



IBM Software Group

## IBM WebSphere Application Server Feature Pack for Communications Enabled Applications

### *Feature pack overview*



@business on demand.

© 2009 IBM Corporation  
Updated July 30, 2009

This presentation provides an overview of the new function provided by the IBM WebSphere® Application Server Feature Pack for Communications Enabled Applications.



## Section

# *Overview and architecture*



This section provides an overview of the goals and architecture of the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications.

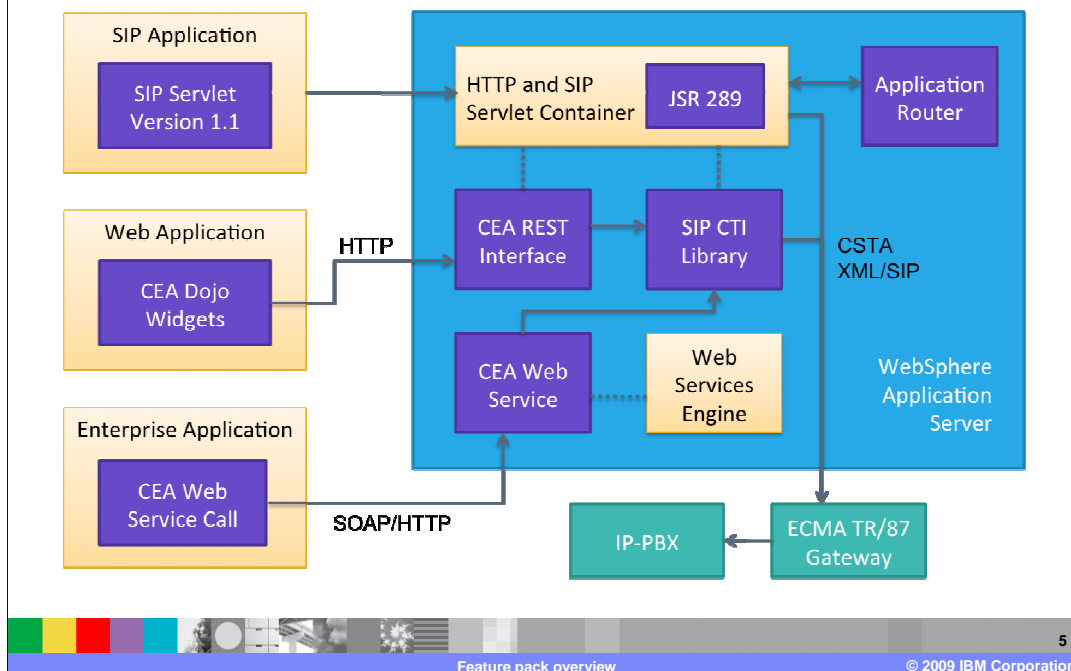
## Feature pack overview

- Enable any application to quickly add communications support
- Reuse Java™-based skills by hiding communications' complexity through Web services and REST interfaces (widgets)
- Simplify making and monitoring calls from browser-based applications
- Session data sharing through co-browsing
- Integrate with current communications infrastructure
- Support for the JSR 289 specification



The goal of this feature pack is to provide a fast path to integrate communications support into your Web applications and enterprise applications. Typically, the overhead for incorporating communications and telephony services into an application requires in-depth knowledge of communications technology and the session initiation protocol. This feature pack simplifies the development process by allowing any developer with Java skills to use standard interfaces – like JAX-WS Web services or widgets created using the Dojo toolkit – to embed communication services directly into an application. These interfaces simplify making and monitoring calls from browser-based applications, and support new types of communication, like sharing session data through a co-browsing link that allows two participants to see the same content in their browsers. These communication services can be integrated with your existing applications and your current communications infrastructure. In addition to the new interfaces for communication services, the feature pack also includes support for the JSR 289 specification, SIP servlet 1.1.

## Architecture



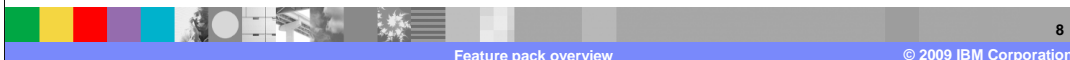
This diagram provides an overview of the new features in the IBM WebSphere Application Server Feature Pack for Communications Enabled Applications. The purple boxes indicate new functionality in this feature pack. Typically, there are two types of applications that you might run on top of this feature pack – SIP-based communications applications that directly implement communications functions and business applications that access communications function. In the diagram shown here, both the enterprise application and the Web application are considered types of business applications. Business applications communicate outside of the application server through the SIP computer telephony integration (CTI) layer, accessing an IP-PBX through a standard ECMA TR/87 gateway. In order to use your IP-PBX with the feature pack, it must support the ECMA TR/87 standard protocol or use an ECMA TR/87 compliant gateway.





## JSR 289 implementation

- Support the JSR 289 SIP servlet 1.1 specification
  - ▶ Application routing
  - ▶ Annotations
  - ▶ Converged applications
  - ▶ Back-to-back user agent helper class
- Maintain compatibility with JSR 116
- Enable application routing across a WebSphere Application Server cluster
- Asynchronous work dispatching across a cluster
  - ▶ IBM proprietary API



The Feature Pack for CEA provides a fully compliant implementation of the JSR 289 specification for SIP servlet 1.1. The SIP servlet specification provides the Java API standards for Session Initiation Protocol (SIP) applications. JSR 289 is an update to the existing SIP servlet specification that addresses new requirements determined by industry users, while maintaining compatibility with the previous SIP servlet specification, defined in JSR 116. The SIP container has the ability to distinguish between JSR 289 and JSR 116 applications and act according to the rules in each specification. Also, JSR 289 and JSR 116 applications can co-exist in the same SIP container.

Application routing enables developers to build complex services out of smaller applications. On initial requests the container calls the application router to determine which application to invoke based on the type of request. The application router is the central hub for selecting application order. Annotations provide a fast way to develop applications by embedding metadata directly in application source code. Annotations simplify development and application packaging and have been used in many other Java-based programming models, like EJB 3.0 and Java EE 5. This specification update includes specialized annotations to create SIP constructs, like `@SipServlet` and `@SipApplication`. A converged application contains SIP servlet components and other Java EE components, like HTTP servlets and enterprise beans. The JSR 289 specification provides a new, standardized mechanism for building converged applications. JSR 289 also simplifies the SIP back-to-back user agent pattern, a common pattern in SIP applications, by providing a back-to-back user agent helper class.



## JSR 289 implementation

- Support the JSR 289 SIP servlet 1.1 specification
  - ▶ Application routing
  - ▶ Annotations
  - ▶ Converged applications
  - ▶ Back-to-back user agent helper class
- Maintain compatibility with JSR 116
- Enable application routing across a WebSphere Application Server cluster
- Asynchronous work dispatching across a cluster
  - ▶ IBM proprietary API



In addition to supporting JSR 289, the feature pack includes a new API for asynchronously dispatching pieces of work to specific `SipApplicationSessions` in a thread safe way. This API is not a part of JSR 289 and is only supported by IBM with the Feature Pack for CEA. When you use this new API, your applications can dispatch tasks to specific sessions in a single server or across a WebSphere Application Server cluster.

Only the asynchronous invocation API and the JSR 289 implementation support clustering in the Feature Pack for CEA. Other new features described in this overview – like the REST APIs and CEA widgets – are not yet supported in a cluster.

## Telephony access services

- New programming model that requires no SIP knowledge to access telephony services
- Make and disconnect a call
- Receive incoming call notifications
- Access communication services using:
  - ▶ REST interface
  - ▶ JAX-WS Web service
- Ability to proxy to external Web service implementations
- Base services using CSTA/SIP (TR/87)



With the Feature Pack for CEA, enterprise developers do not need to have extensive knowledge of telephony or Session Initiation Protocol (SIP) to implement communications enabled applications. The CEA capability delivers call control, notifications, and interactivity and provides the platform for more complex communications. Telephony services allow you to access a unified communications environment from within business applications to increase the efficiency of processes, reduce communications errors, and optimize business interactions in real time. CEA is based on SIP-enabled services that use Representation State Transfer (REST) servlets and Web services in a converged HTTP and SIP application.

The Feature Pack for CEA includes a library of Dojo-style widgets for use in Web applications. Dojo widgets are prepackaged components of JavaScript and HTML code that add interactive features that work across platforms and browsers. CEA widgets are extensible, allowing developers to customize them to handle more advanced tasks. These widgets provide capabilities like making and disconnecting calls and receiving incoming call notifications. Telephony access services are designed to integrate with existing communications infrastructure by communicating using standard protocols, like ECMA TR/87.

## Multi-modal Web collaboration services

- Support for linking two Web sessions using widgets that push data across the link
- Collaboration through co-browsing, synchronized forms
  - ▶ **Example:** Co-browse enables the controlling user to highlight content in a page, and the remote user can see the highlighting
  - ▶ **Example:** Form sharing enables a customer to watch while a customer support representative fills out information about them in a form



In addition to telephony access services, the Feature Pack for CEA provides session data sharing for multi-modal Web collaboration services. Multi-modal Web interaction allows you to provide session linking, or shared sessions, between users browsing the same Web site from different locations. With session linking, users can interact dynamically in collaborative ways, such as co-browsing or co-shopping Web sessions, while maintaining their individual unique sessions within the server. Commerce Web sites can use this service to provide product or customer support, while protecting information on the internal site. For example, commerce sites can use the collaborative shopping experience to attract more customers to their sites. With a combination of click-to-call functionality and multi-modal interaction, you can support two-way synchronized text forms between a customer and a customer service representative. Two-way forms can be used to allow a customer and a customer service representative to fill out an online form together at the same time. This allows the customer to do things like verify name spelling and contact information and to be able to take over control of the form and provide sensitive information – like an identification number or credit card information – in a masked field that the customer service representative is not able to see.

## Sample applications

- IP-PBX application included
  - ▶ Use for unit testing
- Plants by WebSphere is a Java EE sample application provided with the application server
  - ▶ Current version is Dojo enabled without frames, provided with WebSphere Application Server V6.1 and V7, and with the Web 2.0 Feature Pack
  - ▶ Includes widgets for telephony and collaboration services
- Web services sample
  - ▶ Sample application showing how to use the Web services interfaces to access communication services



The IBM WebSphere Application Server Feature Pack for Communications Enabled Applications comes with sample applications to help you get started using new features quickly. Some type of private branch exchange, or PBX, is needed in your testing environment to allow calls to connect. In a production environment, most users will use specialized PBX hardware for this purpose. To make it simpler to set up a testing environment, the feature pack comes with a sample IP-PBX application that you can deploy to your application server and use in your testing. This PBX application is a sample and should not be used in production.

The feature pack also provides two sample applications that illustrate how to build and use some of the new CEA components. The Plants By WebSphere application is a standard Java EE sample application that comes with WebSphere Application Server. The version provided with the feature pack has been enhanced to include widgets for telephony and collaboration. The application includes full source code so that you can see examples of how to embed CEA widgets. The Web services sample is a simple, stand-alone sample that illustrates how to access communications services using the new Web services interface provided with the feature pack. This sample also provides full source code to help you in understanding the CEA Web services programming model.

## Sample packaging

Sample application	Packaging location
IP-PBX	<WAS_HOME>/feature_packs/cea/samples/commsvc.pbx.ear
PlantsByWebSphere	<WAS_HOME>/feature_packs/cea/samples/plantsbywebsphere/PlantsByWebSphere.ear
Web services sample	<WAS_HOME>/feature_packs/cea/samples/webservice.sample/commsvc.ws.sample.ear



All of the sample applications that come with the IBM WebSphere Application Server Feature pack for Communications Enabled Applications are packaged in the <WAS\_HOME>/feature\_packs/cea/samples directory. The exact file names for each of the samples are shown here for your reference. The directories that hold the Plants By WebSphere sample and the Web services sample each include a README file that explains how to deploy, configure, and get started using the sample.

## Section

# *Summary*

This section contains a summary of this presentation.

## Summary

- The WebSphere Application Server Feature Pack for Communications Enabled Applications makes it possible to quickly add communications support to new or existing applications, including:
  - ▶ Widgets and Web services interfaces to embed communication services
  - ▶ Multi-modal Web collaboration
  - ▶ JSR 289 support, with flexible application routing configuration based on the characteristics of SIP requests
  - ▶ Asynchronous invocation API to allow task dispatching across a cluster
  - ▶ Sample applications to assist in learning the programming models



The Feature Pack for CEA lets you quickly add communications support to new or existing applications or business processes. The feature pack supports telephony services – like making calls and receiving call notifications – and multi-modal Web collaboration – like sharing a view of a Web page or collaboratively filling out an online form. These communications services are exposed through a REST interface, which can be accessed through CEA widgets, and through a Web services interface. You do not need to have in-depth knowledge of communications infrastructure or the session initiation protocol to be able to integrate these communications services into your applications. The feature pack also supports the SIP servlet 1.1 specification, JSR 289, which includes new features like flexible application routing configuration, standardized support for converged applications, and a new annotation-based programming model to simplify development and application packaging. The asynchronous invocation API is another new feature in the product that provides a thread safe mechanism for dispatching pieces of work to servers in a cluster, based on a target session ID. Sample applications are provided with the feature pack to simplify testing and assist in learning the new CEA programming models.

## Feedback

### Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_CEAFP\\_Overview.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_CEAFP_Overview.ppt)

This module is also available in PDF format at: [../CEAFP\\_Overview.pdf](..../CEAFP_Overview.pdf)



You can help improve the quality of IBM Education Assistant content by providing feedback.



## 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>

EJB, Java, JavaScript, 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.

© Copyright International Business Machines Corporation 2009. All rights reserved.

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.

