



IBM Software Group

WebSphere Business Modeler V6.2

Data maps



@business on demand.

© 2009 IBM Corporation
Updated April 14, 2009

This presentation will introduce you to the modeling elements available for creating business processes and business process flows with WebSphere® Business Modeler.

Goal

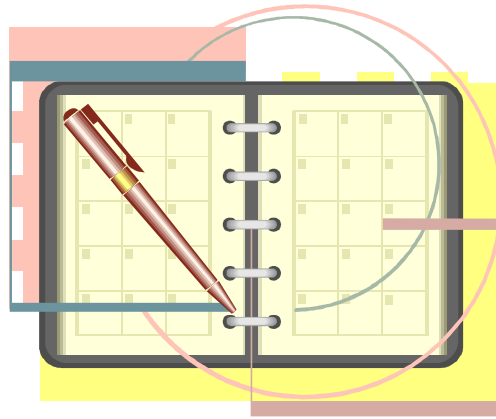
- To present the *data map* modeling element.



This presentation will present the data map element and some examples.

Agenda

- Introduction
- Examples

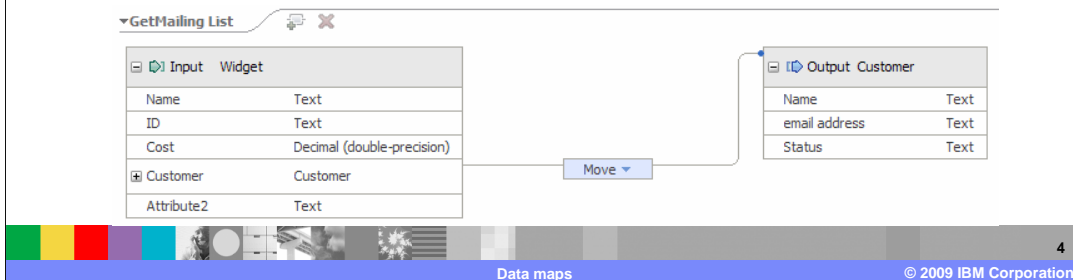
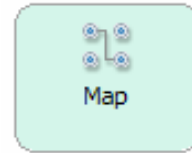


The map element is not complex or difficult to understand and is very handy when you need one.

This presentation introduces you to the map element and then gives several examples of how and where it is used.

Data map

- A specialized task that transforms data from one structure to another
- Map interfaces
- Synchronize data flow from parallel flows
- Intermediary between repository and loop or subprocess
- Same functionality as a local task



A map is a specialized task that transforms data from one structure to another. It provides a mapping editor that allows you to specify the mappings from the source to the destination using wires. This is illustrated in the picture on the bottom.

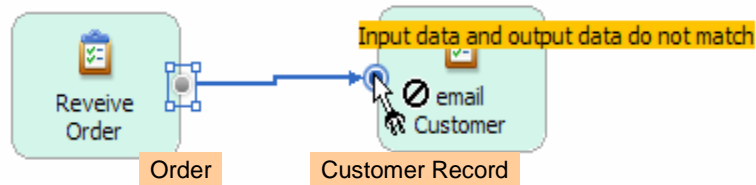
Maps are used to specify how to transform input into output, that is to say one interface to another interface. When the input for a task does not match the output from the previous task, it will need to draw the input from several different sources. When the input is a subset of the other sources, the map can be used to bring the parts together to form the business item needed for the input.

Maps can also be used to synchronize data flows from parallel flows, recompose complex business items or prepare a repository of data to be used by a loop or local subprocess.

Although a map task is provided on the palette, it provides the same functionality as a local task. It is provided only for diagramming purposes

Mapping interfaces

- Output of source does not match the input of the destination
 - ▶ In this case the *customer record* business item is a subset of the *order* business item



Shown here is an example of when the output of a task does not match the input needed by the next task. It so happens that the customer record is part of the order, therefore a map can be used to extract the customer record.

Mapping a subset

The screenshot displays a business process flow and a detailed mapping editor. The flow consists of three tasks: 'Receive Order', 'Map', and 'email Customer'. The 'Map' task is positioned between 'Receive Order' and 'email Customer'. The mapping editor shows the 'Input Order' table on the left and the 'Output Customer Record' table on the right. The 'Input Order' table has fields: Customer (Customer Record), CustomerNumber (Integer), CompanyName (Text), ContactFirstName (Text), ContactLastName (Text), StreetAddress (Text), City (Text), Country (Text), PostalCode (Text), Email (Text), Rating (Integer), and AvailableCredit (Decimal (double-precision)). The 'Output Customer Record' table has fields: CustomerNumber (Integer), CompanyName (Text), ContactFirstName (Text), ContactLastName (Text), StreetAddress (Text), City (Text), Country (Text), PostalCode (Text), Email (Text), Rating (Integer), and AvailableCredit (Decimal (double-precision)). The mapping editor shows four 'Move' operations connecting the input fields to the output fields: ContactLastName to ContactLastName, CompanyName to CompanyName, ContactFirstName to ContactFirstName, and Customer to Customer. The 'Rating' and 'AvailableCredit' fields are not mapped.

Continuing with the example, here you see the business process flow with the map element inserted between the “Receive Order” and the “email Customer” tasks.

Once the inputs and output for the map element have been established, you can then open the map editor using the button shown in the inset on the upper left. Notice that this is the attributes view and that there is a special tab for mapping.

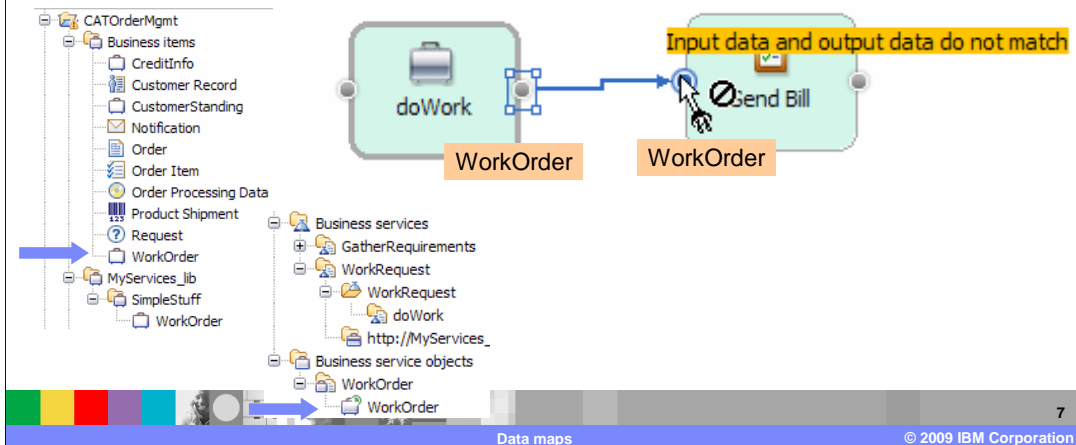
The inputs and outputs of the map element can be established by using the attributes view or by creating connections on the canvas. The types must be defined before they can be mapped.

Opening the mapping editor reveals the picture shown on the bottom. This is where you wire up the inputs to the outputs. The “move” operation is the only operation available. You can move one element from the input to an element on the output side, provided they are compatible types.

In this example, individual fields are being selected but you can also map the entire customer record in a single operation if you need to.

Mapping service interfaces

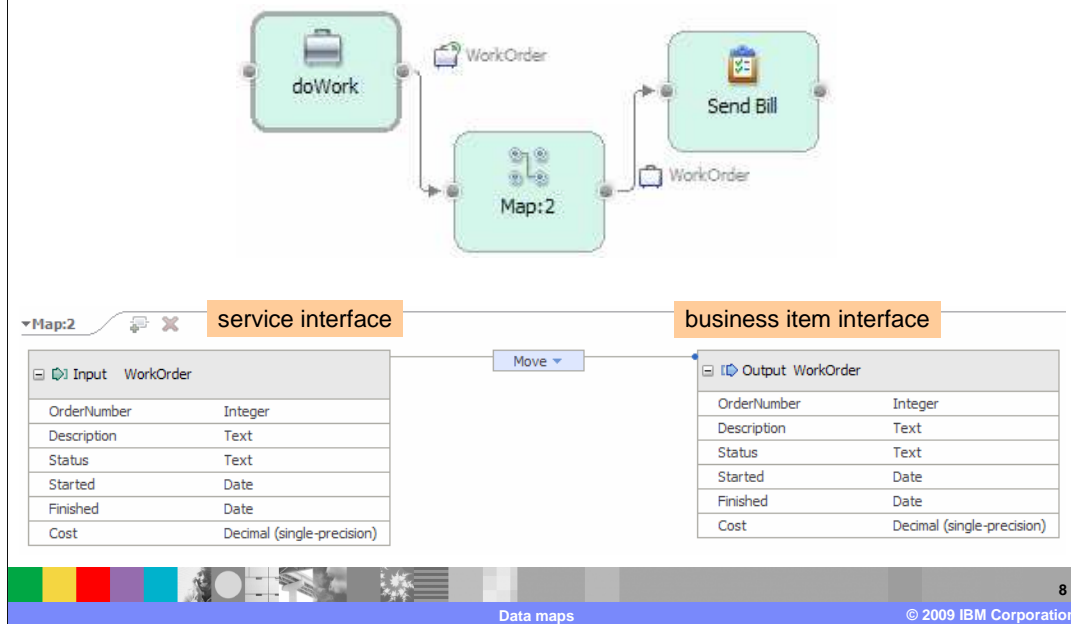
- When importing a business service and corresponding business object interface
 - ▶ It is not type compatible with a business item
 - Even if the business item is created by importing the same xsd



Maps are needed when working with business services and business service objects. What makes a business item type compatible is not always clear. In this case the definition of the “WorkOrder” was imported from an xsd file. For demonstration purposes it was imported into two places. Shown in the tree on the far left, it is a regular business item. In the second tree it is a “business service object”.

Although the structure of the two data elements is the same, they are not type compatible. In this case a map can be used to manage the type compatibility mismatch.

Mapping the service interface

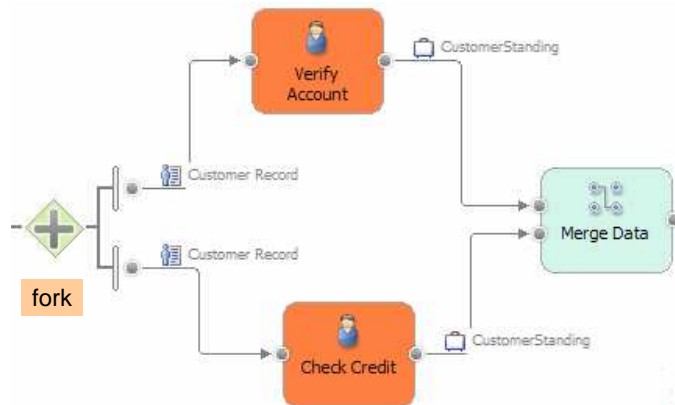


Here is the same flow with the map inserted. Notice the icon for the “WorkOrder” on the output of “doWork”. There is a little green icon behind the satchel icon. This indicates that it is a business service object.

In this example the entire structure is mapped in a single move operation.

Synchronizing data flow

- Parallel flows created by the fork
- Each flow contributes to different parts of the *CustomerStanding* business item
- Data map waits for both inputs

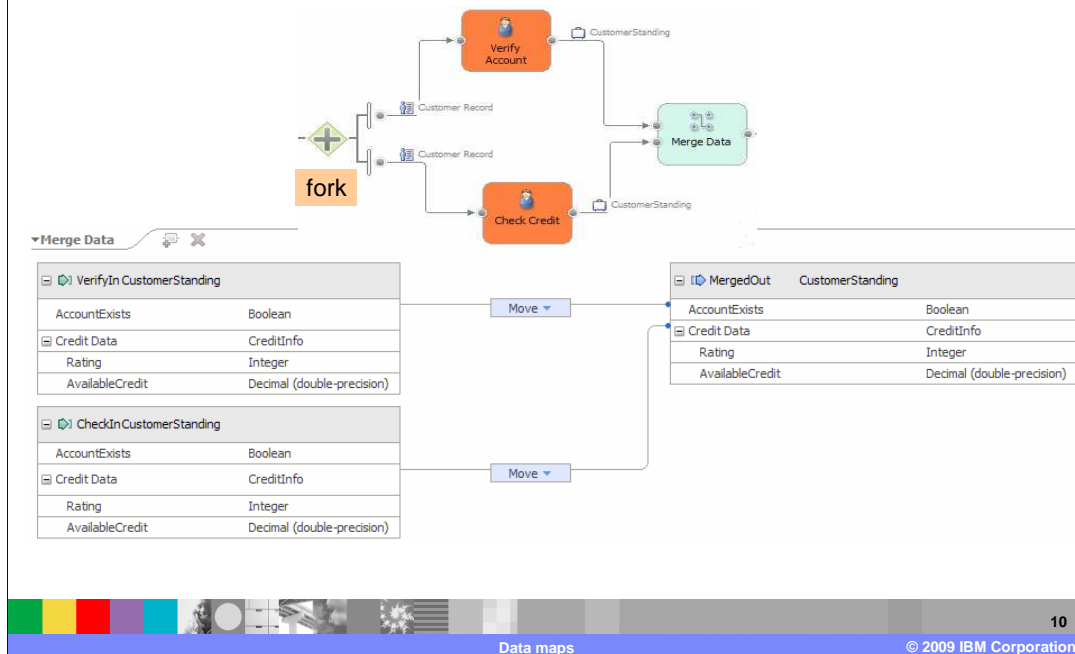


This case is not so obvious and when you get here you might ask yourself, does the map wait for all the arriving inputs?

The answer is yes, it does.

The example here shows the “CustomerStanding” business item being sent to two separate parallel tasks. “CustomerStanding” is a complex business item and each task updates different areas of it. The data map will wait for both inputs to arrive and then move the data from each input to its respective area in the customer record.

The mapping merge

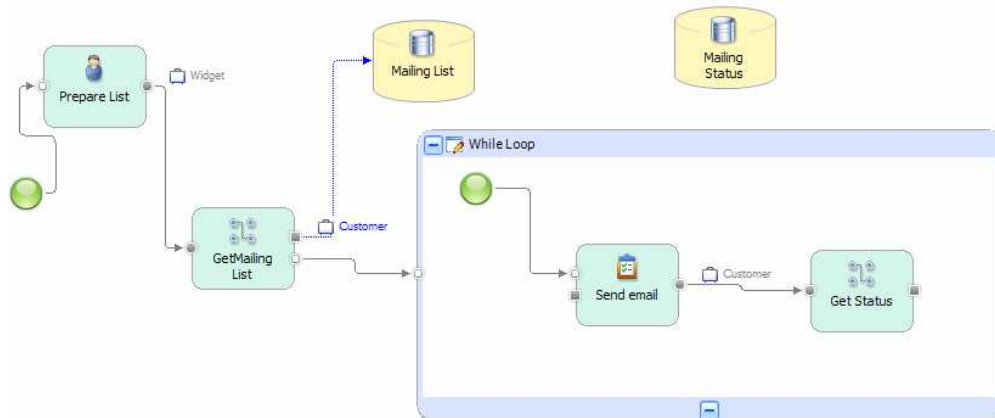


Each of the parallel flows contributes to the “CustomerStanding” business item.

The verify account task provides the value for the “account exists” and the check credit task provides the “credit data.”

When the business process is implemented with WebSphere Integration Developer or the interactive process design feature, it will wait for all the inputs to be satisfied before proceeding.

Map with loop



- Repositories are analogous to variables
- Square inputs and outputs are to repositories
- Open inputs and outputs indicate control flow



In this example a person prepares the customer list and then sends an e-mail to each person on the list. The purpose of this example is to show how the map can be used to prepare the local repository with the data needed by the tasks inside the loop.

The “Get Mailing List” map extracts the customer information from the widget business item and places it in the repository.

After the e-mail has been sent, the “Get Status” map extracts the status information and places it in the mailing status repository.

Summary

- A specialized task that transforms data from one structure to another
- Map interfaces
- Synchronize data flow from parallel flows
- Intermediary between repository and loop or subprocess
- Same functionality as a local task



To summarize, a map is a specialized task that is used to transform data from one structure to another. This is useful when matching up inputs and outputs to tasks and processes where the business items are not the same.

With the map, specific elements can be moved from one structure to another as long as the elements are of the same type. This is done using the mapping editor which allows you to graphically specify the move operations.

The map is useful when working with business services and business service objects where the data definitions are not type compatible with the regular modeler business items.

Maps can also be used to synchronize the data from parallel flows and to prepare repositories that need to pass data to loops or subprocesses.

Feedback

Your feedback is valuable

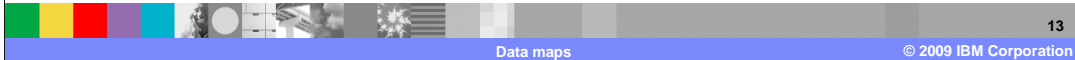
You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- 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_WBPMv62_Modeler_DataMaps.ppt

This module is also available in PDF format at: ..\\WBPMv62_Modeler_DataMaps.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>

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 NON-INFRINGEMENT. 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.

