



IBM Software Group

**WebSphere® Process Server V6.0**  
**WebSphere Integration Developer V6.0**  
**WebSphere Adapters V6.0**

***eXchange Scenario - Business Description  
and Lab Perspective***



@business on demand.

© 2005 IBM Corporation  
Updated December 16, 2005

This presentation will introduce the eXchange scenario which is the basis for several of the lab exercises provided with this education.

## Goals

- Introduce the eXchange scenario
  - ▶ ClipsAndTacks.com – fictional office supply company
- Describe Requirements and Process Flow
- Understand how eXchange is modified for labs
- Understand flow from Service Component Architecture (SCA) Module perspective
- Introduce SCA Module wiring diagrams
  - ▶ Components and Imports/Exports for each module

There are two main goals of this presentation, the first being the introduction of the eXchange scenario from a business perspective and the second describing the set of lab exercises which are based on a modified version of this scenario. To introduce the business perspective you will take a look at a fictional office supply company called ClipsAndTacks.com. The company's requirements and the process flow needed to address those requirements are described. Using that as a base, modifications to the process flow are described. These modifications are needed to make the scenario practical for use with lab exercises. You will then dig in a little deeper and look at the overall flow in terms of Service Component Architecture (SCA) modules and their relationship to each other. Finally, you will look inside each of the SCA modules to see the wiring diagram for the module including the SCA components, Imports and Exports.

## Section

# *eXchange – Requirements and Process Flow*

You now begin by looking at the eXchange scenario requirements and process flow from the business perspective of ClipsAndTacks.com.

## ClipsAndTacks.com - Requirements

- They want to **participate in a global repository** to ensure that potential B2B customers can find their products. The repository is accessible via web services and has standardized schemas for product descriptions.
- As products are described in the LOB systems, **corporate** wants to know the product updates are coming and have the **information available** to the corporate sales team.
- Since the sales team doesn't need all of the information and some of the jargon and code words will be confusing to them, the content is automatically **cleansed** as it is transformed from one format to the other.
- They want to **automate** their business processes.
- They want to gain greater flexibility and agility as they can easily react on **modifications due to changed business requirements or conditions**.
- They plan to offer customers (both individual and corporate) a service to navigate the product catalog and **order office supply from the company portal server** rather than through their telephone center. A discount rate, decided by the corporate business analysts, will be applied to the order according to the sales period. Once an order is fulfilled, the shipping companies will get notified and ship the order to the customers based on business policies.

ClipsAndTacks.com is a fictional office supply company. It is organized into several different lines of business which produce products for sale. Information about these products needs to be merged into a company wide sales and delivery mechanism. They sell both wholesale and retail and have several different channels for selling, such as B2B interactions, a corporate sales organization, a call center and website.

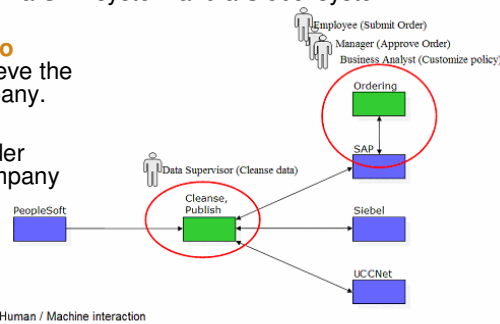
The following are the requirements their system must address. First, they want to participate in B2B transactions which requires that they publish data to a global repository with a standardized schema so that B2B customers can use web services to locate and order their products electronically. Secondly, they want to provide all of the product information to their corporate sales team both for current products and newly coming products. In addition they provide both a call center and a web site for retail customers to place orders. This order system needs to deal with a set of business policies regarding the application of different discounts during different sales periods.

The system should be automated with human intervention kept to a minimum. As each line of business provides information updates about its product offerings the information needs to be propagated to the various systems used for selling the products. However, the data produced by each line of business may contain product details or descriptions that contain jargon or line of business specific information that is not needed for sales. Therefore, the data must be cleansed as it flows from each of the lines of business into the sales systems, removing, adding and changing the information appropriately.

All these processes must be automated to the greatest degree possible in order to make sales information available quickly as the lines of business update their product information.

## Process Flow

- The company business personnel **update product catalog** in a PeopleSoft machine. Possible updates include
  - ▶ new catalog entries created
  - ▶ existing catalog entries modified
  - ▶ existing catalog entries deleted
- The updated parts of the product catalog flows into a business process for **cleansing** purpose. **Human being interaction** may be required when manual cleansing is required or when exception occurs.
- At the end of the business process, the cleansed product catalog delta is used to **synchronize** the product catalog information in a SAP system and a Siebel system.
- The product catalog delta is also **published to global repository** to allow customers to retrieve the up-to-date product information from the company.
- Customers **browse the product catalog**, order products and check order status from the company portal. Outstanding orders can be cancelled

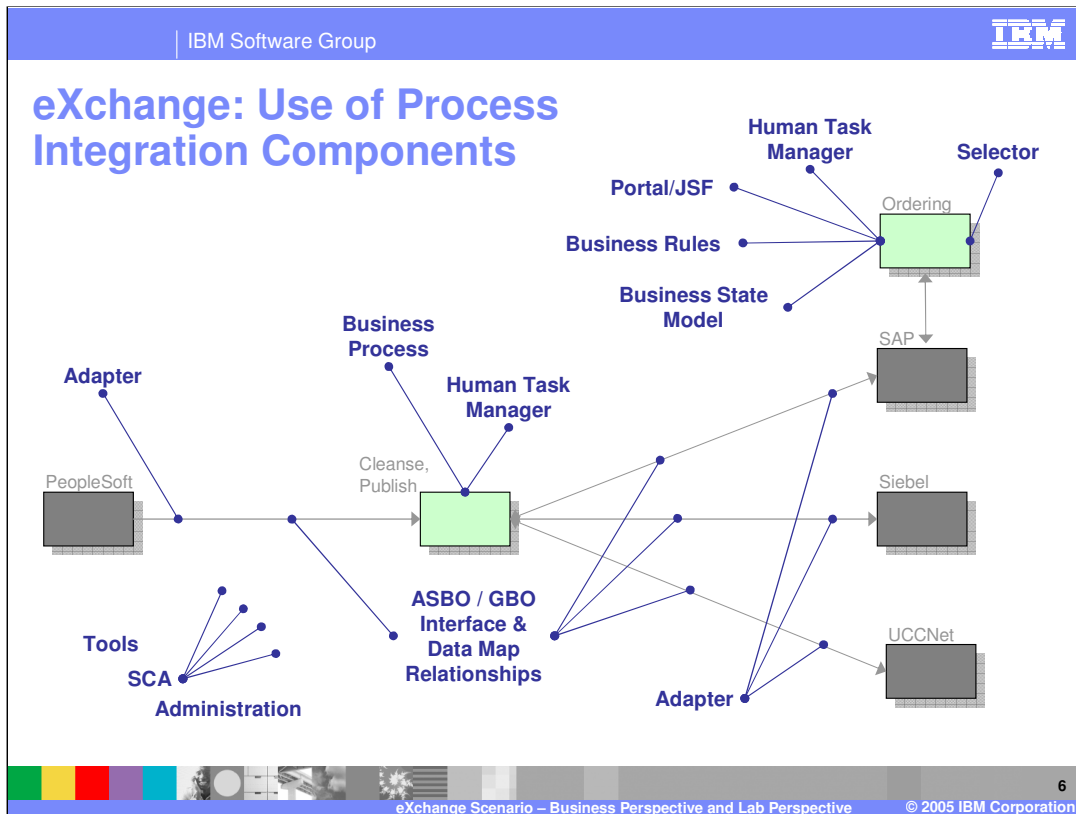


eXchange Scenario – Business Perspective and Lab Perspective

© 2005 IBM Corporation

Now that you understand the requirements, you will look at the process flow which is depicted in the graphic on the lower right hand side. The PaperClip division of ClipsAndTacks.com will be used as the line of business for the example.

The flow begins on the left side of the diagram with the PeopleSoft system. When the PaperClip division updates their product offerings in PeopleSoft, that information will be pushed to a cleanse/publish process shown in the middle of the diagram. The cleanse/publish process is used to cleanse the data, extracting the line of business specific details and jargon and updating the data so that it is ready for the various sales systems. The cleansing process is mostly automated, although there are times when human interaction is necessary to ensure accuracy. When the cleansing has completed the data is pushed out to SAP, used for the call center and web based ordering system; to Siebel, used by the corporate marketing team; and to the UCCNet global repository, used for B2B interactions.



On this slide, the different components that are a part of process integration are highlighted and shown where they are used in the overall process flow of the ClipsAndTacks.com system.

In the lower left, you see that tools, SCA infrastructure, and administration capabilities apply to the entire scenario.

The PeopleSoft, SAP, and Siebel applications and UCCNet global repository all require adapters in order to be able to exchange data with the Cleanse/Publish module in the server. The Cleanse/Publish module sits between them, receiving input from PeopleSoft and pushing it out to SAP, Siebel and UCCNet. There is a business process in the Cleanse/Publish module which is responsible for cleansing the data, including the use of a human interaction to ensure the data is ready to be published to the sales systems. As data flows from PeopleSoft to Cleanse/Publish and from Cleanse/Publish to SAP, Siebel and UCCNet there must be mapping operations to map between the application specific business objects of the external systems and the generic business objects used in the Cleanse/Publish module. In addition, in order to maintain a synchronization of data, there needs to be relationships established and maintained that link the data representing the same entities across the various systems containing the data.

In the Order Processing part of the application, there is a portal used to interact with the customers and the call center employees. The order processing logic is well suited to a state machine model, so a Business State Machine is used to drive the processing. The Human Task Manager is needed when a supervisor is required to perform a task, such as approving an order that requires some kind of exception processing. Because the system applies different discounts based on timing of the order and other criteria, business rules and selectors are used to allow business analysts to control how the discounts are

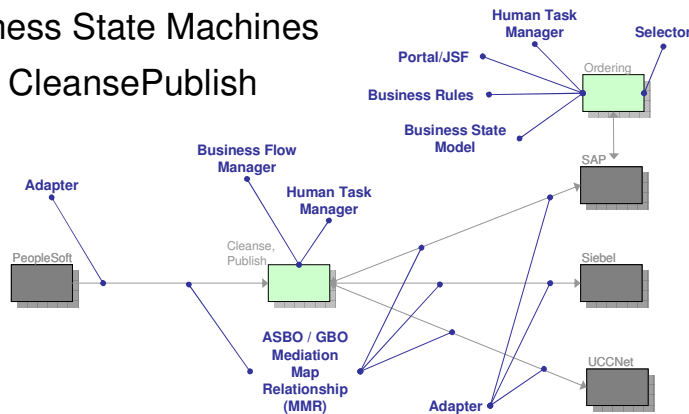
## Section

# ***Modifications to eXchange***

Now that you have an understanding of the application required by ClipsAndTacks.com, you need to look at how you can use this scenario in a set of lab exercises. Obviously you are not going to install PeopleSoft, SAP and Siebel to perform a lab exercise. The main focus is to illustrate the functionality of the WebSphere Process Server and how that functionality is built using the WebSphere Integration Developer. To that end, the scenario needs to be modified so that it can serve as the basis for realistic lab exercises.

## Business → Lab Scenario – Function Changes

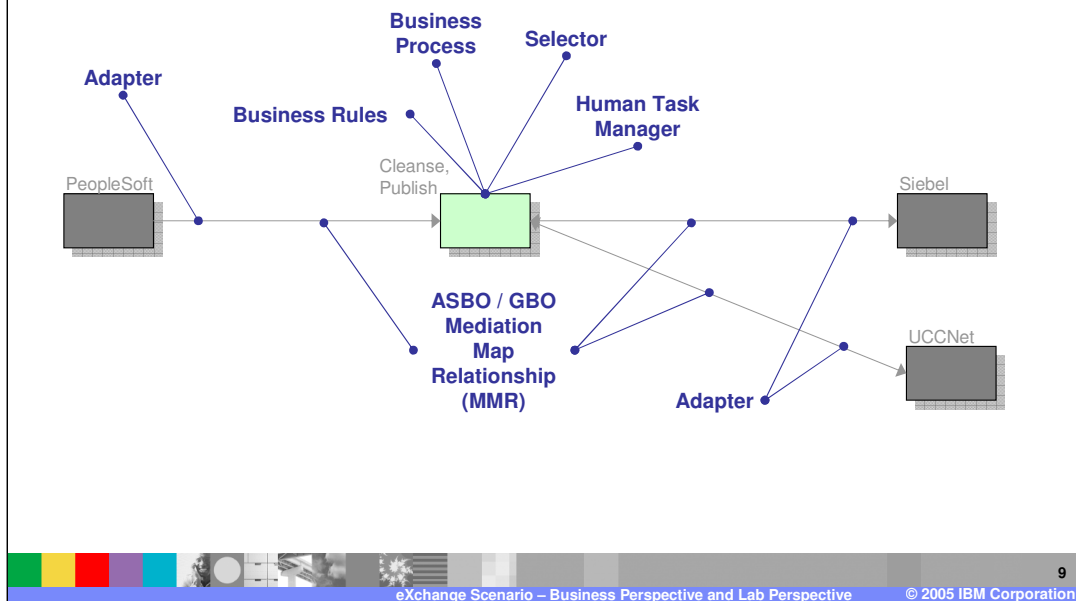
- Removed Ordering Function
  - ▶ No SAP application
  - ▶ No Portal/JSP
  - ▶ Dropped Business State Machines
  - ▶ Integrated into CleansePublish
    - Business Rules
    - Selectors



The first step of simplification is to remove the Ordering function. This removes the SAP application from the scenario which reduces the number of adapters, maps and relationships needed in the lab exercises. It also removes the Portal/JSP interface which is not part of the technology the lab exercises are designed to illustrate. The functionality of the Business Rules and Selectors is part of the technology the lab exercises are designed to illustrate, so they are included in the scenario by integrating a use for them into the Cleanse/Publish module. Finally, the use of the Business State Machine is removed from the scenario. This technology will be illustrated with a lab exercise that is independent of the eXchange scenario and is tailored specifically to Business State Machine capabilities.



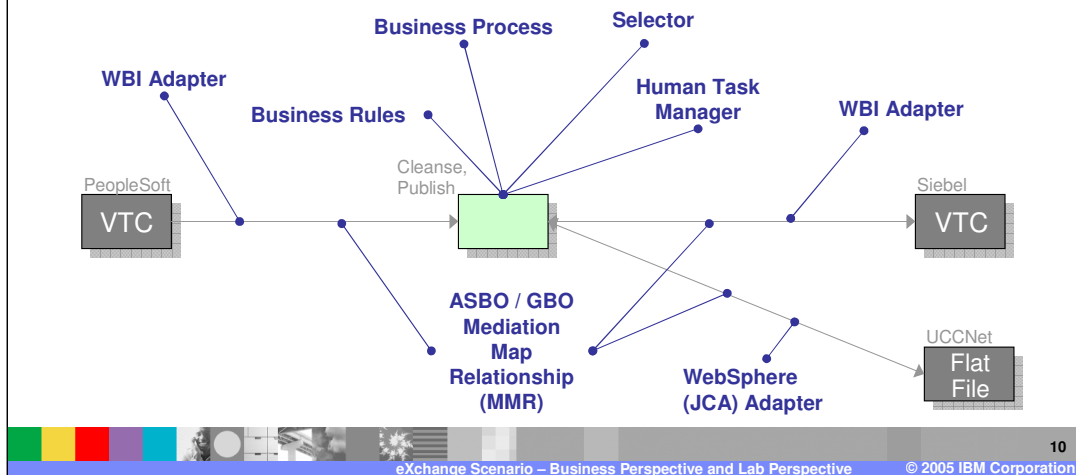
## eXchange – After Eliminating Ordering Function



After the set of changes discussed on the previous slide, the eXchange scenario is represented by this diagram. Notice that there is still the requirement of PeopleSoft, Siebel and UCCNet to run this scenario. This concern is addressed on the next slide.

## Business → Lab Scenario – Environment Changes

- PeopleSoft & Siebel – Use Visual Test Connector
- UCCNet – Write to Flat File



The WebSphere Business Integration (WBI) Adapter Toolkit provides a tool called the Visual Test Connector, or VTC, which is used to simulate an application that is using a WBI Adapter to communicate with a server. The Visual Test Connector is used for the lab exercises rather than requiring a real PeopleSoft or Siebel system to run the scenario. Because of this, the connections to PeopleSoft and Siebel are implemented with WBI Adapters rather than as WebSphere (JCA) Adapters. Rather than trying to interface with UCCNet, this part of the scenario is used to illustrate the use of the WebSphere (JCA) Adapter technology through use of the WebSphere Adapter for Flat Files and simply writing the UCCNet data to a file.

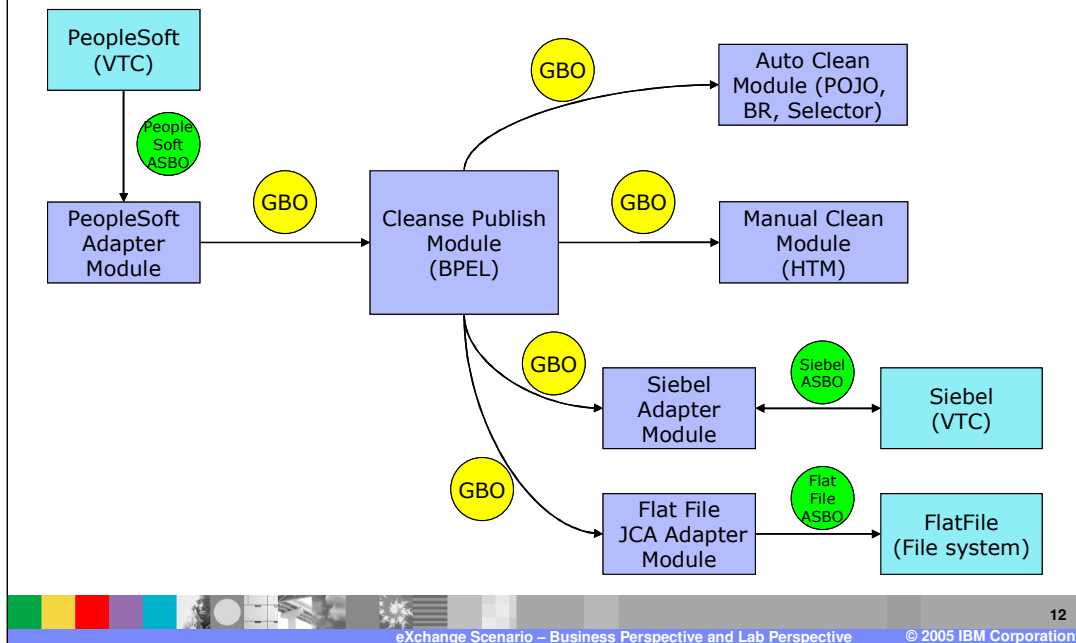
With this set of changes, the scenario is now shown as implemented in the lab exercises

## Section

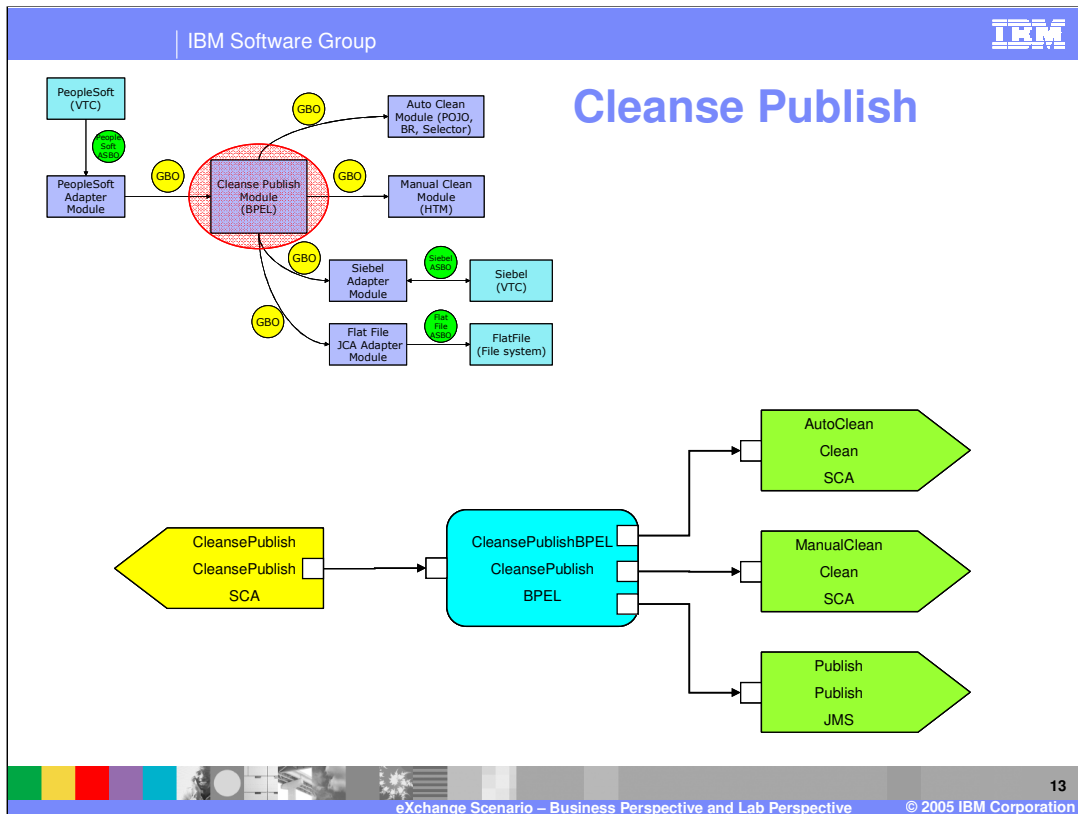
# *eXchange – SCA*

Now that you understand the business view of the eXchange scenario and how the scenario has been modified to make it practical for lab exercises, you will continue by examining the Service Component Architecture structure that makes up the scenario. You will first look at the overall scenario in terms of the SCA modules involved and will then look at the SCA components, imports and exports within each of the modules.

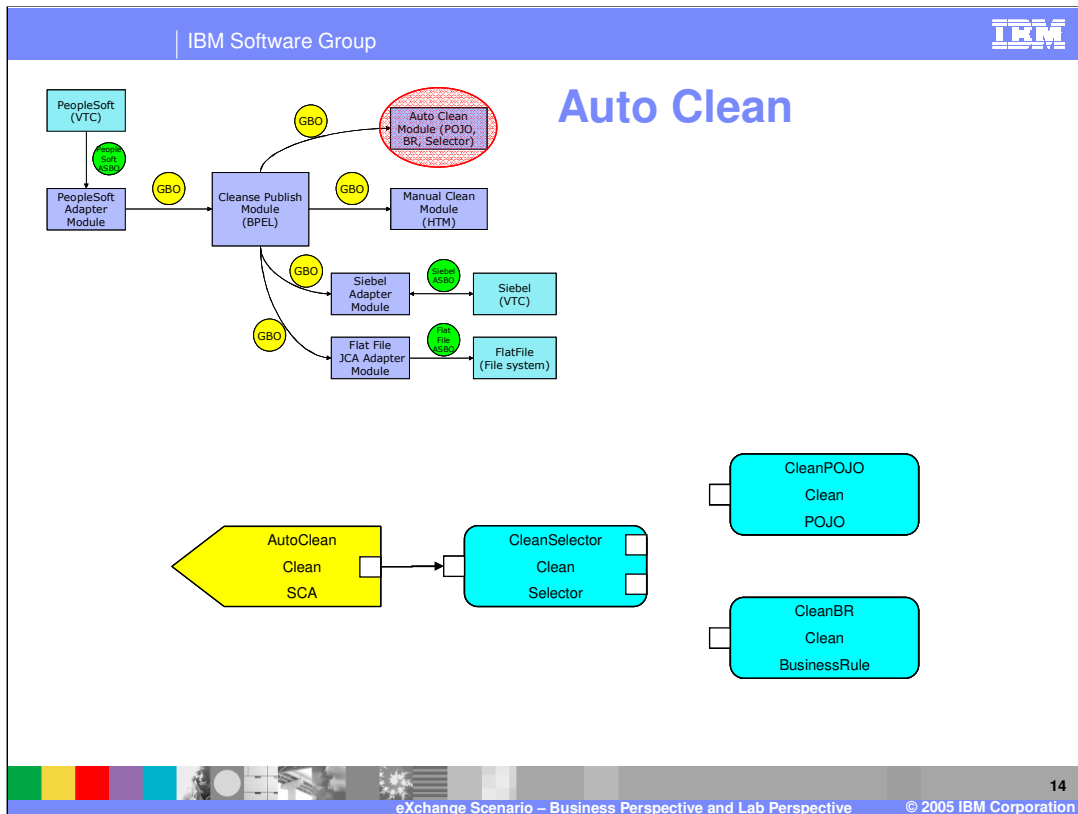
## eXchange Lab Modules – Overall Flow



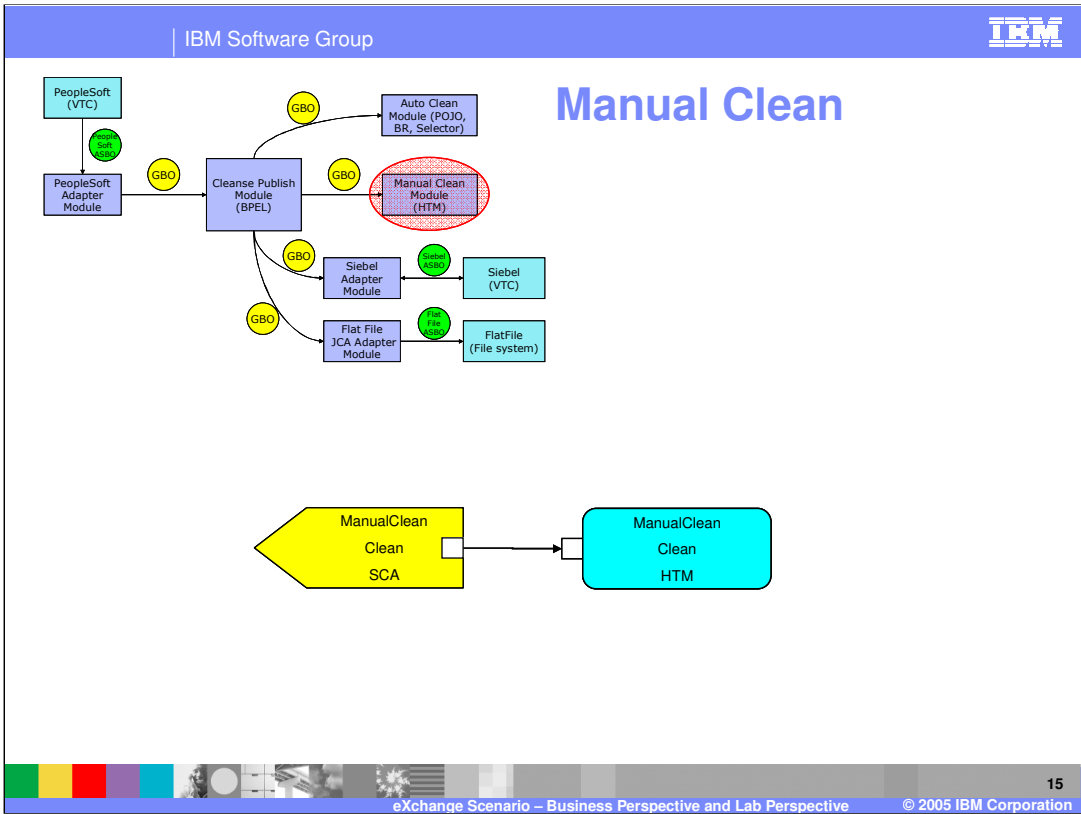
This diagram shows the Service Component Architecture modules that are involved in the eXchange scenario lab exercises. The blue boxes represent the individual SCA modules while the aqua boxes represent the non-SCA parts of the scenario. Starting in the upper left corner is the Visual Test Connector used to simulate the PeopleSoft application. It sends an application specific business object to the PeopleSoft Adapter module where it is converted to a generic business object and passed on to the Cleanse Publish module. The Cleanse Publish module is implemented in BPEL and controls the cleansing process. It first passes the generic business object to the Auto Clean module which manipulates the data based on selectors and business rules. The Cleanse Publish module then passes the updated generic business object to the Manual Clean Module which uses the Human Task Manager to complete the updating of the data. Then the generic business object is published to the Siebel Adapter module where it is converted to an application specific business object which is then passed to the Visual Test Connector representing the Siebel application. The publish step also passes the generic business object to the Flat File JCA Adapter module which converts it and writes it to a file using the WebSphere Adapter for Flat Files.



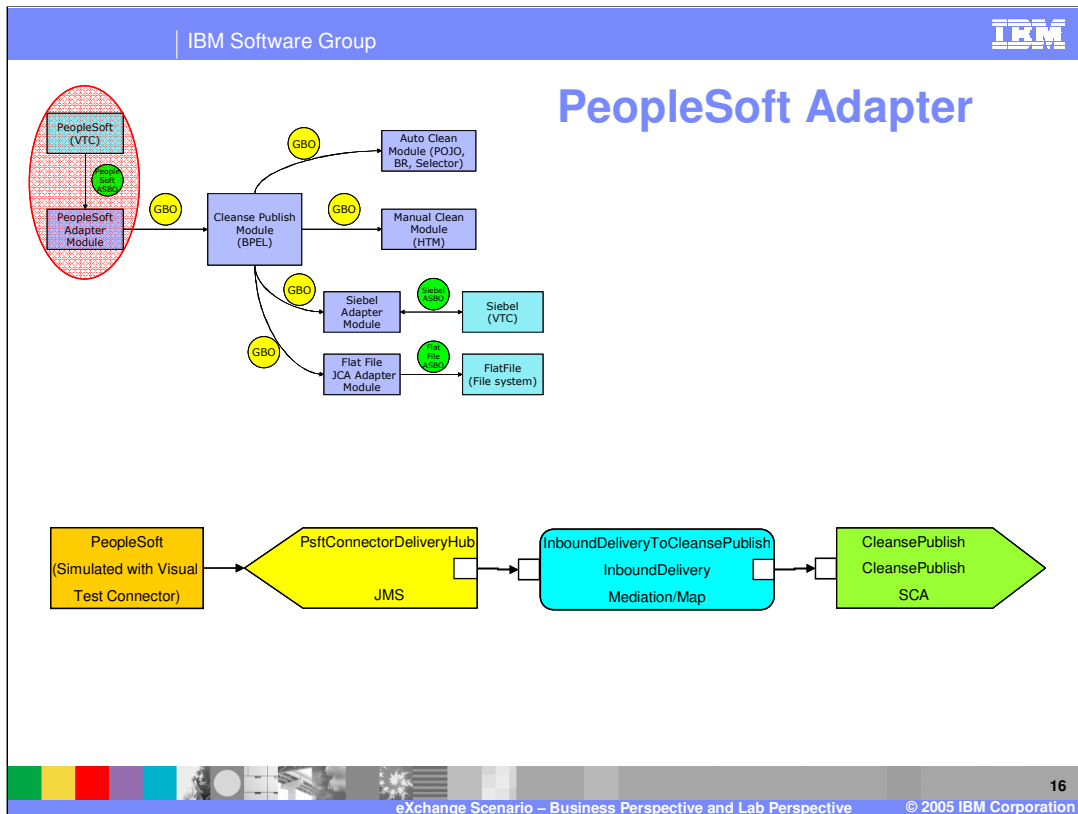
This slide describes the SCA wiring diagram for the Cleanse Publish module. The aqua box in the center is the CleansePublishBPEL component, which supports the CleansePublish interface and is implemented in BPEL. On the left is the export which exposes the CleansePublish interface for use by other modules and it has been defined to use an SCA binding. This export will be utilized by the PeopleSoft Adapter module to call CleansePublish. On the right side are the imports. Two of the imports support the Clean interface and are used to call out to the AutoClean and ManualClean modules and are defined with SCA bindings. The third import supports the publish interface. An interesting thing about this import is that this single import can be used to call both the Siebel Adapter module and the Flat File JCA Adapter module. This is done by using a JMS binding that is configured to publish to a JMS topic that can be subscribed to by multiple exports with JMS bindings.



This slide illustrates the Auto Clean module. On the left is the export which exposes the Clean interface by which this module is called. The CleanSelector component supports the Clean interface and is implemented as a selector which will use a date and time to determine whether the CleanPOJO or CleanBR component will be called. Both of these components also support the Clean interface, with the CleanPOJO using a Java™ implementation to update the generic business object while the CleanBR uses a Business Rule implementation to update the generic business object. Note that there are no wires connecting the CleanSelector component to either the CleanPOJO or CleanBR components. This is how selectors are represented in SCA wiring diagrams, without a visual cue as to which components it may be calling.

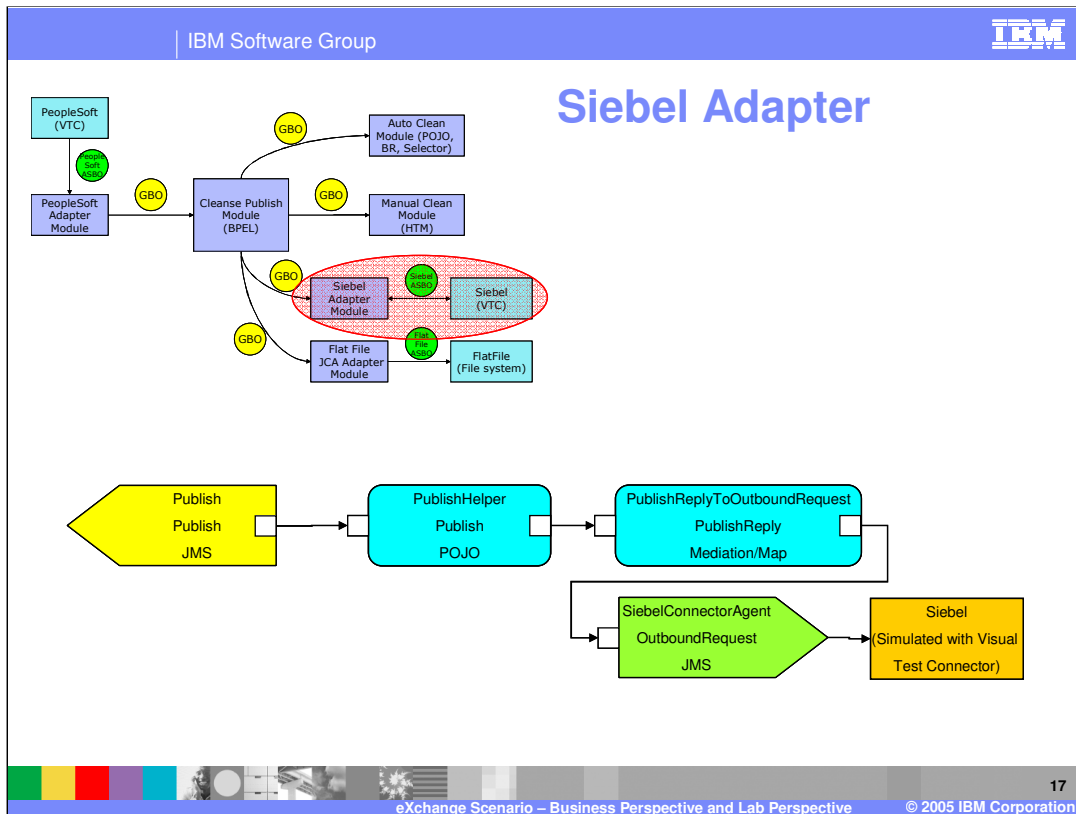


The ManualClean module is represented here. The export exposes the Clean interface used by the ManualClean component which is implemented as a Human Task. This human task enables further updating of the generic business object.



The PeopleSoft Adapter module is examined on this slide. The brown box on the left is not actually part of the PeopleSoft Adapter module, but rather is the Visual Test Connector used to drive input to the PeopleSoft Adapter module. The PsftConnectorDeliveryHub export uses JMS to receive input from the Visual Test Connector. This import understands that it is receiving data from a WBI Adapter and does the required that the business object coming in is in a form required by the WebSphere Process Server. The InboundDeliveryToCleansePublish component supports the InboundDelivery interface and is implemented as an Interface Mediation containing several Business Object Maps. These maps cover the incoming application specific business objects to generic business objects. In addition, there are Relationships defined within the maps that will allow synchronization of data between PeopleSoft and Siebel. The generic business objects are then passed to the import using the CleansePublish interface. This import uses an SCA binding to connect it to the export in the CleansePublish module.

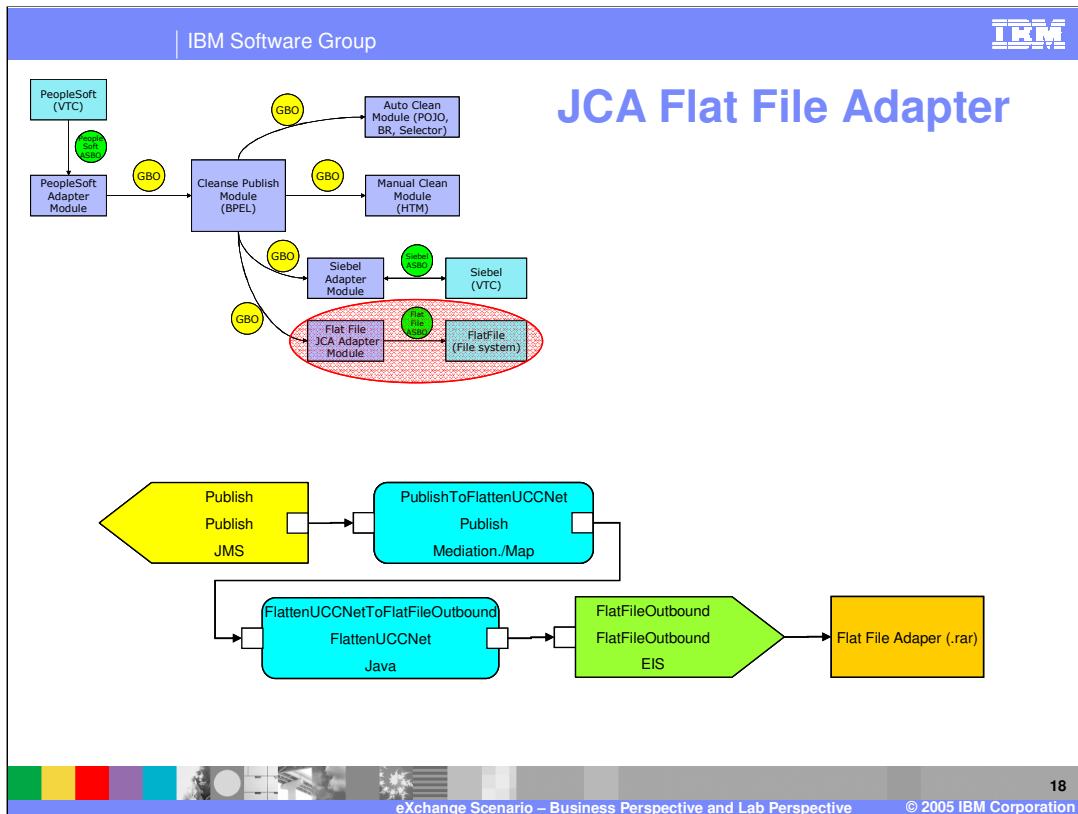




The Siebel Adapter module is examined on this slide. This module is slightly complicated because the Publish interface uses a one way operation rather than a request response operation, yet the inclusion of Relationship processing within the maps requires the component containing the business object maps to be called using a request response operation in order for the appropriate relationship processing to complete. Therefore, an additional component needs to be inserted to convert the one way operation to a request response operation.

With that background in mind, the following is the description of this module. On the left is an export that exposes the Publish interface. It uses a JMS binding that is configured to subscribe to a specific JMS topic, the same topic that the Cleanse Publish module uses on its import for the Publish interface. The incoming message is passed to the PublishHelper component where the Java implementation passes the call onward using the PublishReply interface. The PublishReply interface operation is the same as the operation on the Publish interface except that it is expecting a reply. In the PublishReplyToOutboundRequest component, the implementation is an Interface Map containing Business Object Maps which also define a Relationship. These maps convert the generic business objects to application specific business objects which are then sent over the OutboundRequest interface on the JMS Import that sends the message to the Visual Test Connector. When the Visual Test Connector replies, the response flows back through the import to the PublishReplyToOutboundRequest component where the relationship is updated. It then responds to the PublishHelper component where the return flow terminates.

WPSWIDv6\_exchangeBusinessAndLabD  
escription.ppt



This diagram shows the Flat File Adapter module. The Publish interface is exposed using an export that subscribes to a JMS topic, similar to the export used with the Siebel Adapter. This allows both the Siebel and Flat File Adapter modules to be invoked when the Cleanse Publish module does a publish. The PublishToFlattenUCCNet component contains an interface map and business object maps which take the incoming generic business objects and transforms them into the Business Object needed to be published on UCCNet. This is passed to the FlattenUCCNetToFlatFileOutbound component which uses a Java implementation to prepare the business object to be written to a file and sets up the call to the FlatFileOutbound import. This import interacts with the WebSphere Adapter for Flat Files to write the data to a file. As a note, the PublishToFlattenUCCNet component is not included in the lab exercises. Alternatively, the incoming generic business object is serialized in the FlattenUCCNetToFlatFileOutbound component using the business object services BOXMLSerializer method.

## Summary

- Developed an understanding of the eXchange business scenario
- Looked at how eXchange is modified for labs
  - ▶ Simplification of scenario
    - Reduce amount of explanation
    - Less setup
  - ▶ Still addresses major Process Integration components
    - Business State Machines will be covered outside of eXchange
- Looked at overall flow on SCA module basis
- Looked at module wiring diagrams

In this presentation you examined the requirements of ClipsAndTacks.com to understand the eXchange business scenario. In order to provide practical lab exercises based on this scenario it was necessary to simplify it and to reduce the setup dependencies and requirements. However, all the major components except Business State Machines are still addressed. You then looked at the Service Component Architecture module flow and the internal wiring diagrams for each of the modules. With this background, it will be easier to understand the lab exercises built around the eXchange scenario.

## Trademarks, Copyrights, and Disclaimers

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

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e (logo) business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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.

Information is provided "AS IS" without warranty of any kind. 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 (e.g., 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 2005. 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.