

*Integrating Siebel 8.0 Applications*

## Module 4: Integration Objects

# 4

# Module Objectives

After completing this module you should be able to:

- Explain the role of internal and external integration objects
- Create an internal integration object
- Create an external integration object based on an XML schema definition (XSD)

Why you need to know:

- Integration objects provide the templates for importing and exporting data from a Siebel application

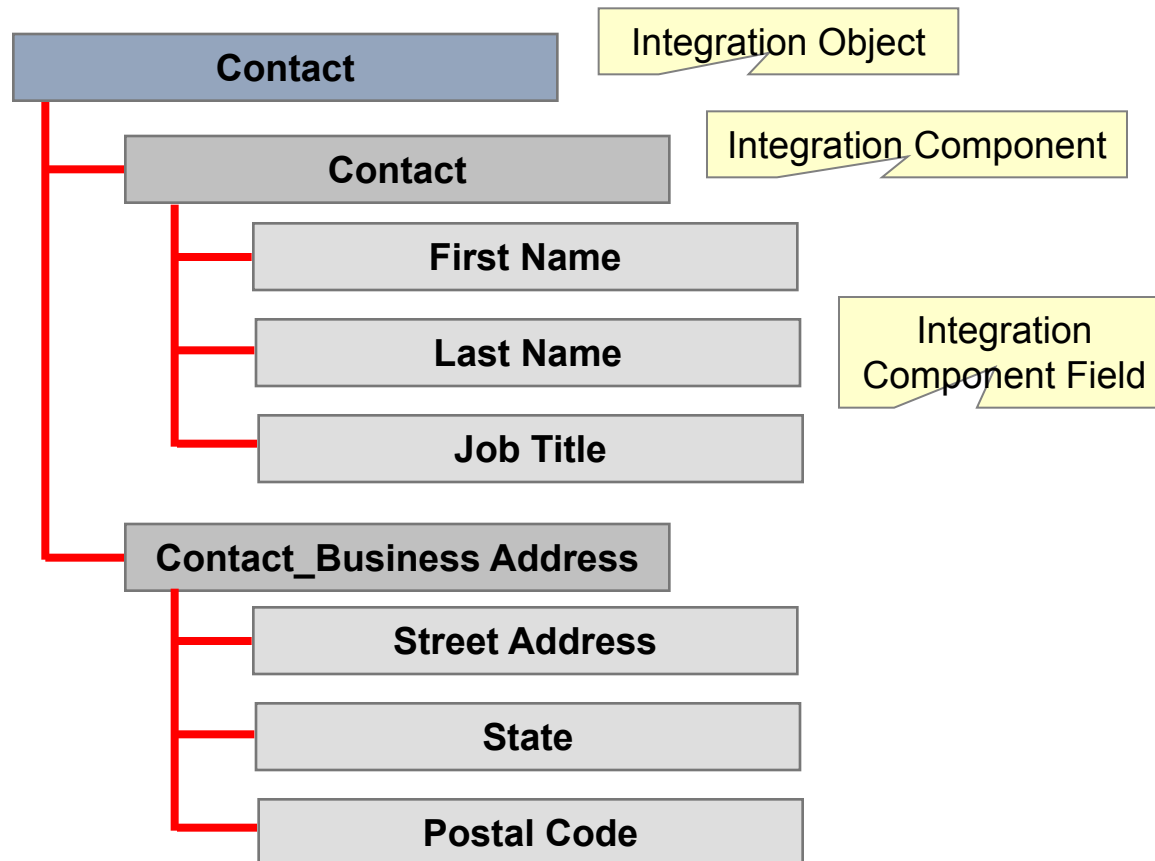
# Integration Object

---

- Defines the structure of data being exchanged between a Siebel and an external application
- Is used as a template to create memory-resident instances of data for processing by business services

# Integration Object Continued

- Is designed to represent hierarchical data structures
- Consists of multiple integration components
  - Each integration component consists of multiple integration component fields

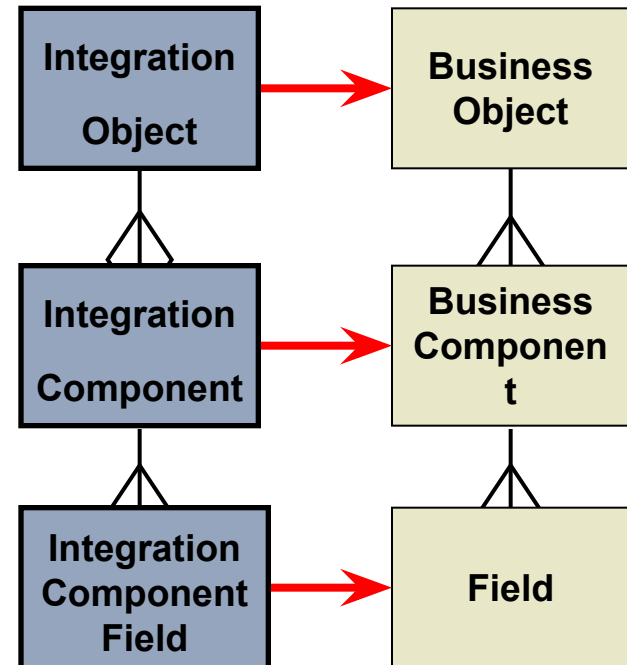
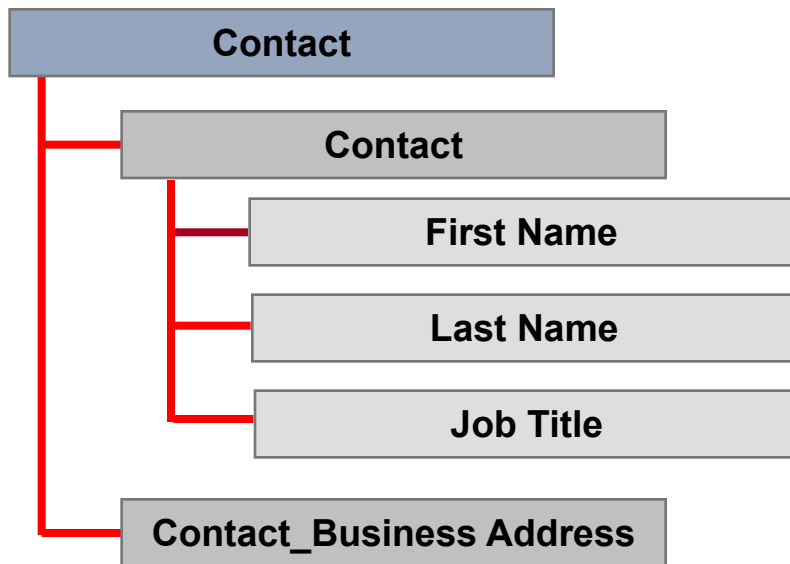


# Types of Integration Objects

- Internal integration object
  - Specifies the structure of the data extracted from or inserted into the Siebel application
- External integration object
  - Specifies the structure of the data that is transported between the Siebel and the external application
  - Is required only when the external application is not able to handle data in Siebel format (“Siebel data”)
    - Additional processing is required within the Siebel application to convert the integration object instance

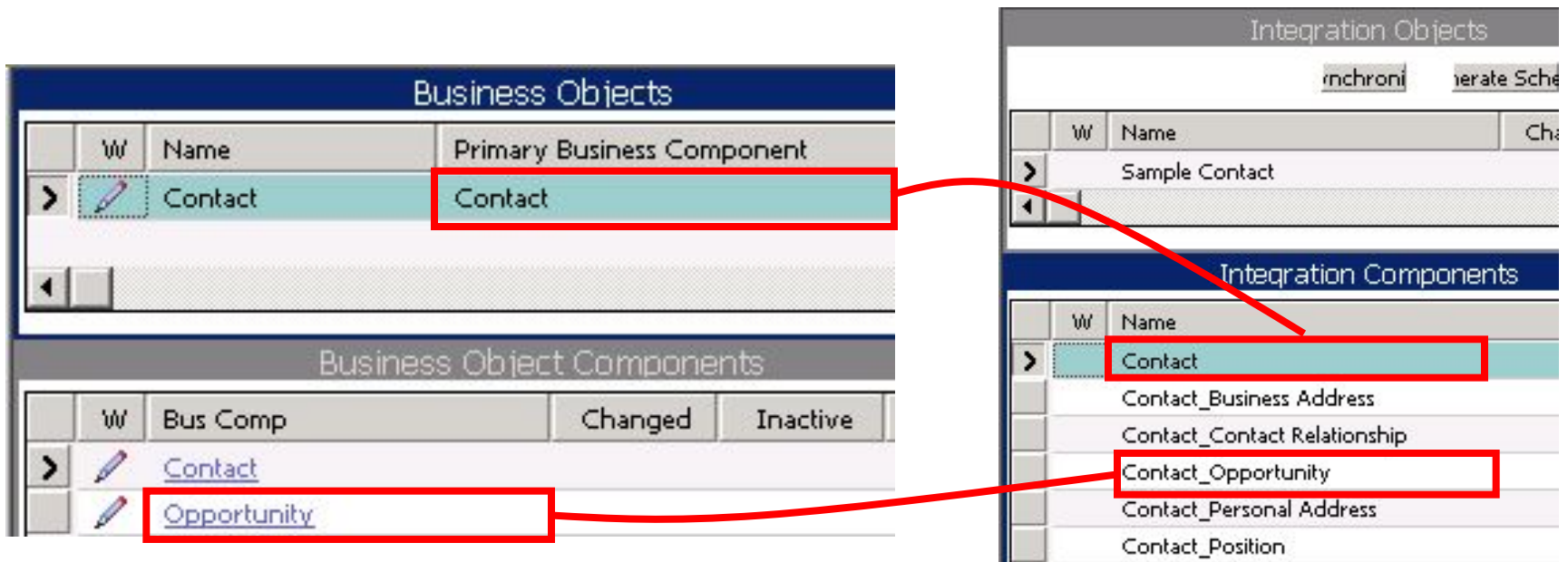
# Internal Integration Object

- Represents a subset of a Siebel business object
  - Integration components represent business components
  - Integration component fields represent business component fields



# Integration Components

- An integration object includes the following:
  - One parent integration component corresponding to the primary business component in a business object
  - Zero or more child integration components corresponding to other business components in the business object



# Integration Components Continued

- In addition, a child integration component represents the business component for each MVG in the primary and child business components

**Business Components**

Name	
>	Employee
<	

State MVF in Employee points to State MVF in Position

**Multi Value Fields**

Name	Multi Value Link
>	State Position

**Integration Objects**

Name	
>	Sample Employee
<	

**Integration Components**

Name	Parent Inter	
>	Employee	
	Position	Employee
	Position_Business Address	Position

State MVF in Position points to State MVF in Business Address

**Business Components**

Name	
>	Position
<	

**Multi Value Fields**

Name	Multi Value Link
>	State Business Address

In the integration object, these expand into Position and Position\_Business Address



# Integration Component Key

- Is a field or set of fields that uniquely identify a record being exchanged
- Is based on user keys for the tables referenced by the business components
- Is constructed by the Integration Object Wizard

Integration Components						
	W	External Name Context	Name	Changed	Parent Integration Component	
>		Contact	Contact			

Integration Component Keys						
	W	Name	Inactive	Changed	Key Sequence Number	Key Type
		Status Key			12	Status Key
>		V70 Wizard-Generated User Key: 1			1	User Key
		V70 Wizard-Generated User Key: 10			4	User Key
		V70 Wizard-Generated User Key: 11	✓		11	User Key
		V70 Wizard-Generated User Key: 2			2	User Key

# Status Key

- Is a special type of integration component key that is used to return the result of an operation, such as insert or update
  - Type is Status Key
  - Can be used to indicate:
    - Success or failure of an operation
    - Actual operation performed (for example during an upsert)
    - ROW\_ID for a newly created record
  - Integration component key fields specify data to be returned when status is requested

Integration Component Keys					
	W	Name	Changed	Key Sequence Number	Key Type
>		Status Key		12	Status Key
<					

Integration Component Key Fields						
	W	Name	Inactive	Changed	Field Name	Sequence
>		First Name			First Name	
		Last Name			Last Name	
		operation			operation	

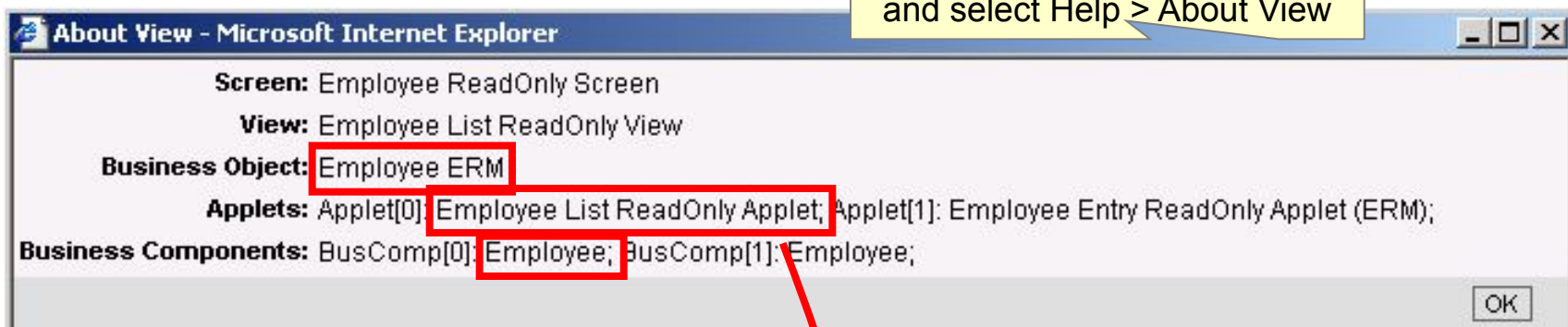
# Creating an Internal Integration Object

1. Identify the Siebel Data to Be Exchanged
2. Specify the Business Object
3. Select the Integration Components
4. Inactivate Unneeded Integration Component Fields
5. Inactivate Unneeded Integration Component Keys

# 1. Identify the Siebel Data to Be Exchanged

- Identify the data in the Siebel UI to integrate
  - Determine the underlying Siebel business objects, business components, and fields

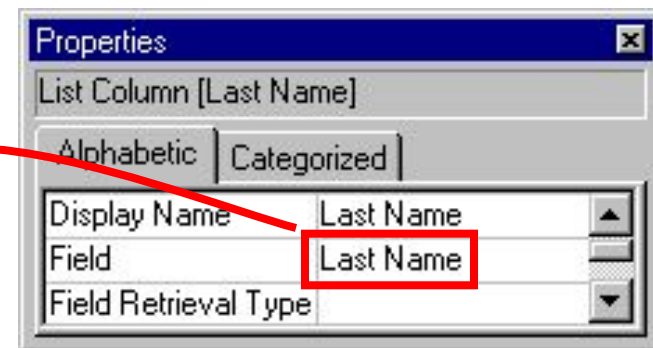
In the Client, display the view and select Help > About View



In Tools, select an applet and display the Web Applet Editor

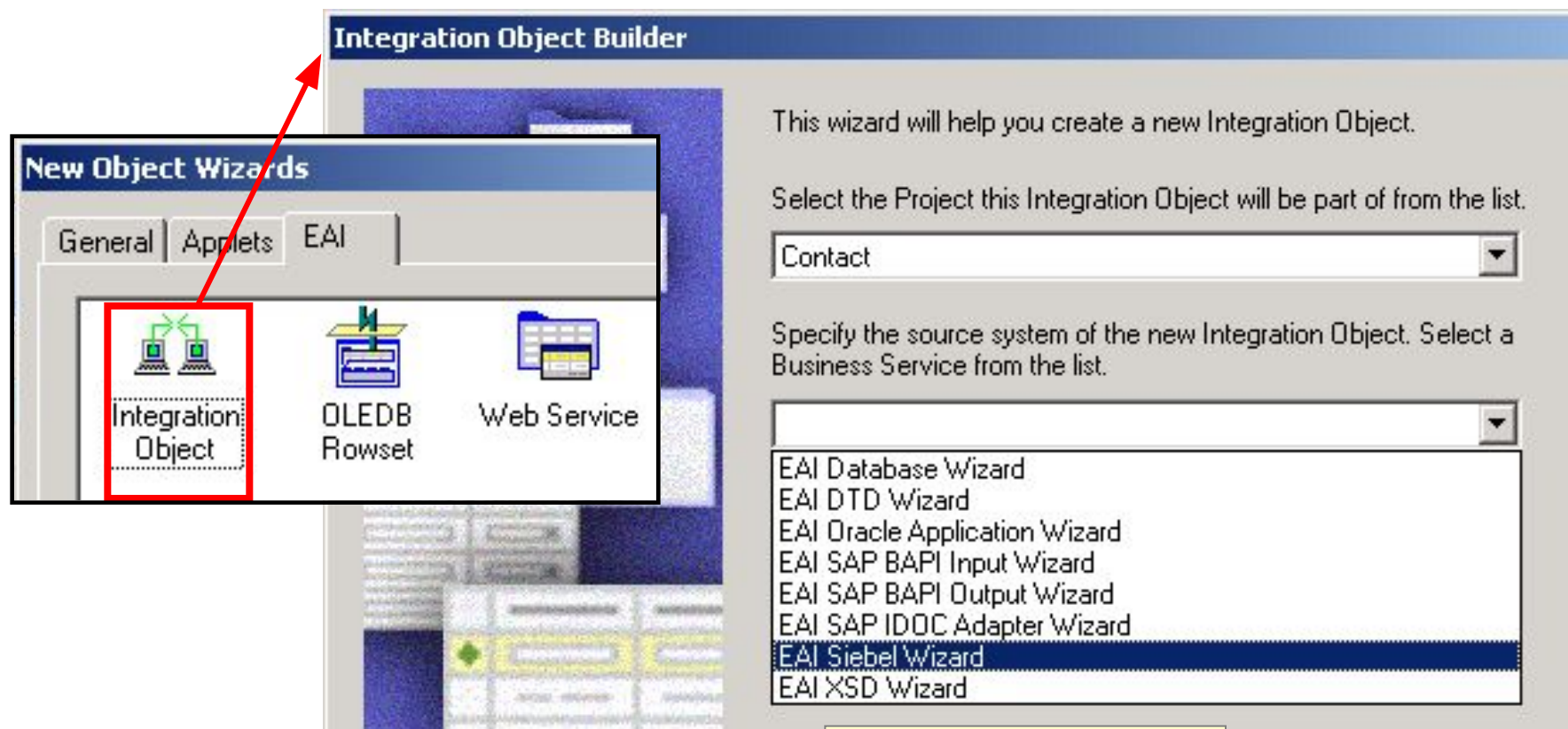


Select a field and display the Properties Window



## 2. Specify the Business Object

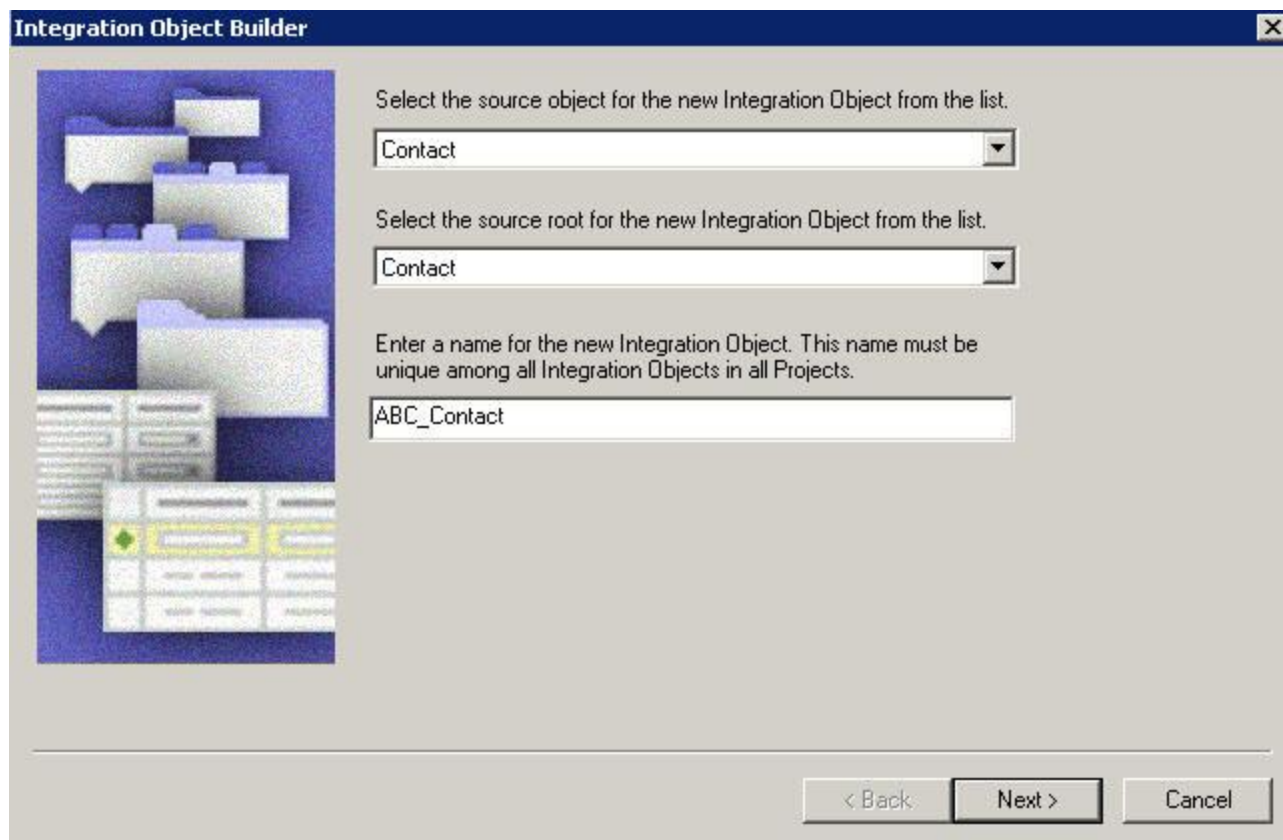
- Use the Integration Object Builder to create an integration object based on a Siebel business object
  - Select the EAI Siebel Wizard business service



Bases integration object  
on a business object

## 2. Specify the Business Object Continued

- Select the business object
- Provide a name for the integration object



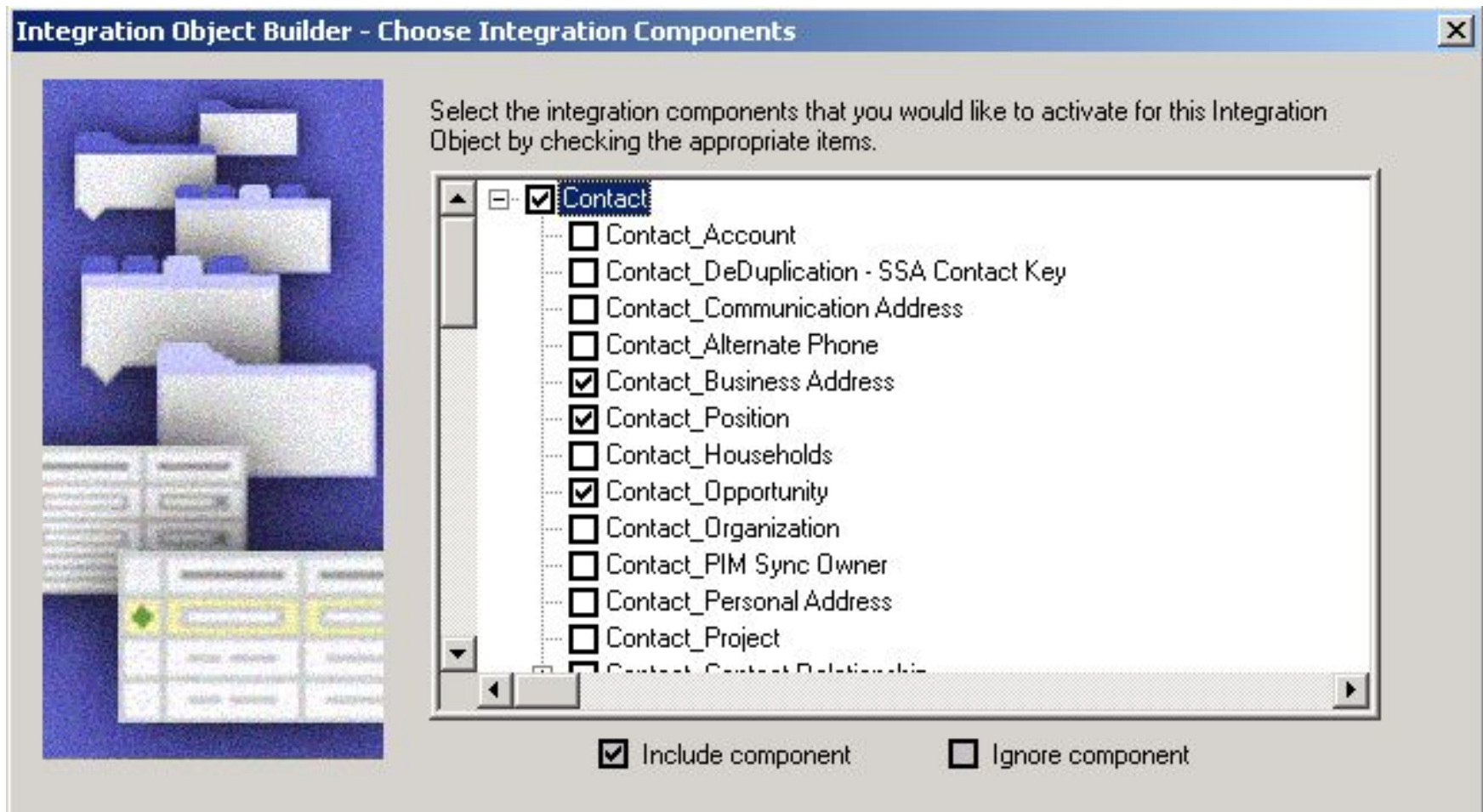
The screenshot shows the "Integration Object Builder" dialog box. It has a blue title bar with the text "Integration Object Builder" and a close button (X) on the right. On the left side, there is a graphic showing several white 3D-style folders and documents on a blue background. The main area of the dialog is light gray and contains the following elements:

- A text label: "Select the source object for the new Integration Object from the list." followed by a dropdown menu containing the text "Contact".
- A text label: "Select the source root for the new Integration Object from the list." followed by a dropdown menu containing the text "Contact".
- A text label: "Enter a name for the new Integration Object. This name must be unique among all Integration Objects in all Projects." followed by a text input field containing the text "ABC\_Contact".

At the bottom right of the dialog, there are three buttons: "< Back", "Next >", and "Cancel".

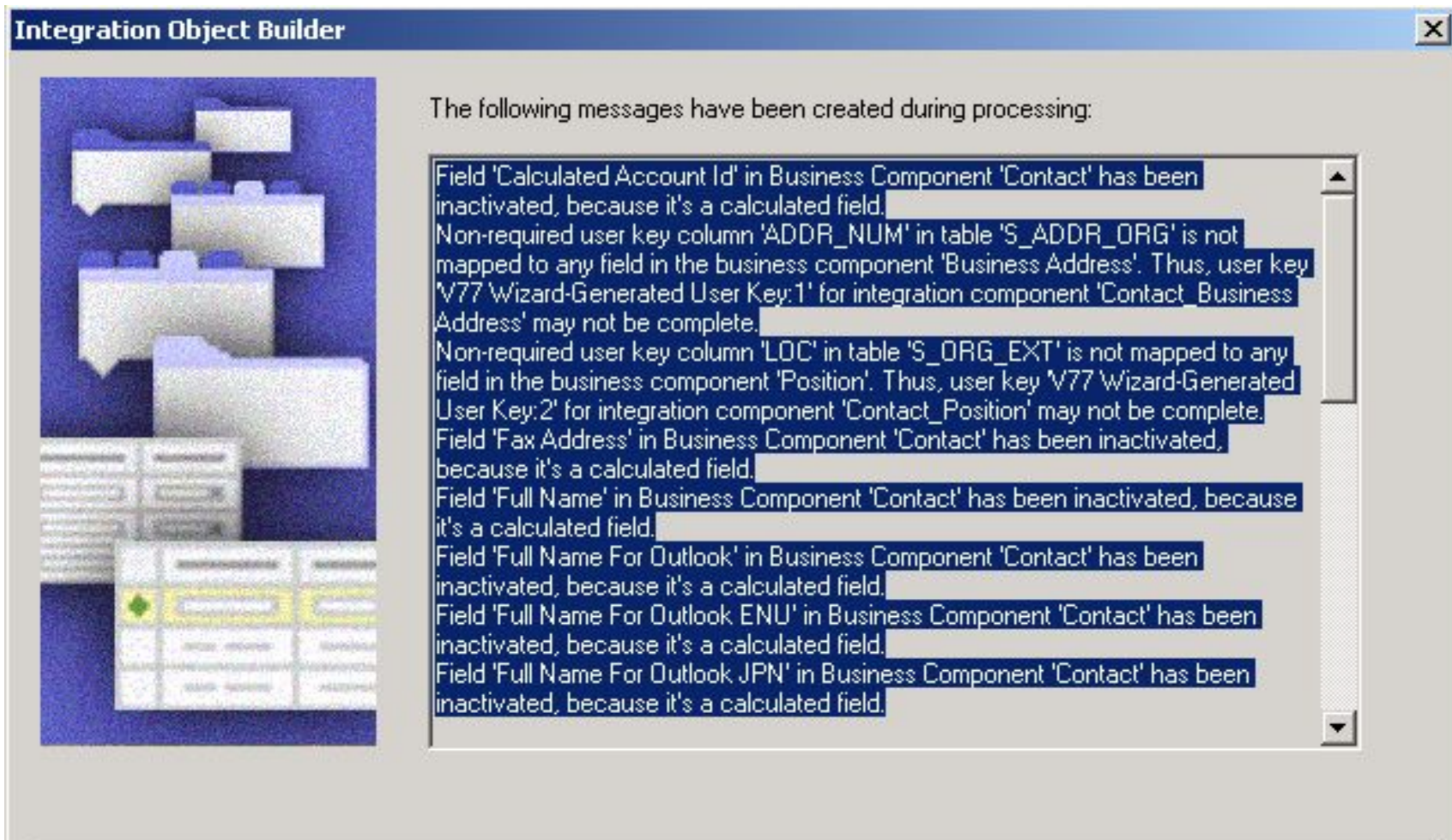
### 3. Select the Integration Components

- Select components to include in the integration object



### 3. Select the Integration Components Continued

- Review messages about fields made inactive
- Click Finish to configure the integration object





## 3. Select the Integration Components Continued

- Integration Object Wizard:
  - Finds all links, business components, and underlying tables
  - Identifies all business component fields that map to user keys in tables
  - Creates integration component user keys based on those fields
  - Notes any key columns that are not exposed in a business component
  - Creates calculated fields as integration component fields, but marks them inactive
    - Such fields cannot be updated

## 4. Inactivate Unneeded Integration Component Fields

- Set unneeded fields to inactive in each integration component to improve performance
- Do not delete unneeded fields—deleted fields become active when the business object is upgraded

The screenshot shows the Siebel Object Explorer interface. On the left, the 'Object Explorer' pane displays a tree view of Siebel Objects, including Siebel Objects, Applet, Application, Business Component, Business Object, Business Service, EIM Interface Table, Entity Relationship Diagram, Integration Object, and Integration Component. The main area is divided into two panes: 'Integration Components' and 'Integration Component Fields'.

The 'Integration Components' pane shows a table with columns 'W', 'Name', and 'Inactive'. The 'Contact' component is selected, and its fields are listed in the 'Integration Component Fields' pane. The 'Last Clnse Date' field is highlighted in blue, and its 'Inactive' checkbox is checked.

W	Name	Inactive
	Invoice Comments	<input checked="" type="checkbox"/>
	Job Title	<input type="checkbox"/>
	Joined Account Id	<input checked="" type="checkbox"/>
	Last Clnse Date	<input checked="" type="checkbox"/>
	Last Name	<input type="checkbox"/>
	Last Name, First Name	<input checked="" type="checkbox"/>

Field will not be included in the integration object

# 5. Inactivate Unneeded Integration Component

## Keys

- Make sure that the fields in the integration component keys are consistent with the component fields
  - Inactivate unneeded key fields in each integration component
  - Inactivate a key itself if it has no active fields

Integration Component Keys				
	W	Name	Changed	Key Sequ
>		V77 Wizard-Generated User Key:1	✓	1
		V77 Wizard-Generated User Key:2	✓	2

Inactivate entire key

Integration Component Key Fields				
	W	Name	Inactive	C
		Account Integration Id		
		Employee Number	✓	
>		Personal Contact	✓	

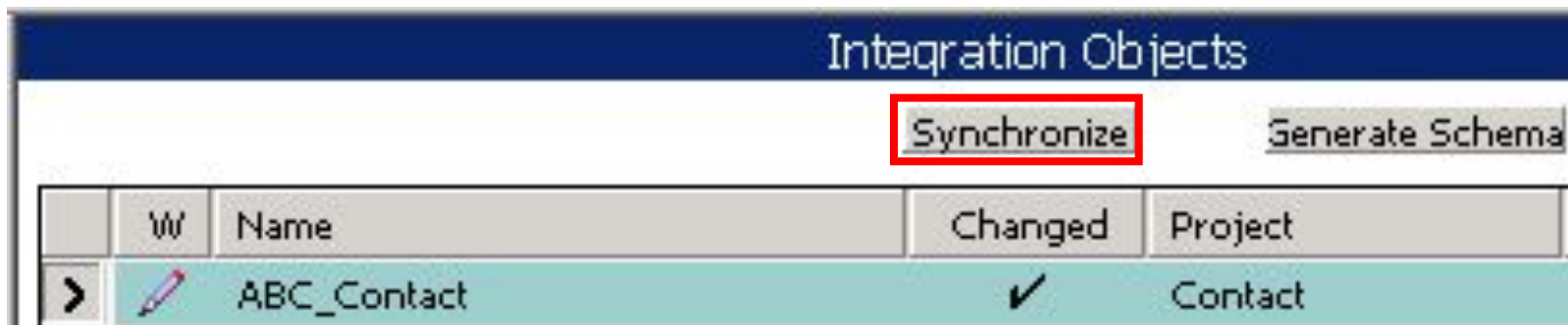
Inactivate select fields in key

Integration Component Keys				
	W	Name	Inactive	Changed
		V77 Wizard-Generated User Key:1		✓
>		V77 Wizard-Generated User Key:2	✓	✓

Integration Component Key Fields					
	W	Name	Inactive	Changed	Field
>		Person UId		✓	Per
		Personal Contact		✓	Per
		Primary Organization		✓	Prim

# Synchronizing Integration Objects

- When business objects are modified or updated, it is necessary to make sure the integration object is still consistent
  - If business object changes are minor, click Synchronize to update the integration object
    - Example: adding a new single value field
  - If business object changes are extensive, delete and recreate the integration object
    - Example: creating a new MVG



Integration Objects			
		Synchronize	Generate Schema
W	Name	Changed	Project
>	ABC_Contact	✓	Contact

# Creating an External Integration Object

1. Obtain a Schema of the External Data
2. Create the External Integration Object
3. Select Integration Components
4. Verify the Integration Object

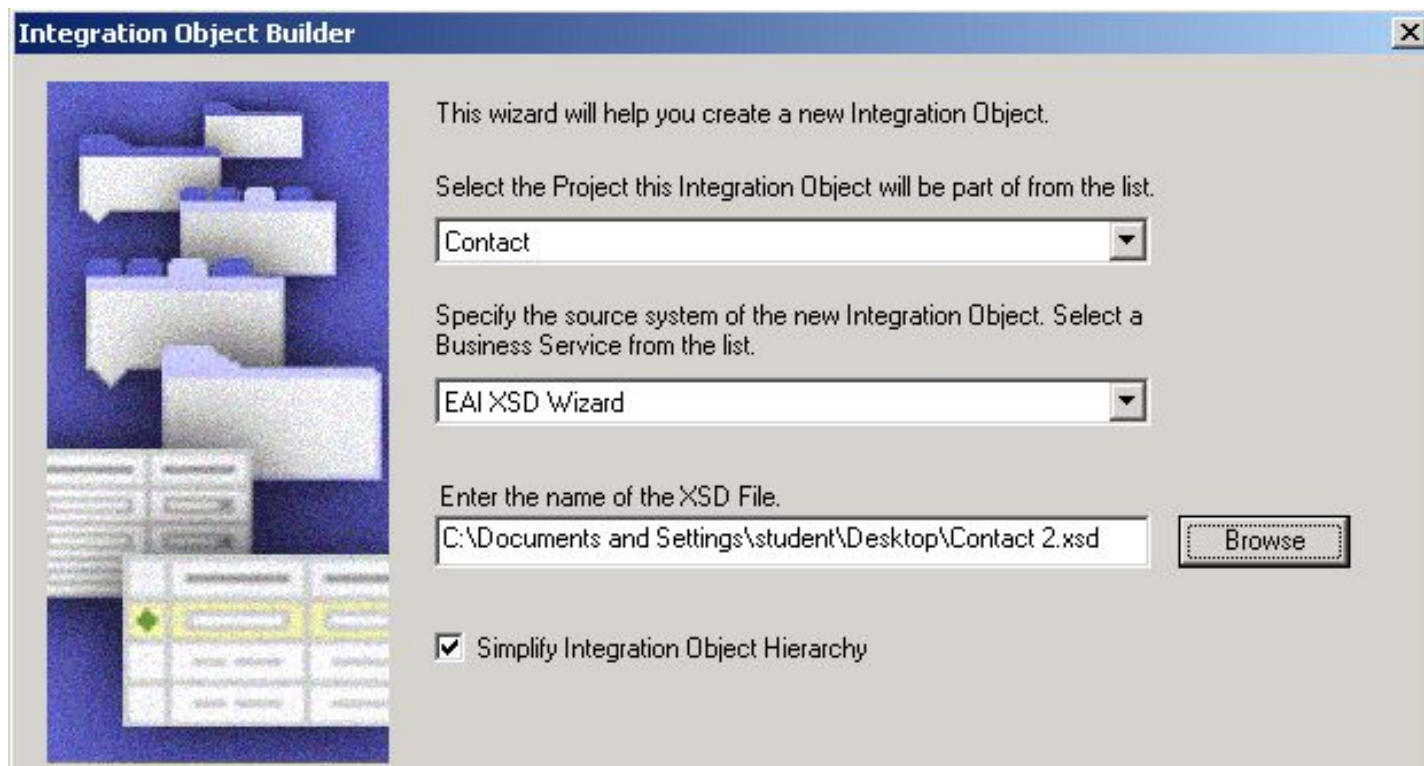
# 1. Obtain a Schema of the External Data

- Use the schema if published by the external application
- Alternatively, create a file containing a sample XML file received/sent by the external application
  - Use a third-party XML utility to generate the schema as an XSD

```
<?xml version="1.0" encoding="windows-1252" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://www.OU.org"
  targetNamespace="http://www.OU.org"
  elementFormDefault="qualified">
  <xsd:element name="Contact">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:element name="title" type="xsd:string"/>
        <xsd:element name="email">
          <xsd:complexType>
            <xsd:simpleContent>
              <xsd:extension base="xsd:string"/>
            </xsd:simpleContent>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

## 2. Create the External Integration Object

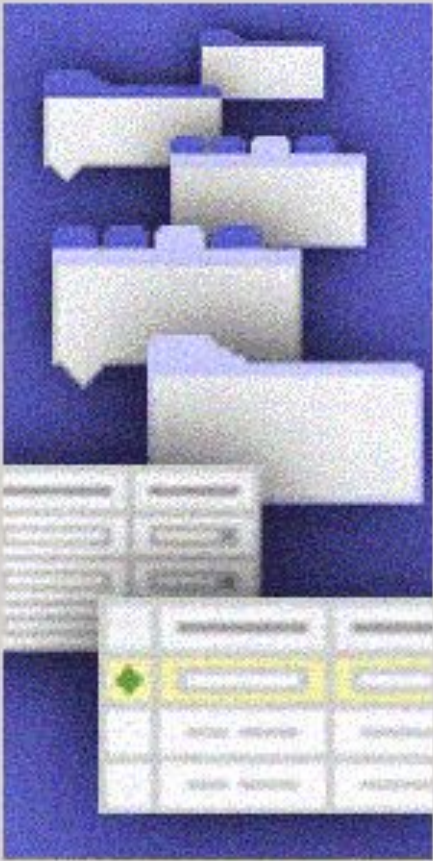
- Use the Integration Object Builder to create an integration object
  - Select the EAI XSD Wizard or EAI DTD Wizard business service
  - Check *Simplify Integration Hierarchy* to create an integration object with leaf elements as fields
    - Otherwise, all leaf elements are converted to integration components



## 2. Create the External Integration Object Continued

- Specify the source object to serve as the root-level node

**Integration Object Builder**



Select the source object for the new Integration Object from the list.

contact

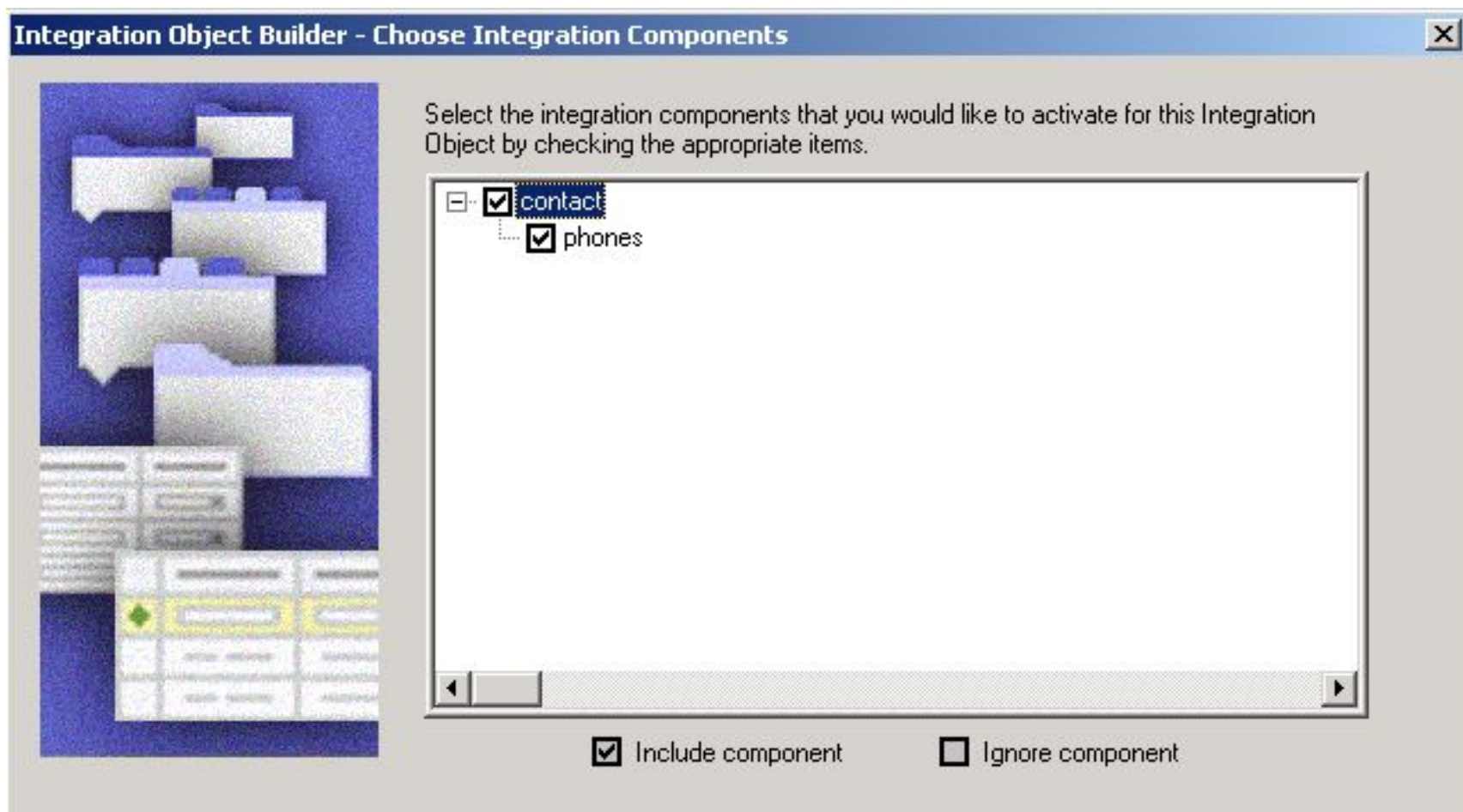
Enter a name for the new Integration Object. This name must be unique among all Integration Objects in all Projects.

XYZ\_Contact



## 3. Select Integration Components

- Select integration components



## 4. Verify the Integration Object

- Examine the configured integration object that contains the desired components and fields
- Inactivate unneeded integration component fields as necessary

Integration Components					
	W	External Name Context	Name		
>		/contact	contact		
		/phones	phones		
<					

Integration Component Fields					
	W	Name	Inactive	Changed	Data Type
>		email		✓	DTYPE_TEXT
		isActive		✓	DTYPE_TEXT
		name		✓	DTYPE_TEXT
		title		✓	DTYPE_TEXT

# Module Highlights

- Integration objects define the structure of data being exchanged between a Siebel and an external application
  - Integration objects are used as templates to create memory-resident instances of data for processing by business services
- Internal integration objects are a subset of Siebel business objects
- Use the Siebel Wizard to configure an internal integration object
  - Requires inactivating multiple fields and user keys
- Use the EAI XSD Wizard to build an external integration object based on the schema associated with external data

# Lab

---

- In the lab you will:
  - Create an internal integration object
  - Create an external integration object based on an external .xsd file