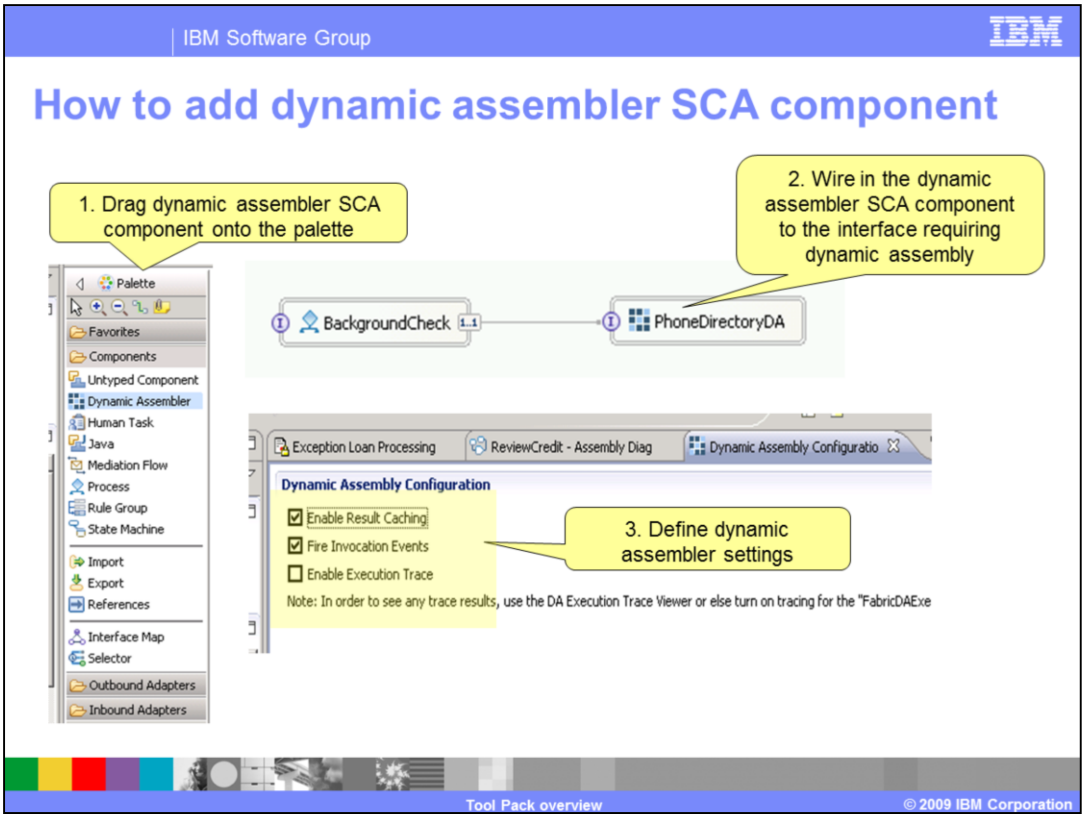
- Provide developers with a local test environment
- Installed as a new profile of WebSphere Process Server
- Contains all the components of foundation pack

Next you will look into the WebSphere Business Services Fabric Unit Test Environment. This is an extension of the embedded WebSphere Process Server environment in WebSphere Integration Developer. The Tool Pack installer create a Fabric profile on the embedded WebSphere Process Server and installs all the Foundation Pack components on this profile. A Fabric developer can then use this as test environment to publish and run business services and applications.

Development model



In order to understand this SCA component take a look at the development model. The dynamic assembler SCA component can be added to any other SCA component or BPEL. The interface associated with the dynamic assembler SCA component identifies which interface service providers are selected at runtime. The developer or architect can get better visibility and control over the working of the dynamic assembler by using extensions. These extensions are covered further in this presentation.



IBM Software Group

How to add dynamic assembler SCA component

1. Drag dynamic assembler SCA component onto the palette

2. Wire in the dynamic assembler SCA component to the interface requiring dynamic assembly

Dynamic Assembly Configuration

☑ Enable Result Caching

☑ Fire Invocation Events

☐ Enable Execution Trace

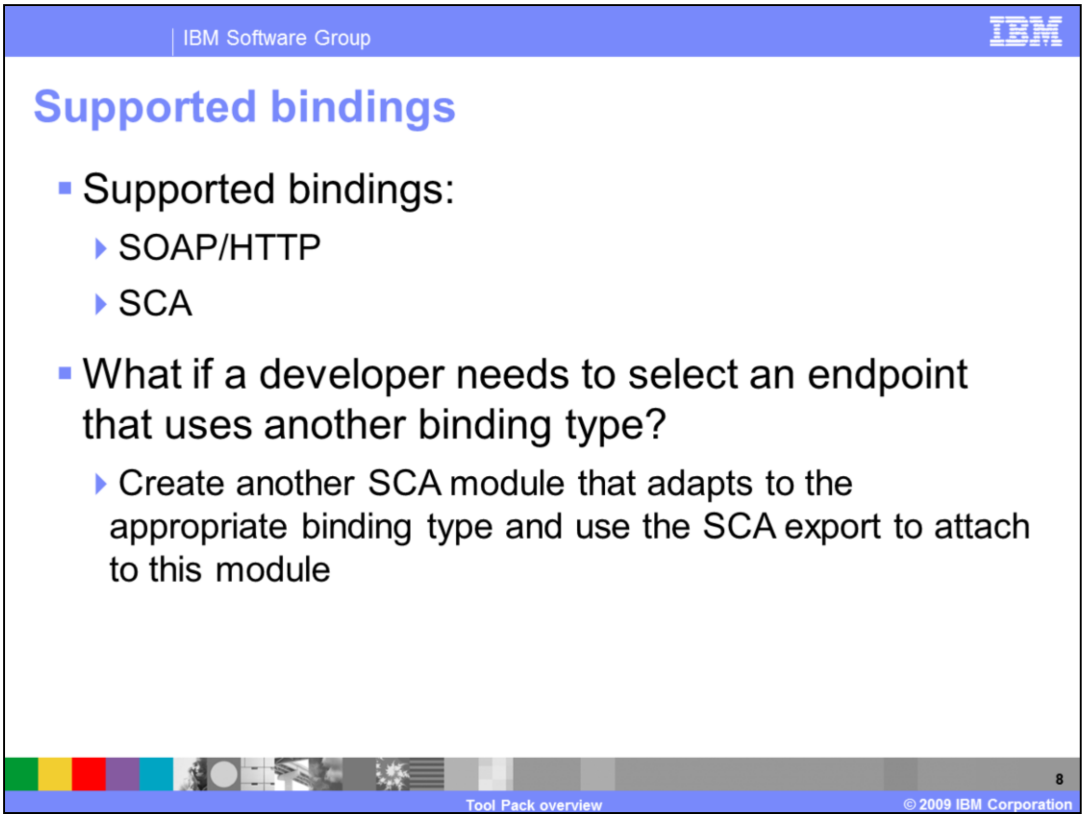
Note: In order to see any trace results, use the DA Execution Trace Viewer or else turn on tracing for the "FabricDAExe

3. Define dynamic assembler settings

Tool Pack overview

© 2009 IBM Corporation

In order to add a dynamic assembler SCA component in an assembly diagram first switch to Business Integration perspective if you are not already there. Then is the assembly diagram window from the palette select dynamic assembler SCA component and drag onto the work area. Next wire the dynamic assembler SCA component to the interface requiring dynamic assembly. In the end define the dynamic assembler settings. These settings are enable result caching, fire invocation events and enable execution trace. Enable Result Caching controls whether Dynamic Assembler component is required to recalculate the selected endpoint each time. Fire Invocation Events controls whether the Dynamic Assembler component emits events summarizing each invocation. And enable execution trace turns on execution tracing for an individual Dynamic Assembler or all Dynamic Assemblers. Then each time a Dynamic Assembler request is handled, activity information is shown in a Web application or written to log files.



IBM Software Group

Supported bindings

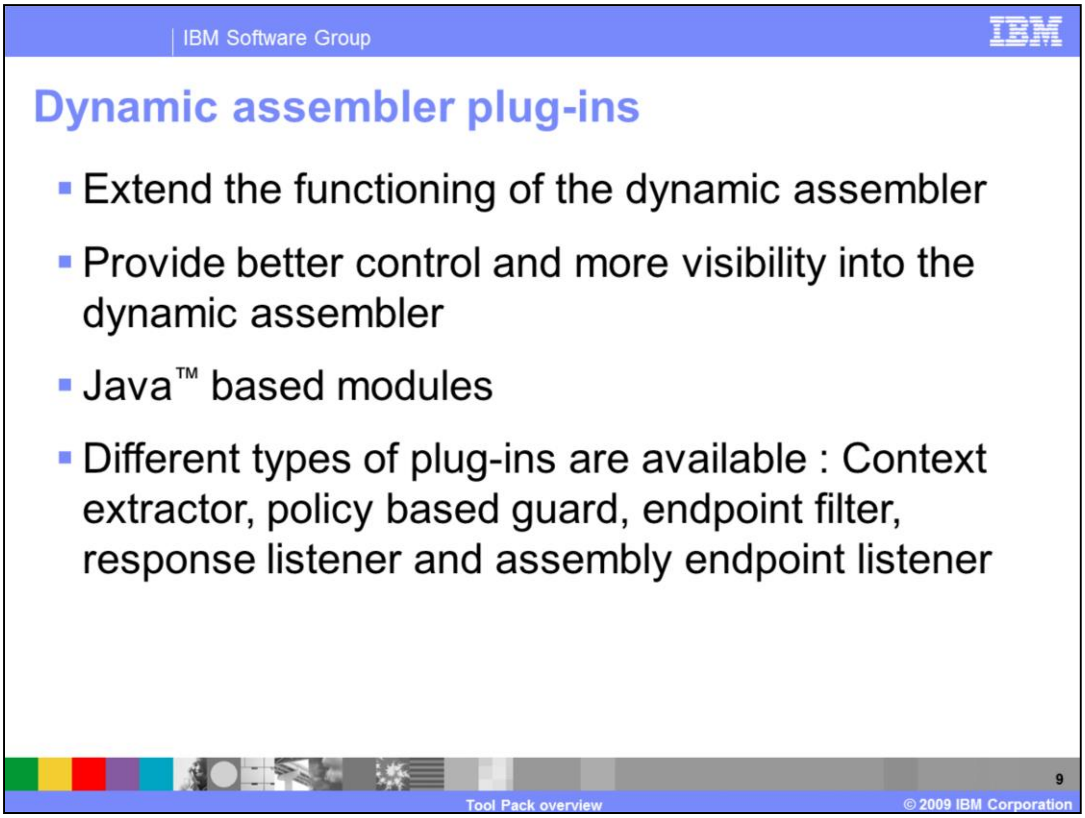
- Supported bindings:
  - SOAP/HTTP
  - SCA
- What if a developer needs to select an endpoint that uses another binding type?
  - Create another SCA module that adapts to the appropriate binding type and use the SCA export to attach to this module

8

Tool Pack overview

© 2009 IBM Corporation

The dynamic assembler SCA component supports SOAP/HTTP and SCA bindings. What does that mean? An endpoint selected by the Dynamic Assembler must be either accessible as a Web service using SOAP/HTTP or accessible as an SCA export. Therefore if a end point that uses another binding type needs to be included, then create another SCA module and use the SCA export to attach to this module.



IBM Software Group

Dynamic assembler plug-ins

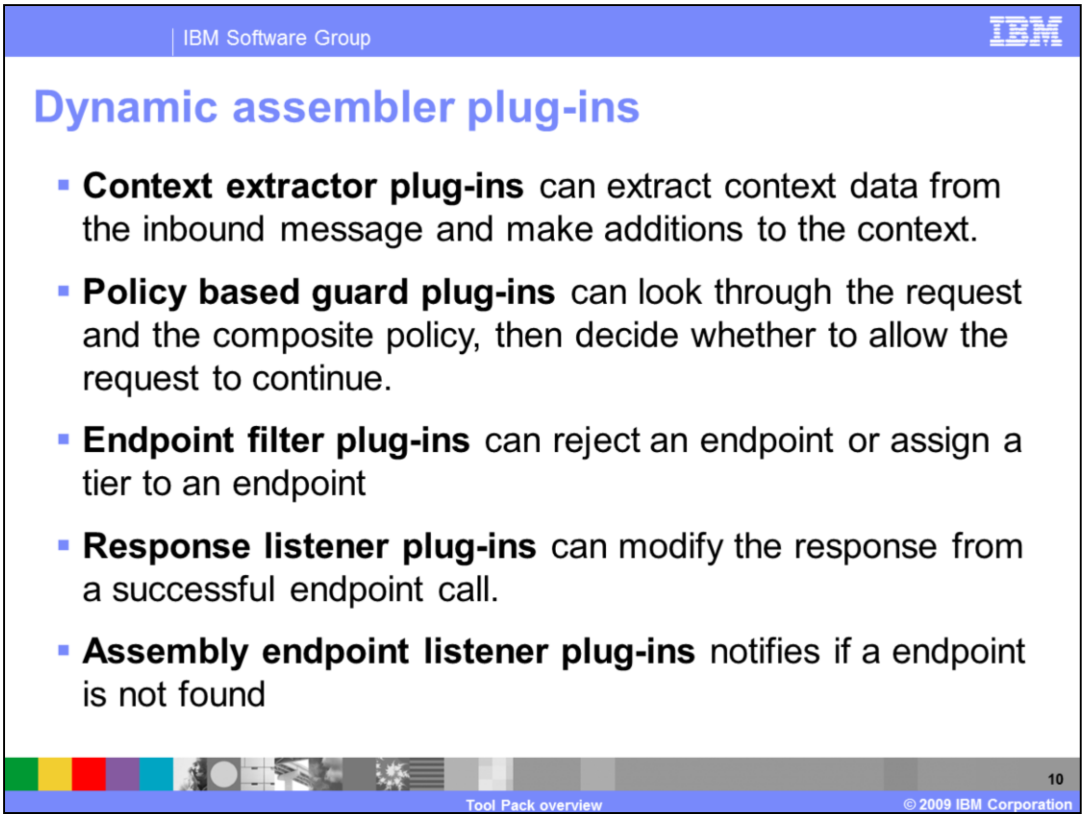
- Extend the functioning of the dynamic assembler
- Provide better control and more visibility into the dynamic assembler
- Java™ based modules
- Different types of plug-ins are available : Context extractor, policy based guard, endpoint filter, response listener and assembly endpoint listener

9

Tool Pack overview

© 2009 IBM Corporation

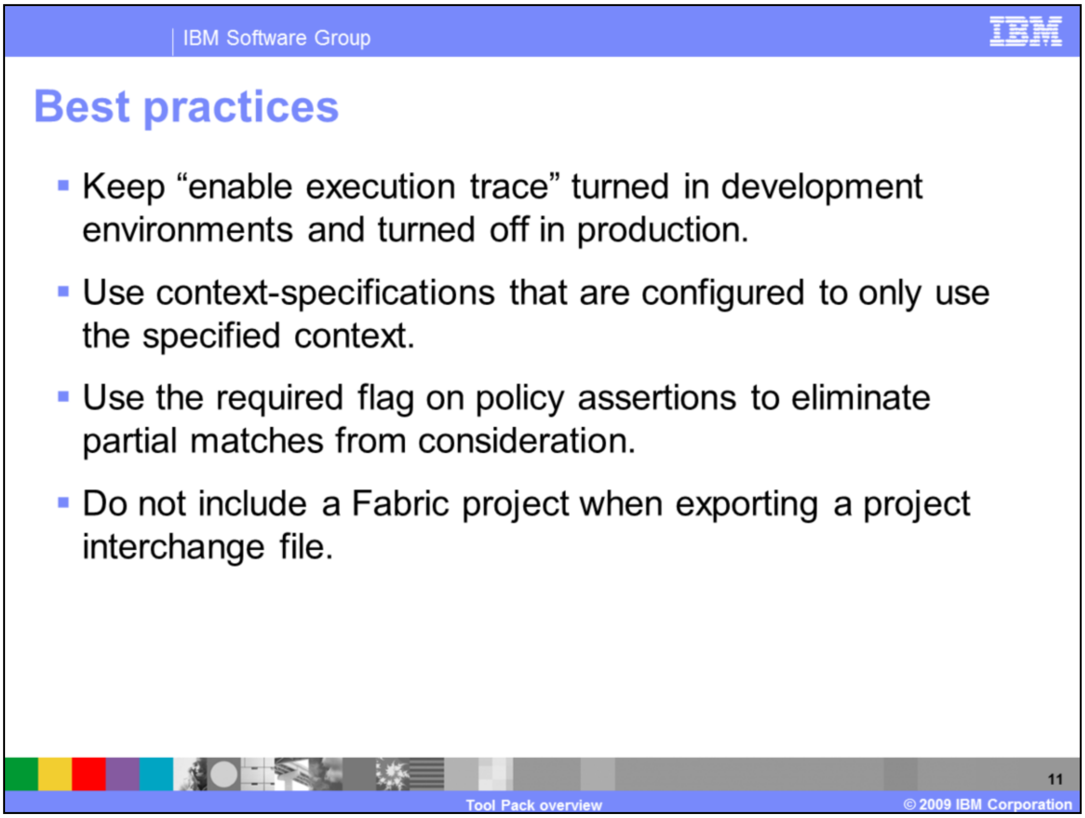
Dynamic Assembler plug-ins are Java based modules that when implemented help extend, control and provide more visibility into the dynamic assembler. Each plug-in type is called with defined parameters under defined conditions. Defining plug-ins requires writing Java code. Some plug-ins are only informational, meaning they are called when certain actions have taken place. Plug-ins that can affect a request or a response are called operational plug-ins. Available plug-ins types include Context Extractor, Policy Based Guard, Endpoint Filter, Response Listener and Assembly endpoint Listener.



Dynamic assembler plug-ins

- **Context extractor plug-ins** can extract context data from the inbound message and make additions to the context.
- **Policy based guard plug-ins** can look through the request and the composite policy, then decide whether to allow the request to continue.
- **Endpoint filter plug-ins** can reject an endpoint or assign a tier to an endpoint
- **Response listener plug-ins** can modify the response from a successful endpoint call.
- **Assembly endpoint listener plug-ins** notifies if a endpoint is not found

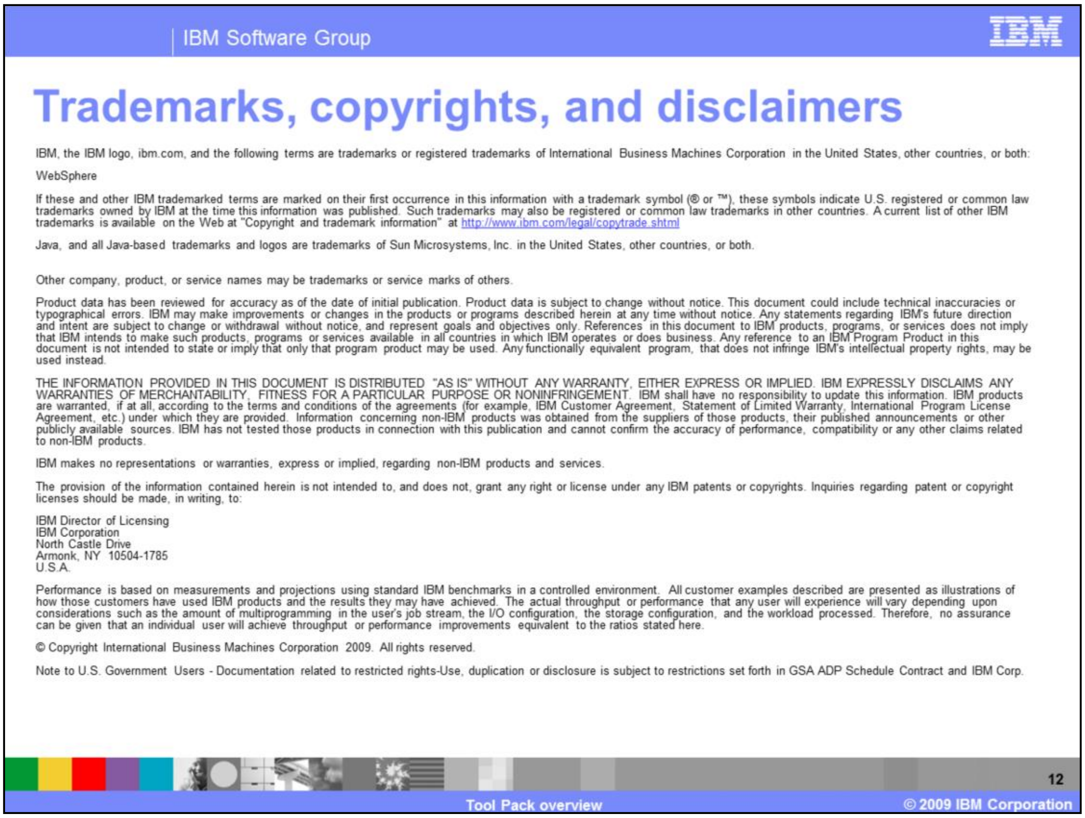
The slide walks through and defines the different types of plug-ins. The Context Extractor is called early in Dynamic Assembler processing, to extract context data from the inbound message and make additions to the context. The Policy Based Guard Plug-ins can look through the request and the composite policy, then decide whether to allow the request to continue. It can cause a request to fail without actually invoking an endpoint. The Endpoint Filter Plug-ins can reject an endpoint or assign a tier to an endpoint after the Dynamic Assembler has performed has given a policy matching score to all the endpoints, but before an endpoint is selected. It is invoked with the entire list of all candidates that have been considered. The Response Listener Plug-ins can modify the response from a successful endpoint call. Assembly Event Listener are informational plug-ins that are invoked after all Response Listeners have been run or, if an endpoint can not be found, so that notifications can be sent to other frameworks.



Best practices

- Keep “enable execution trace” turned in development environments and turned off in production.
- Use context-specifications that are configured to only use the specified context.
- Use the required flag on policy assertions to eliminate partial matches from consideration.
- Do not include a Fabric project when exporting a project interchange file.

In the end here are some best practices that will help in your development work. First Keep “enable execution trace” turned on in development environments and turned off in production. This feature provides valuable information for debugging at the cost of performance. Second, on the “Dimensions” tab in the context specification editor, there is an option called "Only use specified context at runtime”. When this option is checked, only those dimensions that have been specified as required or optional in the context specification are used for decision-making. When checked you know that the context-specification tells the full story of what is provided at runtime. When unchecked, new context properties can affect how policies are composed. It is a best practice to use context-specifications that are configured to only use the specified context. Third, use the required flag on policy assertions to eliminate partial matches from consideration. Finally, do not include a Fabric Project when exporting a project interchange.



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.