

---

# 3

## Creating a Sample Mapping

This chapter presents a walkthrough of the mapping process, using a sample database, based on a hypothetical Web site called AutoSource. Using this site, users can get price quotes for a variety of cars. To do this, they specify the make and model, and select a trim package. The *trim package* represents the features of a particular model, including its appearance and associated options packages. Many trim packages also have photographs that users can see.

The example in this chapter focuses on a database of trim packages provided by AcmeAuto for the AutoSource site. The AcmeAuto database contains two tables, whose internal structure might or might not be compatible with the data formats that the AutoSource database uses to display cars to visitors of that site.

In this example, you map data elements from an ATS schema to mapping targets in the Abstract Conceptual Model. From these mappings, you will generate a mapping specification.

You will see how to use the Modulant Balisarda Mapping Tool to create the mappings between the elements of the AcmeAuto database to elements of the Abstract Conceptual Model so that they can be transformed into a format that the AutoSource site can understand and display for users.

This walkthrough follows the stages discussed in [Chapter 2, “The Mapping Process,”](#) and includes the following sections:

- [Introducing the AcmeAuto Database](#)
- [Identifying the Mapping Targets](#)
- [Importing the ATS Schema](#)
- [Mapping Data Elements](#)
- [Defining the Transformation](#)
- [Creating a Mapping Specification](#)
- [Testing a Mapping Specification](#)

## Introducing the AcmeAuto Database

The AcmeAuto database used by the AutoSource Web site has two tables:

- **ACME\_TRIM** contains the characteristics of each trim package.
- **ACME\_TRIM\_IMAGE** contains a URL pointing to a photograph of each trim package, with information about the graphics file format.

Figure 7 shows the structure of the sample database, in EXPRESS-G graphical notation. For information about the symbols in this diagram, see [Appendix A, “EXPRESS-G Language Notation.”](#)

**Figure 7: The AcmeAuto Database**

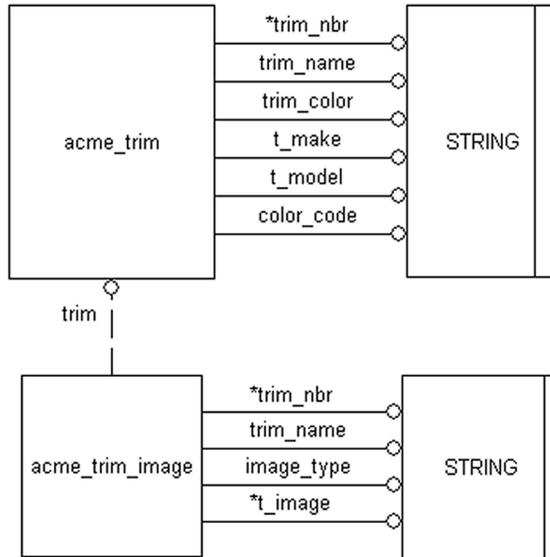


Table 1 shows the attributes of the **ACME\_TRIM** table, and Table 2 shows the attributes of the **ACME\_TRIM\_IMAGE** table.

**Table 1: ACME\_TRIM Table**

Attribute	Data Type	Description
<b>TRIM_NBR</b>	String	A unique identifier for each trim package.
<b>TRIM_NAME</b>	String	The name of the trim package.
<b>TRIM_COLOR</b>	String	The exact color specification of cars with a particular trim package; for example, <b>Powder Blue</b> .
<b>COLOR_CODE</b>	String	An abbreviation of a high-level classification of color; for example, <b>B</b> (for blue).

Table 1: ACME\_TRIM Table

Attribute	Data Type	Description
T_MAKE	String	The manufacturer of the car.
T_MODEL	String	The model of the car.

Table 2: ACME\_TRIM\_IMAGE Table

Attribute	Data Type	Description
TRIM_NBR	String	Identifies a trim package. This attribute is a migrated key representing <b>TRIM_NBR</b> in the <b>ACME_TRIM</b> table. Combines with <b>T_IMAGE</b> to provide a primary key.
TRIM_NAME	String	The name of the trim package. This attribute is the same as <b>TRIM_NAME</b> in the <b>ACME_TRIM</b> table.
IMAGE_TYPE	String	The file type of the graphic image.
T_IMAGE	String	A URL pointing to a graphic file containing the image of the trim package.
TRIM	<b>ACME_TRIM</b>	Identifies the relationship between this table and the <b>ACME_TRIM</b> table—each trim image represents a particular trim package.

## Identifying the Mapping Targets

To map the data elements in the AcmeAuto database effectively, you need to understand more about the meaning of each element and how they are used. Before you begin creating any mappings, it is important to identify the mapping targets in the Abstract Conceptual Model. This is a complex process, and requires a thorough grasp of the structure of the Abstract Conceptual Model.

Figure 8 shows a subset of the Abstract Conceptual Model with all of the target elements you will use in this example, and the relationships between those elements. The rest of this discussion centers on how you create this mapping. For more details about the target data elements, see [Chapter 4, “Understanding the Abstract Conceptual Model.”](#)

This figure shows the structure of the Abstract Conceptual Model in EXPRESS-G graphical notation.

Figure 8: Target Data Elements in the Abstract Conceptual Model

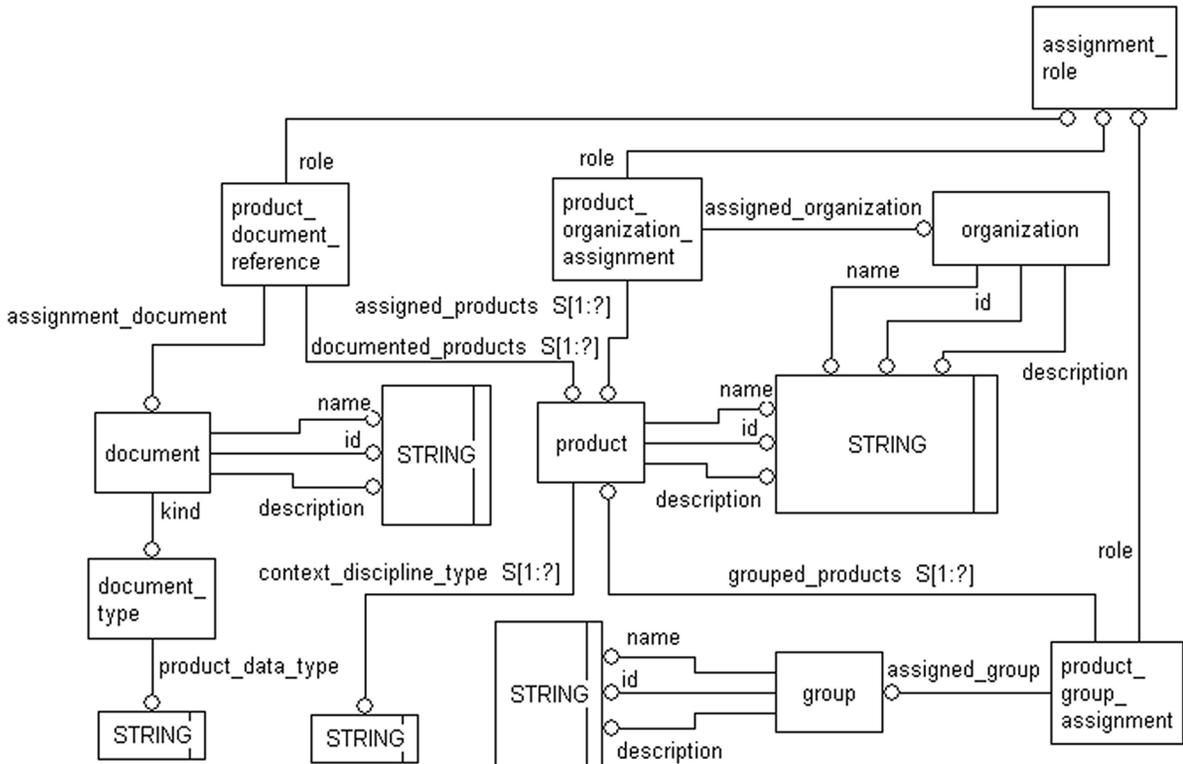


Table 3 shows where to map each of the data elements in the AcmeAuto database. The rest of this walkthrough shows you how to create the mappings. All mapping targets for data elements use the notation **entity\_name.attribute\_name**. When you create mappings, you always map data elements from the source ATS schema to attributes of entities in the Abstract Conceptual Model, never directly to the entities.

This table only shows the primary mapping target for each source data element. As you create each mapping, you will add references to other entities shown in Figure 8 to create paths between entities you have used as mapping targets. For specifics, see [“Mapping Related Data Elements”](#) on page 37.

Table 3: Mapping Strategy for the AcmeAuto Database

Source Data Element	Mapping Target	Explanation
ACME_TRIM: TRIM_NBR	product.id	This is the root element of the <b>ACME_TRIM</b> table. The main strategy is to treat each trim package as a product.
ACME_TRIM: TRIM_NAME	product.name	This is name of the trim package; it identifies which product you are talking about.
ACME_TRIM: TRIM_COLOR	group.name	Colors provide a classification scheme for trim packages. The <b>group</b> entity in the Abstract Conceptual Model represents classifications. The attribute <b>group.name</b> has unique values, as does <b>ACME_TRIM: TRIM_COLOR</b> .
ACME_TRIM: COLOR_CODE	group.name	The color code provides an additional level of classification, beyond the trim color. This creates a group that contains groups of trim colors.
ACME_TRIM: T_MAKE	organization.identifier	The manufacturer of a line of cars can be thought of as an organization—the company who makes the cars.
ACME_TRIM: T_MODEL	group.name	The model name of a car provides a way to classify the cars made by a particular manufacturer.
ACME_TRIM_IMAGE: TRIM_NBR	product.id	Same as <b>ACME_TRIM: TRIM_NBR</b> .
ACME_TRIM_IMAGE: TRIM_NAME	product.name	Same as <b>ACME_TRIM: TRIM_NAME</b> .
ACME_TRIM_IMAGE: IMAGE_TYPE	document_type. product_data_type	The type of image file represents what type of document contains the image.
ACME_TRIM_IMAGE: T_IMAGE	document.id	The URL pointing to the photograph of each trim package provides a way to document what that trim package looks like.
ACME_TRIM_IMAGE: TRIM	skip this one	Since this is a reference back to the <b>ACME_TRIM</b> table, do not map this data element explicitly.

This table does not capture the context information you will add to the mapping. In this walkthrough, you will see how to specify the context at the appropriate points in the process. For more information about determining where to add information about context in a mapping, see [Chapter 5, “Developing a Mapping Strategy.”](#)

Notice that a decision was made to map the idea of an AcmeAuto trim package to a product. This reflects how the application using this data thinks about each trim package. In other words, a trim package can be treated the same way as the application treats other products.

Notice also that several data elements are mapped to **group.name**. As the name suggests, many of the entities in the Abstract Conceptual Model are abstract. That is, like **group**, they can contain different categories of the same data. The entities **group**, **person**, and **organization** are examples of abstract entities.

You can use the idea of a *group* in the Abstract Conceptual Model to categorize a number of different things. When you add contextual information to each mapping, you distinguish among the different groups. In this example, colors and model names represent separate groups.

With the basics of the mapping strategy identified, you can start the next stage of the mapping process.

---

## Importing the ATS Schema

The sample AcmeAuto database is defined as an EXPRESS schema. For a brief description of EXPRESS, see “[The EXPRESS Data Modeling Language](#)” on page 7. For a more thorough description of EXPRESS, see the resources provided by the Universal Data Interoperability Framework at [www.UDIF.org](http://www.UDIF.org).

Before you can specify the ATS schema you want to map, you must start the Modulant Balisarda Mapping Tool and specify the locations of your working database and the Abstract Conceptual Model. Then you can specify the format and location of your ATS schema.

This section describes the following tasks:

- [Starting the Modulant Balisarda Mapping Tool](#)
- [Importing the ATS Schema](#)

## Starting the Modulant Balisarda Mapping Tool

Before you can start the Mapping Tool, you must do the following things:

- [Start the Modulant Curtana Analyst Tool](#)
- [Make a Copy of the Working Database](#)
- [Specify a Database Connection](#)
- [Specify the Abstract Conceptual Model’s Location](#)

## Make a Copy of the Working Database

Before you begin, follow these steps to make a copy of the working database that the Mapping Tool uses to store information about the mappings you define:

- 1 Using Windows Explorer, navigate to the **samples** directory of your Modulant Curtana installation.
- 2 Select the Access database **walkthrough-empty.mdb**.
- 3 Select **Edit > Copy**, and then select **Edit > Paste** to place a copy of this database in the **samples** directory.
- 4 Change the name of the copy from **Copy of walkthrough-empty.mdb** to **mt-walkthrough.mdb**.

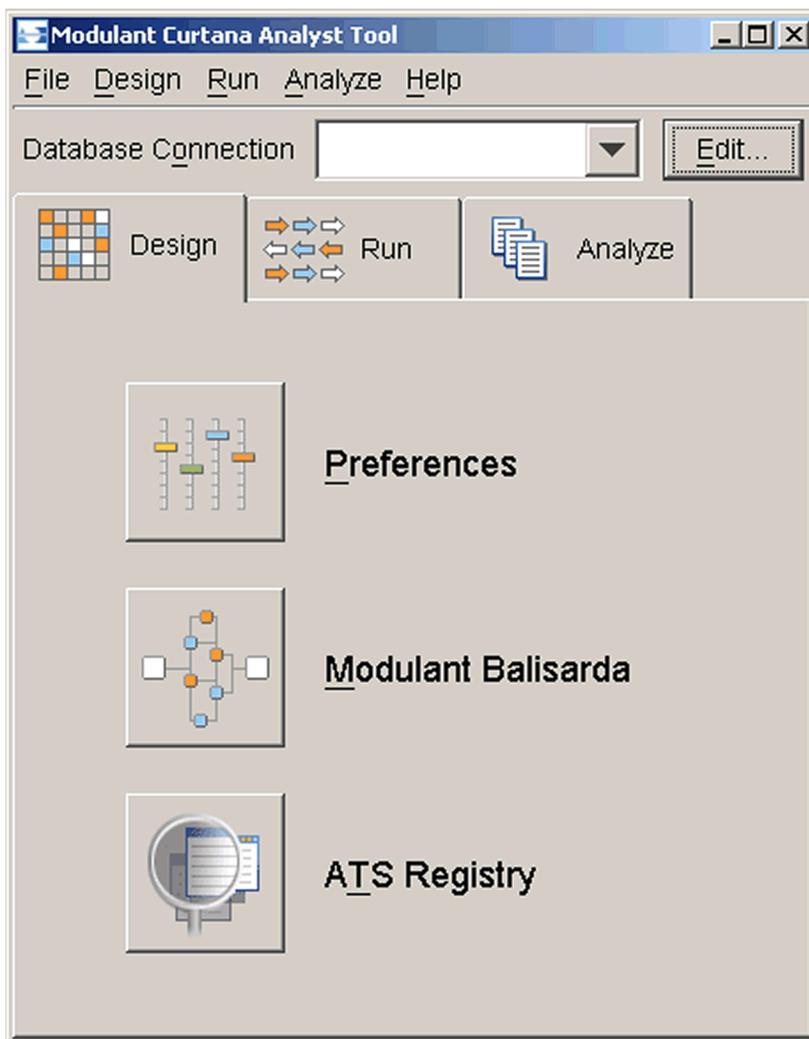
## Start the Modulant Curtana Analyst Tool

To begin creating a mapping, you must start the Analyst Tool, as follows:

- ▶ From the Windows desktop, select **Start > Programs > Modulant > Modulant Curtana**.

### 3 Creating a Sample Mapping

The Modulant Curtana Analyst Tool main window appears:



From here, you will start the Modulant Balisarda Mapping Tool. Before you can create a mapping, you must specify connection information for the internal database that the Mapping Tool uses to store information as you work. The Mapping Tool will use this information to generate an XMP mapping specification.

## Specify a Database Connection

To specify a database connection:

- 1 At the top of the Analyst Tool window, next to **Database Connection**, click **Edit**.

The **Edit Database Connection** dialog box appears:

The screenshot shows the 'Modulant Curtana Analyst Tool: Edit Database Connection' dialog box. It features a table at the top with columns for 'Name', 'Database ...', 'Database', 'Connectio...', 'user ID', and 'Password'. Below the table, the 'Database Connection' section includes a 'Name' text field, 'Database Type' radio buttons (selected for 'Access', unselected for 'Oracle'), a 'Database' text field with a 'Browse...' button, a 'Connection String' text field, a 'User ID' text field, and a 'Password' text field. At the bottom, there are buttons for 'Add', 'Modify', 'Delete', 'Clear', and 'Close'.

- 2 In the **Name** field, type a name to represent your database. For this example, type **walkthrough**.
- 3 Click **Browse** to search for the database file. For this example, use the Access database **mt-walkthrough.mdb** you created in the **samples** subdirectory of your Modulant Curtana installation.
- 4 Leave the **User ID** and **Password** fields blank.
- 5 Click **Add** to add the connection information to the list at the top.
- 6 Click **Close**.

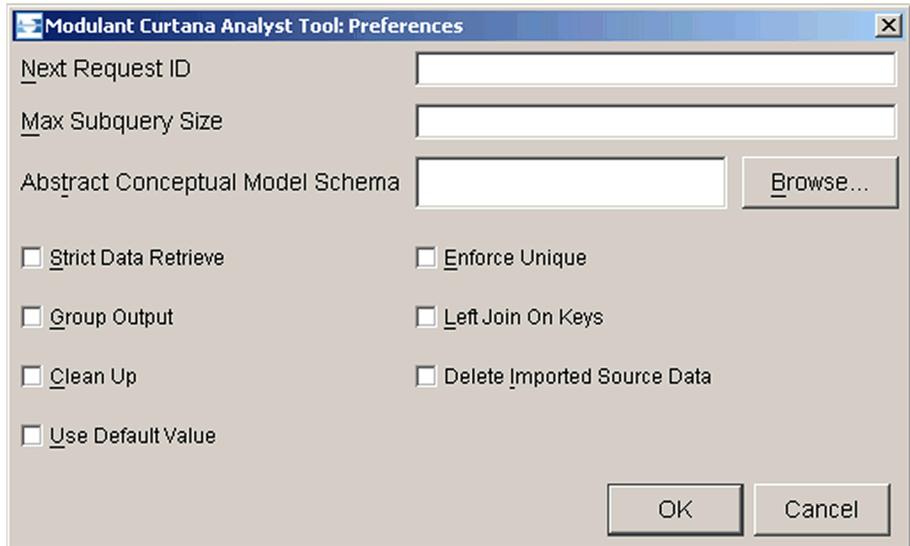
Make sure that the **walkthrough** database connection you defined is now selected in the **Database Connection** field.

## Specify the Abstract Conceptual Model's Location

To specify the location of the Abstract Conceptual Model schema file:

- 1 On the **Design** tab, click the **Preferences** button.

The **Preferences** dialog box appears:



- 2 In the **Next Request ID** field, type 2.

The **Request ID** corresponds to a transformation run that you have defined.

- 3 In the **Max Subquery Size**, type 3.

This number tells the Analyst Tool how to define queries for population and extraction during a transformation run.

- 4 Next to the **Abstract Conceptual Model Schema** field, click **Browse** to locate and select the file **Abstract\_Conceptual\_Model\_01.exp** in the **schema** directory of your Modulant Curtana installation.

- 5 Select the **Group Output**, **Enforce Unique**, and **Left Join On Keys** check boxes and ensure that the remaining check boxes are cleared.

- 6 Click **OK** to save your settings.

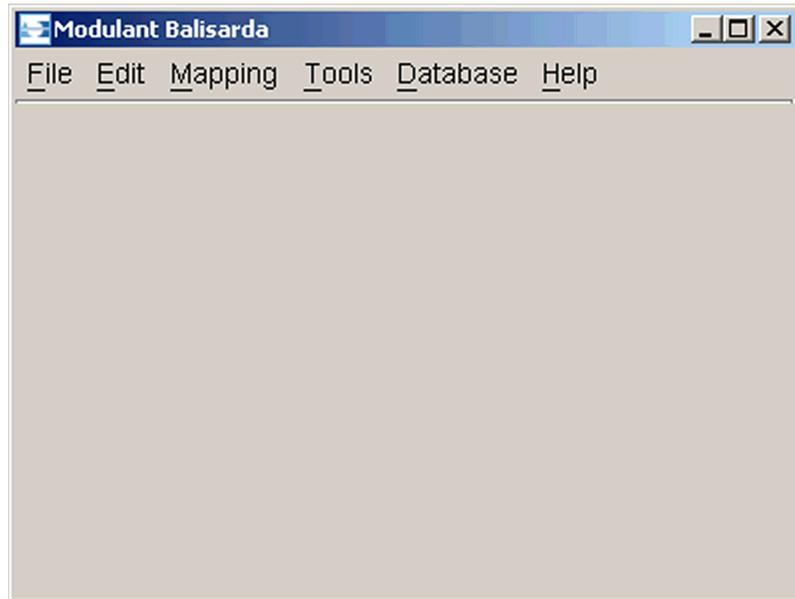
For more information about setting preferences for the Analyst Tool, see the *Modulant Curtana Analyst Tool Guide*.

## Start the Modulant Balisarda Mapping Tool

Now you can start the Mapping Tool:

- ▶ On the **Design** tab, click the **Modulant Balisarda** button.

The Modulant Balisarda Mapping Tool starts in its own window:



## Importing the ATS Schema

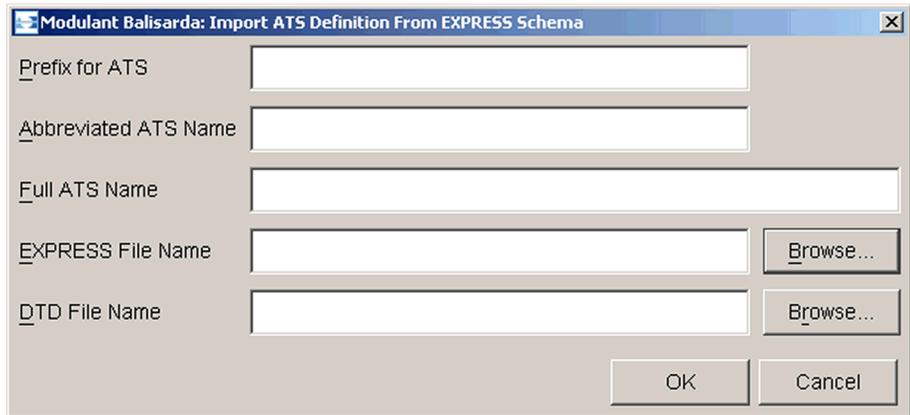
Before you can begin creating a mapping, you must tell the Mapping Tool which ATS schema you want to map. The easiest way to do this is to import an EXPRESS schema file. For more information about importing ATS schema information in a variety of formats, see [Chapter 6, "Importing ATS Schema Information."](#)

**Note:** The ATS schema you will use in this example is in EXPRESS format. You can generate an EXPRESS schema by creating an EXPRESS-G diagram using FirstSTEP XG, or if you are familiar with the EXPRESS language, you can create one using a text editor. You can generate an XML DTD of the required format from your EXPRESS schema using FirstSTEP EXML. To obtain a copy of either of these tools, write to [info@pdit.com](mailto:info@pdit.com).

To import the ATS schema for the AcmeAuto database:

- 1 Select **File > Import > ATS Definition > From EXPRESS Schema**.

The **Import ATS Definition From EXPRESS Schema** dialog box appears:



- 2 In the **Prefix for ATS** field, type `ACME`.

The Mapping Tool prefixes this value to entries for this ATS in its internal database and in the mapping specification.

**Note:** Values for table prefixes are case-sensitive. Modulant recommends that you use all upper-case letters for table prefixes.

- 3 In the **Abbreviated ATS Name** field, type `ACME`.

The Mapping Tool uses this value as the name of a *data element group*. Data element groups identify the data elements in each ATS schema. When you create mappings for more than one ATS schema, the group name is an easy way to distinguish which data elements you are working with at any time.

- 4 In the **Full ATS Name** field, type `ACMEAUTO`.

The Mapping Tool uses this name to identify the ATS schema associated with the AcmeAuto database.

- 5 To locate the EXPRESS schema file:

- a Click **Browse** next to the **EXPRESS File Name** field and navigate to the location of the EXPRESS schema for the AcmeAuto database. For this example, use the file `acmeauto.exp` in the `samples\acmeauto` directory of your Modulant Curtana installation.
- b Select the EXPRESS schema file and click **Open**.

- 6 To locate the DTD associated with the EXPRESS schema file:

- a Click **Browse** next to the **DTD File Name** field and navigate to the location of the DTD file for the AcmeAuto database. For this example, use the file `acmeauto.dtd` in the **DTD** directory of your Modulant Curtana installation.
- b Select the DTD file and click **Open**.

## 7 Click OK.

The Mapping Tool imports the schema information and constructs a list of data elements in your ATS schema. It uses the abbreviation you provided for the ATS name, in this case **ACME**, to create a *data element group* with all of the elements in the ATS schema you just imported.

**Note:** The Mapping Tool imports this information without informing you when the import operation is complete.

Now you are ready to map the data elements in your ATS schema.

---

## Mapping Data Elements

The process of mapping data elements from an ATS schema includes:

- Mapping the Root Element
- Mapping Related Data Elements

For more information about these stages, see [Chapter 7, “Mapping Data Elements.”](#)

## Mapping the Root Element

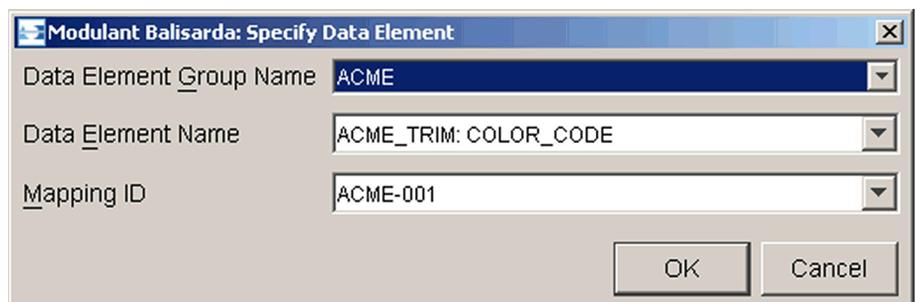
To start your mapping, you map the root data element first. The root element is one that connects a group of other data elements, usually the key field of a database table.

The mapping process expects a root element for each database table. In this case, you start with the key of the **ACME\_TRIM** table, which is **TRIM\_NBR**. According to [Table 3 on page 23](#), you map this root element to **product.id**.

To map the root element:

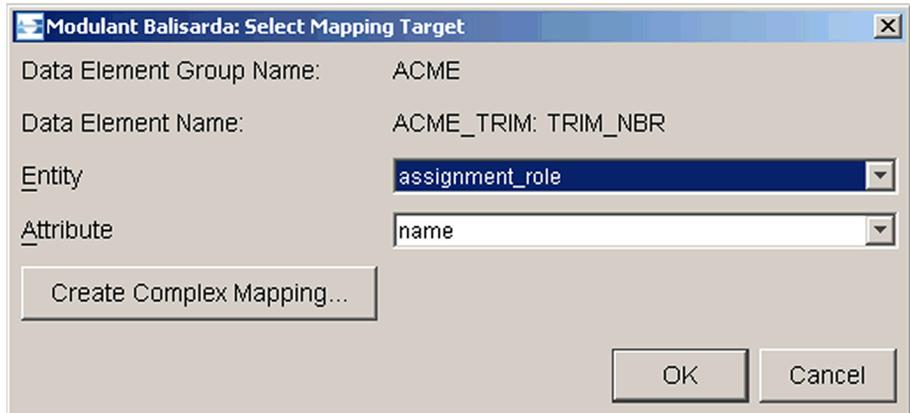
- 1 Select **Mapping > Add > Mapping Target**.

The **Specify Data Element** dialog box appears. By default, the **Data Element Name** field shows the first data element (listed alphabetically, starting with **COLOR\_CODE**) in your source ATS. Notice that the **Data Element Group Name** shows **ACME**.



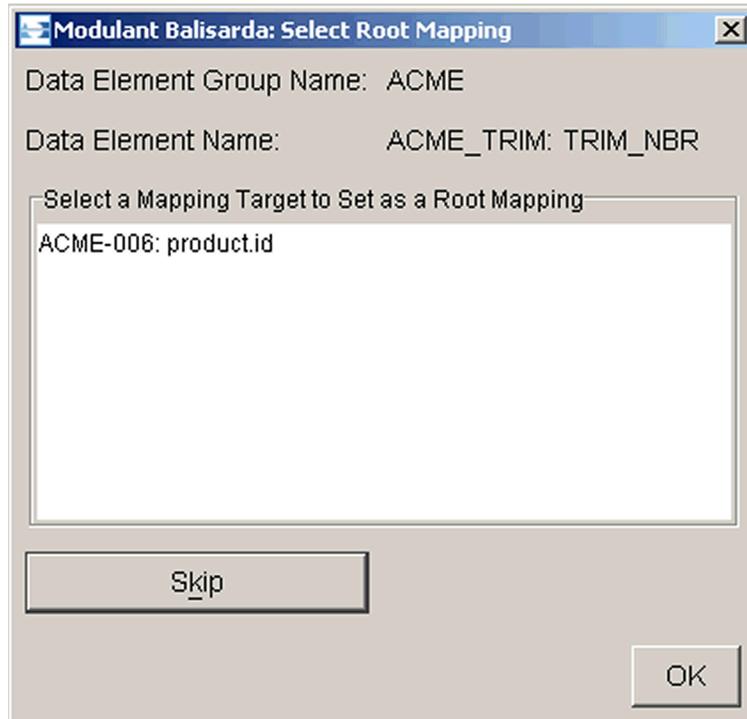
### 3 Creating a Sample Mapping

- From the **Data Element Name** drop-down list, select **ACME\_TRIM: TRIM\_NBR**.  
The value of the **Mapping ID** field changes to show the mapping ID of the data element you selected.
- Click **OK** to continue to the next step.  
The **Select Mapping Target** dialog box appears, showing the data element you just selected:



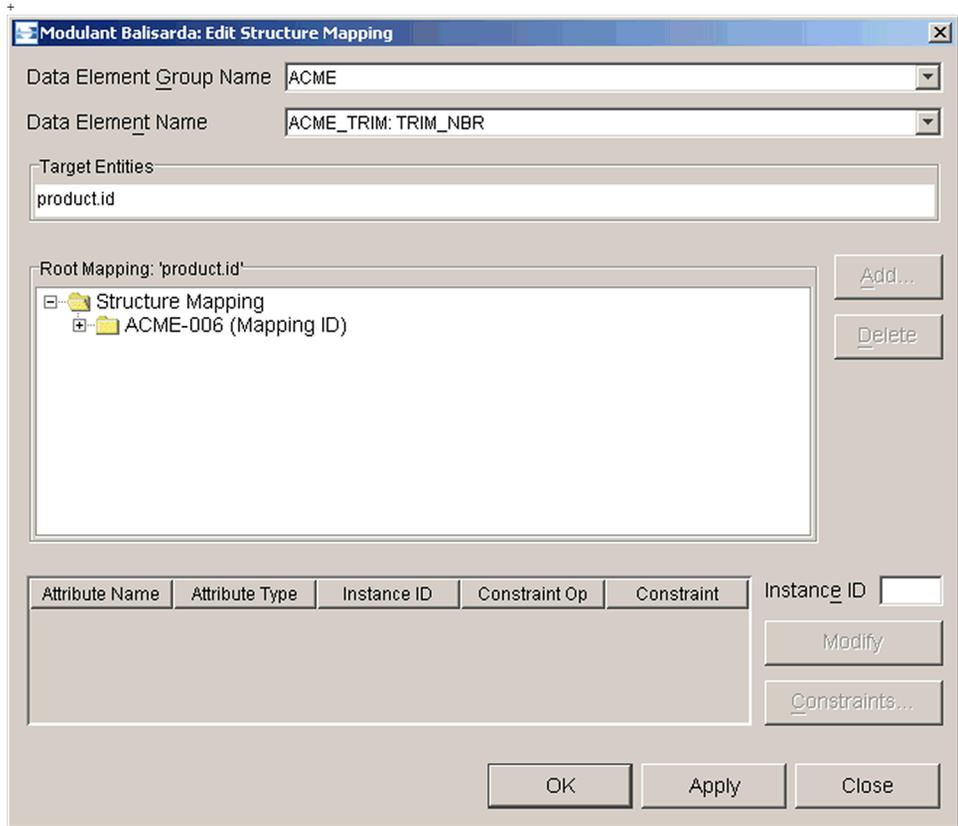
- From the **Entity** drop-down list, select **product**.  
When you select a target entity, the **Attribute** list shows only the attributes of that entity.
- From the **Attribute** drop-down list, select **id**.
- Click **OK** to map **ACME\_TRIM: TRIM\_NBR** to **product.id**.

The **Select Root Mapping** dialog box appears, showing the mapping target you selected:

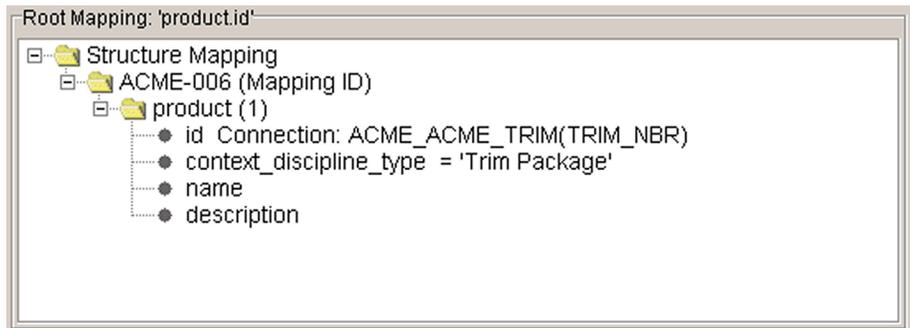


- 7 Click **OK** to identify **TRIM\_NBR** and its mapping target as a root mapping for the **ACME\_TRIM** table.

The **Edit Structure Mapping** dialog box appears:



- 8 Expand all of the nodes of the tree structure under **Acme-006 (Mapping ID)**. The expanded view shows you the path to the mapping target you selected, known as the *structure mapping*:



Notice that the product entity has an attribute called **context\_discipline\_type**. You use this attribute to specify the context of the product, so that it can be preserved through the transformation process. In other words, to more clearly identify this product, you provide a text string as the value of **context\_discipline\_type** that says what this product is about. This information is known as a *domain constraint*.

- 9 To specify the context of this product:
  - a In the structure mapping, select **context\_discipline\_type**.  
The Mapping Tool fills in the table at the bottom of the dialog box with the names of the attributes of **product** and their types. You use this information to specify domain constraints.
  - b In the table at the bottom, select the **Constraint** cell of the **context\_discipline\_type** row.
  - c Click **Constraints**.

The **Add Constraints** dialog box appears:

The screenshot shows a dialog box titled "Modulant Balisarda: Add Constraints". At the top, it displays "Entity Name: product" and "Attribute Name: context\_discipline\_type". Below this is a large, empty rectangular area labeled "Constraints". Underneath the constraints area are two buttons: "Show All Constraints" and "Show Attribute Constraints". At the bottom of the dialog is a section titled "New Constraint Properties". This section contains an "Add Selected Constraint" button, a "Constraint Operator" dropdown menu currently set to "=", and an empty "Constraint" text input field. At the very bottom right of the dialog are "OK" and "Cancel" buttons.

- d** Click **Show Attribute Constraints** to see a list of context strings you have already entered for this attribute.  
Because this will be your first context string, the list is empty at this point. This list will become important later in the mapping process.
- e** In the **Constraint** field, type `Trim Package` to identify the meaning of the **ACME\_TRIM** table.  
You can enter up to 255 characters for a domain constraint string. Shorter strings produce more efficient data transformations, but make sure to use a string that is long enough to convey all of the necessary information.
- f** Click **OK**.  
The structure mapping now shows the context string you entered.

- 10** In the **Edit Structure Mapping** dialog box, click **OK** to accept the structure mapping.
- You will see more examples of structure mappings as you map the remaining data elements of the AcmeAuto database.

**Tip:** You can select **Edit > Structure Mapping** to go back and change the structure mapping at any point in your mapping.

## Mapping Related Data Elements

Once you have mapped the root element, you can map the elements related to that root. In a relational database, those are the other elements in the same table. It does not matter what order you use to map these elements.

The next step of this example is to map the remaining elements of the **ACME\_TRIM** table: **TRIM\_NAME**, **TRIM\_COLOR**, **COLOR\_CODE**, **T\_MAKE**, and **T\_MODEL**.

### Mapping TRIM\_NAME

To map **TRIM\_NAME**:

- 1** Select **Mapping > Add > Mapping Target**.
- 2** In the **Specify Data Element** dialog box, select **ACME\_TRIM: TRIM\_NAME**, and click **OK**.

Notice that **ACME\_TRIM: TRIM\_NBR** no longer appears in the **Data Element Name** list.

- 3** In the **Specify Mapping Target** dialog box, select the mapping target **product.name** and click **OK**.

**Tip:** Select **product** from the **Entity** list, and then select **name** from the **Attribute** list.

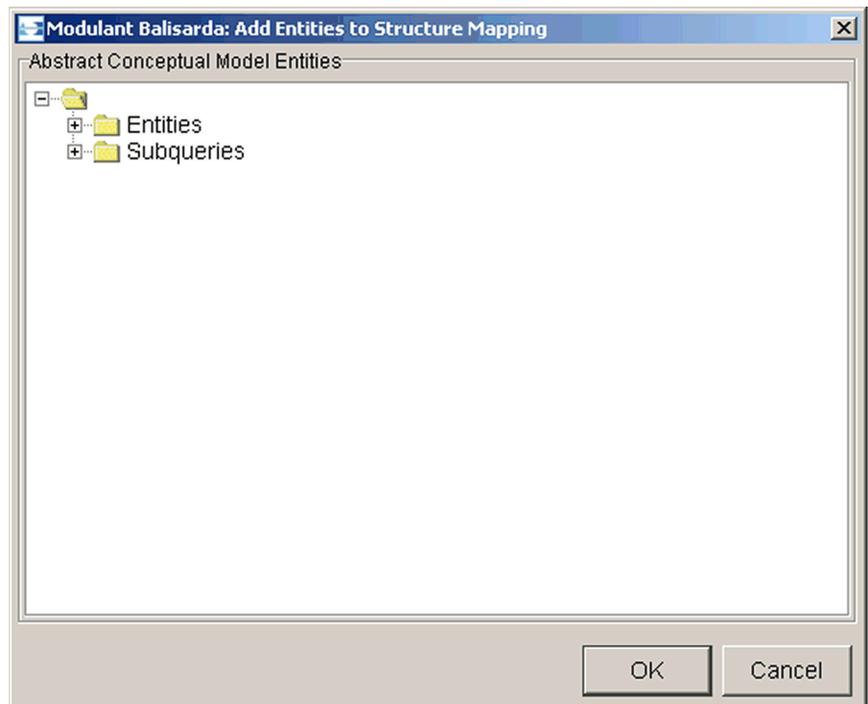
- 4** In the **Edit Structure Mapping** dialog box, expand the structure mapping to show **product** and its attributes.

Because the data element **TRIM\_NAME** refers to the same product as **TRIM\_NBR**, you want to connect the **product** entity in this structure mapping to the **product** entity in the structure mapping you did for **TRIM\_NBR**. To do this, you apply the same domain constraint to **context\_discipline\_type**.

- 5** To apply the domain constraint:
  - a** Select the attribute **context\_discipline\_type**.
  - b** In the table at the bottom, select the **Constraint** cell of the **context\_discipline\_type** row.
  - c** Click **Constraints**.



The **Add Entities to Structure Mapping** dialog box appears:



- c Under **Abstract Conceptual Model Entities**, expand **Entities**, then expand **management data associations**.  
This folder lists all association entities related to the entities that appear in the structure mapping. For more information about association entities, see [“Management Resource Associations”](#) on page 88.
- d Select **product\_group\_assignment**.
- e Click **OK** to add this entity to the structure mapping.

Now you have a relationship between **group** and **product**, which is what you needed. Notice that the Mapping Tool added an entity you did not explicitly select: **assignment\_role**.

The assignment entities are all directly related to **assignment\_role**. This entity has a **name** attribute that plays a critical part in establishing context for the data elements you map. For more information about this entity, see [“Assignment Roles”](#) on page 89.

- 6 Expand **assignment\_role**.
- 7 Select the **name** attribute of **assignment\_role**.
- 8 Following the instructions in [step 9 on page 35](#), enter Trim Color as a domain constraint value for the **name** attribute.

**Tip:** You can usually use the column name of the data element you are mapping for **assignment\_role.name**, unless you need a more specific string. If you have two columns with different names that mean the same thing, however, be sure to use the same string for both.

- 9 Follow the instructions in [step 5 on page 37](#) to specify 'Trim Package' as a domain constraint for **product.context\_discipline\_type**.
- 10 Click **OK** to finish the structure mapping for **TRIM\_COLOR**.

### Mapping COLOR\_CODE

To map **COLOR\_CODE**:

- 1 Select **Mapping > Add > Mapping Target**.
- 2 In the **Specify Data Element** dialog box, select **ACME\_TRIM: COLOR\_CODE**, and click **OK**.

Remember that color codes provide a way to classify the colors assigned to trim packages, and that the **group** entity in the Abstract Conceptual Model represents classification systems.

- 3 In the **Select Mapping Target** dialog box, select the mapping target **group.name** and click **OK**.
- 4 In the **Edit Structure Mapping** dialog box, expand the structure mapping to show **group** and its attributes.

As you did when you mapped **TRIM\_COLOR**, you need to connect **group** to **product**.

- 5 Following the instructions in [step 5 on page 38](#), add **product\_group\_assignment** to the structure mapping.
- 6 Expand **assignment\_role**.
- 7 Following the instructions in [step 5 on page 37](#), to specify 'Trim Color' as a domain constraint value for **assignment\_role.name**.

The trim color provides a classification of the product. The color code provides a classification of trim colors, but is not directly related to products.

- 8 Follow the instructions in [step 5 on page 37](#) to specify 'Trim Package' as a domain constraint for **product.context\_discipline\_type**.

Now you need to create a relationship between the two groups: color codes and trim colors. Trim colors will become a subgroup of color codes.

- 9 To relate the color code group to the trim color group:
  - a Select **ACME-001 (Mapping ID)** again and click **Add**.
  - b Expand **Entities**.
  - c Under **management data resources**, add two entities: **group** and **group\_relationship** to the structure mapping. Notice that you are adding a second **group** entity to your mapping with this step.

**Tip:** Use **CTRL+CLICK** to select more than one entity.

- d** Select and expand **group\_relationship**.
  - e** In the table at the bottom, change the value of the **Instance ID** cell for **related\_group** to 2, and then click **Modify**.  
This specifies the trim color group as the child group of the relationship, the color code group as the parent.
  - f** Select **group\_relationship.name** and click **Constraints** to specify a name for the group.
  - g** Type `Color classification` as a domain constraint to specify the name of the group relationship.
  - h** Select and expand **product\_group\_assignment**.
  - i** In the table at the bottom, set the **Instance ID** of **assigned\_group** to 2, and then click **Modify**.  
This relates the trim color group back to the product entity.
- 10** In the **Edit Structure Mapping** dialog box, click **OK** to finish the structure mapping for **COLOR\_CODE**.

## Mapping T\_MAKE

To map **T\_MAKE**:

- 1** Select **Mapping > Add > Mapping Target**.
- 2** In the **Specify Data Element** dialog box, select **ACME\_TRIM: T\_MAKE**, and click **OK**.  
The make of a car indicates who manufactured the car. Because the manufacturer of a car can be considered as an organization, you will use the **organization** entity in the Abstract Conceptual Model for this mapping.
- 3** In the **Select Mapping Target** dialog box, select the mapping target **organization.identifier** and click **OK**.
- 4** In the **Edit Structure Mapping** dialog box, expand the structure mapping to show **organization** and its attributes.  
You must add an assignment entity that connects products with organizations. In this case, you will use **product\_organization\_assignment** to connect the current mapping target to the root. For more information about this entity, see [“Product and Organization” on page 96](#).
- 5** Select **ACME-002 (Mapping ID)** and click **Add**.
- 6** Following the instructions in [step 5 on page 38](#), add **product\_organization\_assignment** (in the **management data associations** folder) to the structure mapping.  
Once again, the Mapping Tool added **assignment\_role** to the structure mapping.
- 7** Following the instructions in [step 9 on page 35](#), enter **Make** as a new domain constraint value for **assignment\_role.name**.
- 8** Follow the instructions in [step 5 on page 37](#) to select the existing value **‘Trim Package’** as a domain constraint for **product.context\_discipline\_type**.

- 9 Click **OK** to finish the structure mapping for **T\_MAKE**.

### Mapping T\_MODEL

To map **T\_MODEL**:

- 1 Select **Mapping > Add > Mapping Target**.
- 2 In the **Specify Data Element** dialog box, select **ACME\_TRIM: T\_MODEL**, and click **OK**.

The model of a car provides a way of grouping cars from a single manufacturer. Therefore, you will once again use **group** as the mapping target. Notice that each of the groups you have defined (trim colors, color codes, and now models) represents a different classification schema, and are therefore different groups.

- 3 In the **Select Mapping Target** dialog box, select the mapping target **group.name** and click **OK**.
- 4 In the **Edit Structure Mapping** dialog box, expand the structure mapping to show **group** and its attributes.

As you did when you mapped **TRIM\_COLOR** and **COLOR\_CODE**, you need to connect **group** to **product**.

- 5 Following the instructions in [step 5 on page 38](#), add **product\_group\_assignment** (in the **management data associations** folder) to the structure mapping.

Notice that the Mapping Tool again added **assignment\_role** to the structure mapping.

- 6 Following the instructions in [step 9 on page 35](#), enter `Model` as a new domain constraint value for **assignment\_role.name**.
- 7 Follow the instructions in [step 5 on page 37](#) to select the existing value 'Trim Package' as a domain constraint for **product.context\_discipline\_type**.
- 8 Click **OK** to finish the structure mapping for **T\_MODEL**.

## Mapping the ACME\_TRIM\_IMAGE Table

You have now mapped the first table in the AcmeAuto database, **ACME\_TRIM**. Mapping the elements in the **ACME\_TRIM\_IMAGE** table follows the same process. Use the information in Table 4 to map the elements in **ACME\_TRIM\_IMAGE**.

**Table 4: Mapping the ACME\_TRIM\_IMAGE Table**

Source Element	Mapping Target	Notes
ACME_TRIM_IMAGE: TRIM_NBR	product.id	<p>This is the root element of the <b>ACME_TRIM_IMAGE</b> table. When you map this element, you must indicate that the target <b>product</b> entity is the same as the <b>product</b> that is the root of the <b>ACME_TRIM</b> table.</p> <p>To do this, you set <b>product.context_discipline_type</b> to the same string you used before (<b>'Trim Package'</b>). For instructions, see <a href="#">step 9 on page 35</a>.</p>
ACME_TRIM_IMAGE: TRIM_NAME	product.name	<p>Again, you must set the value of <b>context_discipline_type</b> to indicate that this refers to the same product as <b>TRIM_NBR</b>.</p>
ACME_TRIM_IMAGE: IMAGE_TYPE	document_type. product_data_type	<p>In the structure mapping, add the <b>document</b> entity first (from the <b>management data resources</b> folder), and then add the assignment entity <b>product_document_reference</b> (from the <b>management data associations</b> folder) to connect documents to products. Enter <b>Image Type</b> as the value of <b>assignment_role.name</b>.</p> <p>Set the value of <b>product.context_discipline_type</b> to <b>'Trim Package'</b> as you did before.</p>

**Table 4: Mapping the ACME\_TRIM\_IMAGE Table**

Source Element	Mapping Target	Notes
ACME_TRIM_IMAGE: T_IMAGE	document.id	Add the assignment entity <b>product_document_reference</b> (from the <b>management data associations</b> folder). Enter <code>Image File</code> as the value of <b>assignment_role.name</b> .  Set the value of <b>product.context_discipline_type</b> to <b>'Trim Package'</b> as you did before.
ACME_TRIM_IMAGE: TRIM	skip this one	This attribute actually refers back to the <b>ACME_TRIM</b> table, which you have already mapped.

At this point, you have finished mapping the source data elements. If you select **Mapping > Add > Mapping Target**, the only data element left on the list is the one you did not map, **ACME\_TRIM\_IMAGE.TRIM**. For more information about structure mappings and the structure mapping process, see [Chapter 7, "Mapping Data Elements."](#)

Now you are ready for the next step: defining the transformation.

## Defining the Transformation

So far, you have mapped each data element individually and specified the structure mapping for each one. The next step is to specify how to transform the data associated with each element, based on the structure mappings. To do this, you connect database fields to Abstract Conceptual Model attributes in the structure mapping for each data element.

This process helps consolidate the information about the structure of the data represented by your schema. By connecting the same data element with the same attributes in each structure mapping, you specify how you want the Transformation Engine to populate the Abstract Conceptual Model with input data and how you want it to extract output data from the Abstract Conceptual Model.

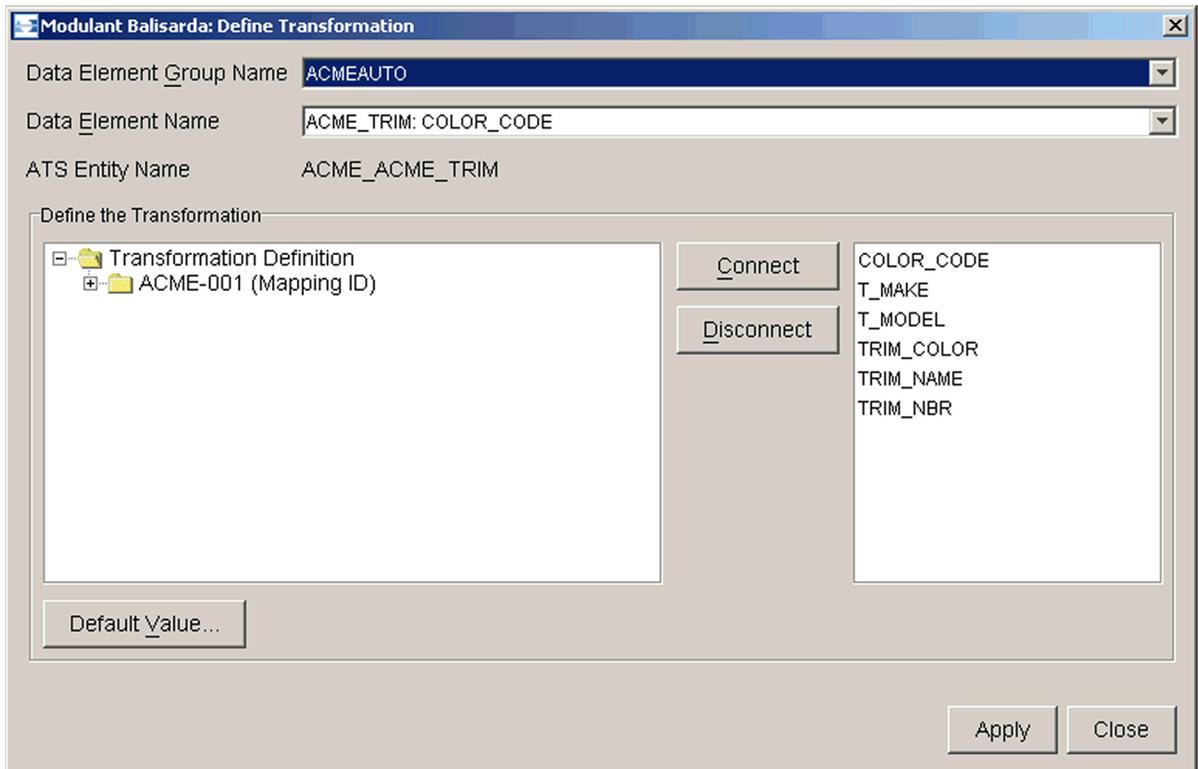
For example, you mapped **ACME\_TRIM: TRIM\_NBR** to **product.id** and **ACME\_TRIM: TRIM\_NAME** to **product.name**. When you use the Analyst Tool to perform transformation runs, you want the values of both **TRIM\_NBR** and **TRIM\_NAME** to always map to the same data fields, rather than creating duplicate copies of identical data, in different **product** entities. Defining the transformation ensures that your data comes out the way you expect it to.

## Starting to Define the Transformation

To define the transformation:

- 1 Select **Tools > Define Transformation**.

The **Define Transformation** dialog box appears:



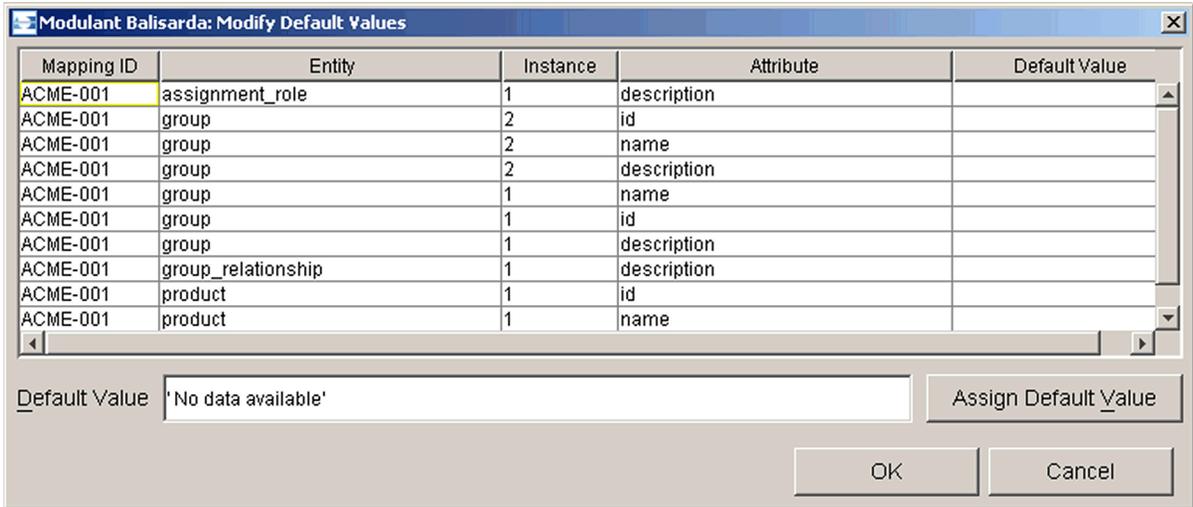
At the top is a drop-down list of the data element names you mapped. To define a complete transformation, you connect data elements from the list of data elements on the right to attributes in the structure mapping on the left. When no data element matches an attribute, you can specify a text string as the default value to use for that attribute in the transformation process.

You can do this in any order. For simplicity, in this walkthrough, you will work on the elements in alphabetical order.

- 2 Expand all of the nodes in the structure mapping for the data element **ACME\_TRIM: COLOR\_CODE**.
- 3 In the structure mapping, select the **name** attribute of **group(2)**.
- 4 In the list of data elements, select **TRIM\_COLOR**.

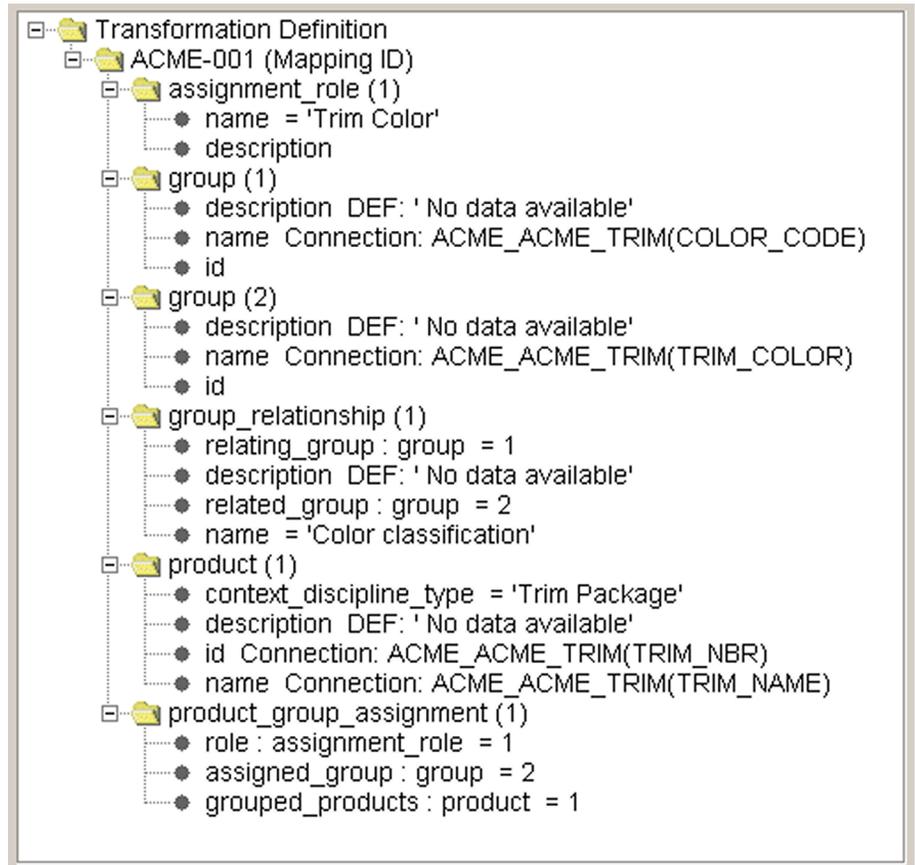
### 3 Creating a Sample Mapping

- 5 Click **Connect** to map the selected elements to each other.  
The structure mapping now shows the connected elements.  
Because none of the source data elements provide further descriptive information about either group of colors, you supply a string to use as the default value when the Transformation Engine populates the Abstract Conceptual Model from the AcmeAuto database.
- 6 To specify a default value:
  - a Select the first **group.description**, and click **Default Value**.  
The **Modify Default Values** dialog box appears:



- b Select the first row that contains **group.description**.
  - c Click **Assign Default Value**, to use the string '**No data available**' as the default value for this field.
- 7 Select the rows that contain the second **group.description**, **group\_relationship.description**, and **product.description**, and click **Assign Default Value** to assign the same default value to each of them.  
**Tip:** Use CTRL+CLICK to select more than one entity.
- 8 Click **OK** to return to the **Define Transformation** dialog box.  
The structure mapping now shows the default values you specified.
- 9 Connect **product.name** to **TRIM\_NAME**.

When you are done, the structure mapping should look like this:



10 Click **Apply** to save your changes.

## Continuing to Define the Transformation

You must repeat this process for the structure mapping for each data element you mapped.

To continue the process:

- 1 From the **Data Element Name** drop-down list, select the next data element, in this case, **ACME\_TRIM: T\_MAKE**.
- 2 Expand all of the nodes of the structure mapping.
- 3 Connect the elements as follows:

<b>organization.name</b>	<b>T_MAKE</b>
<b>product.name</b>	<b>TRIM_NAME</b>

### 3 Creating a Sample Mapping

- 4 Set the text string **'No data available'** as the default value for the following elements:
  - ◆ **organization.address**
  - ◆ **organization.description**
  - ◆ **product.description**
- 5 Select the next data element, **ACME\_TRIM: T\_MODEL**.
- 6 Click **Yes** to commit your changes.
- 7 Connect the elements as follows:

<b>group.id</b>	<b>T_MODEL</b>
<b>group.description</b>	<b>'No data available'</b>
<b>product.name</b>	<b>TRIM_NAME</b>
<b>product.description</b>	<b>'No data available'</b>

- 8 Connect **TRIM\_COLOR** as follows:

<b>group.description</b>	<b>'No data available'</b>
<b>product.name</b>	<b>TRIM_NAME</b>
<b>product.description</b>	<b>'No data available'</b>

- 9 Connect **TRIM\_NAME** as follows:

<b>product.description</b>	<b>'No data available'</b>
----------------------------	----------------------------

- 10 Connect **TRIM\_NBR** as follows:

<b>product.name</b>	<b>TRIM_NAME</b>
<b>product.description</b>	<b>'No data available'</b>

You have now completed the transformation definition for the **ACME\_TRIM** table.

- 11 Use the following tables to define the transformation for the **ACME\_TRIM\_IMAGE** table.

- a To connect **ACME\_TRIM\_IMAGE: TRIM\_NBR**:

<b>product.name</b>	<b>TRIM_NAME</b>
<b>product.description</b>	<b>'No data available'</b>

- b To connect **ACME\_TRIM\_IMAGE: TRIM\_NAME**:

<b>product.description</b>	<b>'No data available'</b>
----------------------------	----------------------------

- c To connect **ACME\_TRIM\_IMAGE: T\_IMAGE:**

<b>document.name</b>	<b>T_IMAGE</b>
<b>document.description</b>	<b>'No data available'</b>
<b>document_type.product_data_type</b>	<b>IMAGE_TYPE</b>
<b>product.name</b>	<b>TRIM_NAME</b>
<b>product.description</b>	<b>'No data available'</b>

- d To connect **ACME\_TRIM\_IMAGE: IMAGE\_TYPE:**

<b>document.id</b>	<b>T_IMAGE</b>
<b>document.name</b>	<b>T_IMAGE</b>
<b>document.description</b>	<b>'No data available'</b>
<b>product.name</b>	<b>TRIM_NAME</b>

- 12 Click **Apply**, and then click **Close** to save your changes and complete the process.

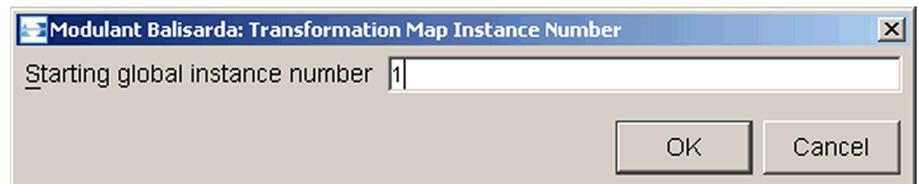
## Creating a Mapping Specification

After you finish defining the transformation, you are ready to generate the mapping specification. You will be able to use this specification with the Analyst Tool to perform transformation runs.

To create the mapping specification:

- 1 Select **Tools > Generate Transformation Map**.

The **Transformation Map Instance Number** dialog box appears:



- 2 Click **OK** to accept the default value of 1 as the starting global instance number.

This number is related to how you will use the mapping specification when you perform transformation runs. The Mapping Tool lets you know when the map has been successfully generated.

The Mapping Tool uses the mapping and transformation information you provided to generate a mapping specification, which you can export into an XMP (XML MaP) file. For more information about the format of XMP files, including the DTD, see [Appendix B, “XMP Reference.”](#)

### Optimizing a Mapping Specification

When you are satisfied that your mappings work the way you want them to, it is time to optimize the mapping specification. This will result in better performance by the Transformation Engine, and will minimize redundant data.

To optimize your mapping specification:

- ▶ Select **Tools > Optimize Transformation Map.**

The optimization process scans your mapping specification and consolidates references to the same data elements. Note that this process can take a while.

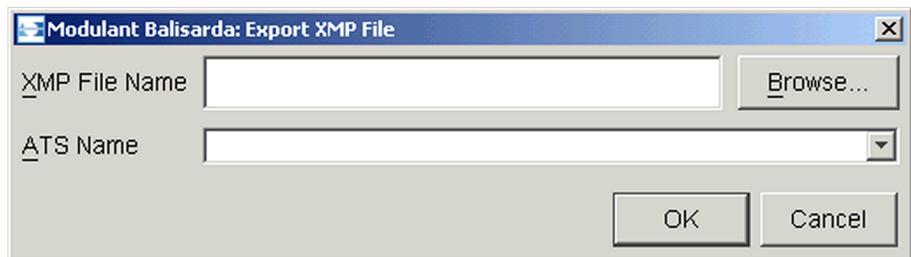
### Exporting an XMP File

To use the mapping specification with the Analyst Tool or the Lifecycle Manager API, you can export the mapping information from the Mapping Tool’s internal database to an XMP file in XML format.

To export an XMP file:

- 1 Select **File > Export > XMP.**

The **Export XMP File** dialog box appears:



- 2 Click **Browse** to specify the location where you want to save the XMP file, and specify a file name. Use the extension **.xml**.

**Tip:** Modulant recommends that you include the letters **xmp** in your file name to identify the file as a mapping specification; for example, **myfile-xmp.xml**.

- 3 From the **ATS Name** drop-down list, select the ATS schema whose mappings you want to save.
- 4 Click **OK**.

When the export operation is complete, click **OK** to continue.

**Tip:** The file `acme-xmp.xml` in the `samples\acmeauto` directory of your Modulant Curtana installation contains all of the mappings you created in this walkthrough.

You can use the generated XMP file with both the Analyst Tool and the Lifecycle Manager API. For more information, see the *Modulant Curtana Analyst Tool Guide* and the *Modulant Curtana Developer's Guide*.

---

## Testing a Mapping Specification

To test your mapping specification, use the Analyst Tool to do a transformation run using the AcmeAuto ATS as both the source and the target. To do this, you can use the same XMP file for both parts of the transformation.

The test includes the following steps:

- [Importing AcmeAuto Data](#)
- [Performing the Transformation Run](#)
- [Understanding the Results of the Transformation](#)
- [Cleaning Up the Internal Database](#)

## Importing AcmeAuto Data

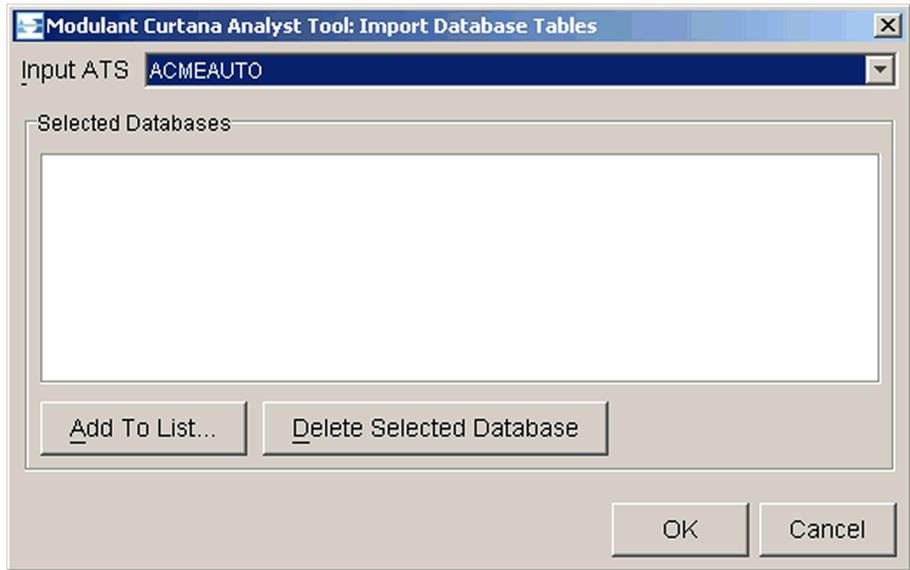
Before you can perform a transformation run, you must import the data from your ATS into the Modulant Curtana internal database. The AcmeAuto database is in the `samples\acmeauto` directory of your Modulant Curtana installation, in the file `acmeauto.mdb`.

To import data from the AcmeAuto database:

- 1 Select **File > Exit** to close the Mapping Tool.
- 2 In the Modulant Curtana Analyst Tool, select **File > Import Data > From Database Tables**.

### 3 Creating a Sample Mapping

The **Import Database Tables** dialog box appears:



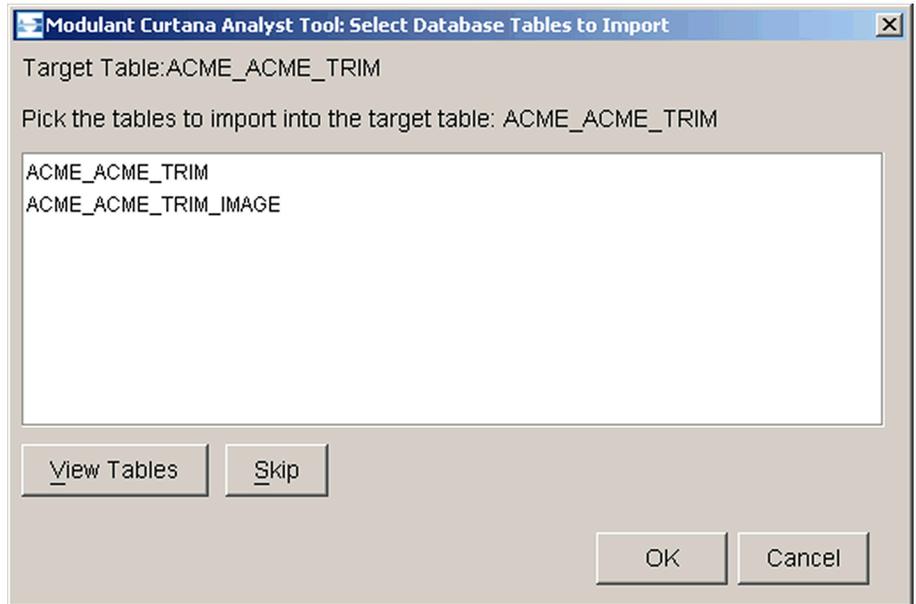
- 3 Click **Add To List** to browse for the location of your database. In this case, select **samples\acmeauto\acmeauto.mdb** in your Modulant Curtana installation, and click **Open**.

The name of the database you selected, and the full path to its location, appears in the **Selected Databases** list.

- 4 Click **OK** to continue.

The Analyst Tool reads the definition of your ATS schema and looks for tables in the selected database that match the names of ATS entities. In this case, the first ATS entity is **ACME\_ACME\_TRIM**, whose name could match both tables in the database.

The **Select Database Tables to Import** dialog box appears:



Because two tables in your database might match the ATS entity **ACME\_ACME\_TRIM** (listed at the top of the dialog box), you must specify which of these tables actually matches.

**Note:** If the Analyst Tool finds no ambiguity in the names of the tables in your database, it imports the data automatically without showing you this dialog box.

- 5 Select the **ACME\_ACME\_TRIM** table and click **OK**.

After importing the data from this table, the Analyst Tool finds that the next entity in the ATS definition is **ACME\_ACME\_TRIM\_IMAGE**. Because the database you selected has a table that matches this name exactly, the Analyst Tool imports the data from this table automatically.

- 6 When the import is complete, click **OK** to proceed.

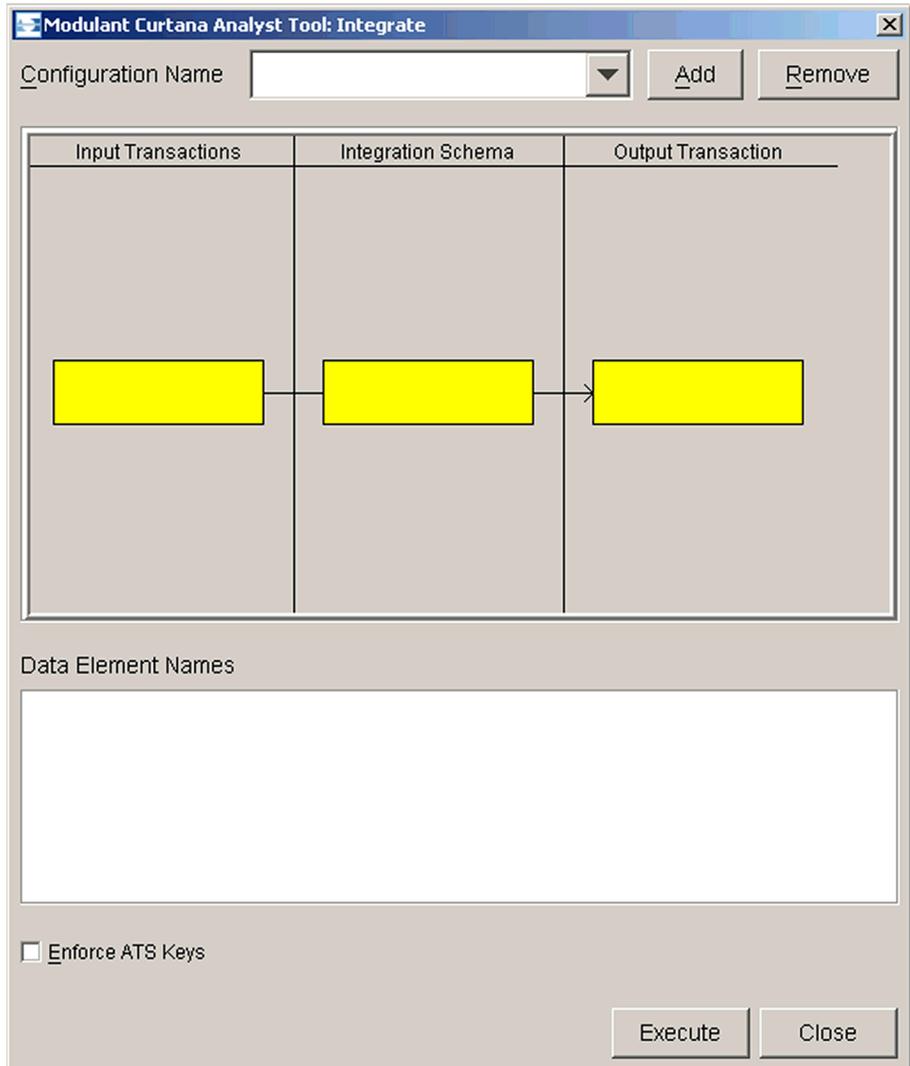
The data from both tables in the AcmeAuto database is now in the internal database, where it is ready for processing.

## Performing the Transformation Run

To do a test transformation run:

- 1 On the **Run** tab, click the **Integrate** button.

The **Integrate** dialog box appears:



- 2 To define a configuration name for this transformation run:
  - a At the top of the dialog box, click **Add**.

The **Add New Configuration** dialog box appears:

Modulant Curtana Analyst Tool: Add New Configuration

Integration Schema walkthrough

Configuration Name

Input Transactions

Output Transactions

Output Table Prefix

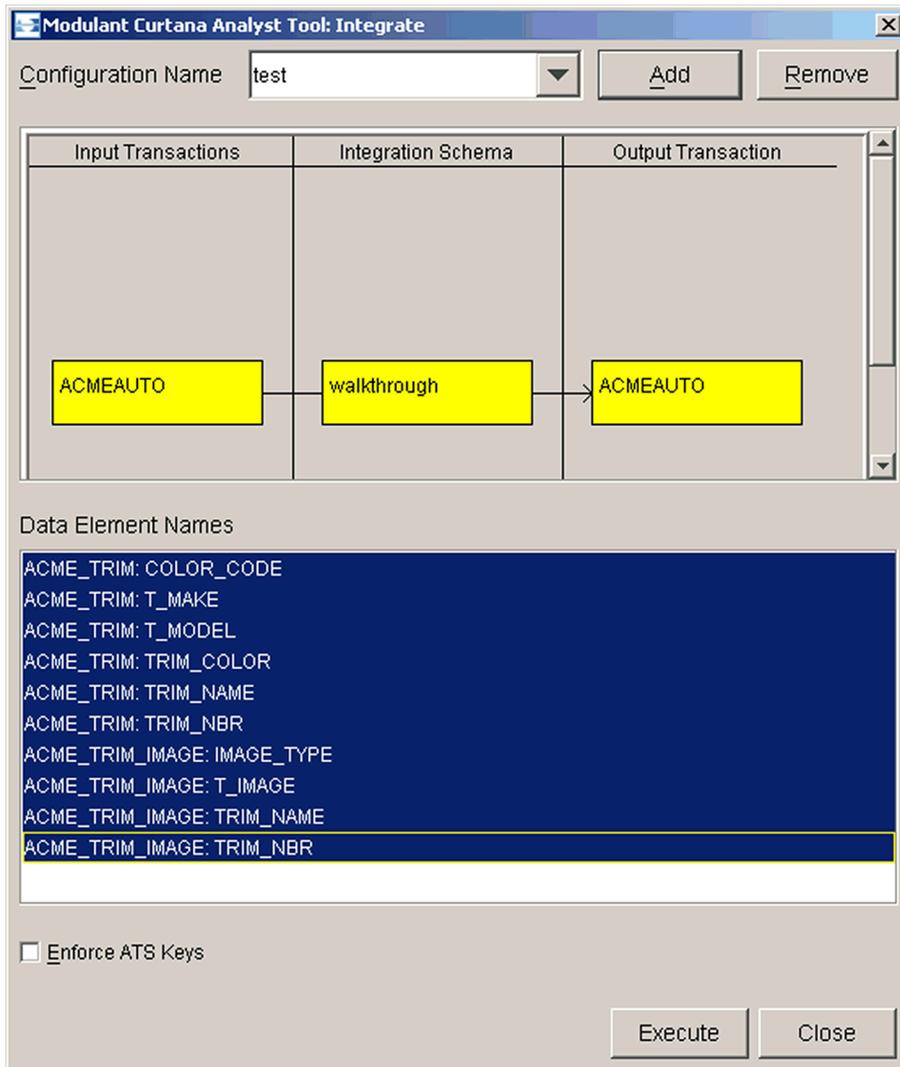
Output Query Prefix

OK Cancel

- b** In the **Configuration Name** field, type `test`.
  - c** Under both **Input Transactions** and **Output Transactions**, select **ACMEAUTO**.
  - d** Leave the **Output Table Prefix** and the **Output Query Prefix** fields blank.  
The Analyst Tool will use the existing prefixes for the ATs you selected.
  - e** Click **OK**.
- 3** In the **Integrate** dialog box:
  - a** Make sure **test** is selected from the **Configuration Name** list.  
The **Data Element Names** list at the bottom now shows any data elements in your output ATS (in this case `AcmeAuto`) that have been mapped and whose mapping usage is either **Output** or **Input/Output**.
  - b** Select all of the data elements in this list.  
The Transformation Engine will extract data for all of the data elements you select.

### 3 Creating a Sample Mapping

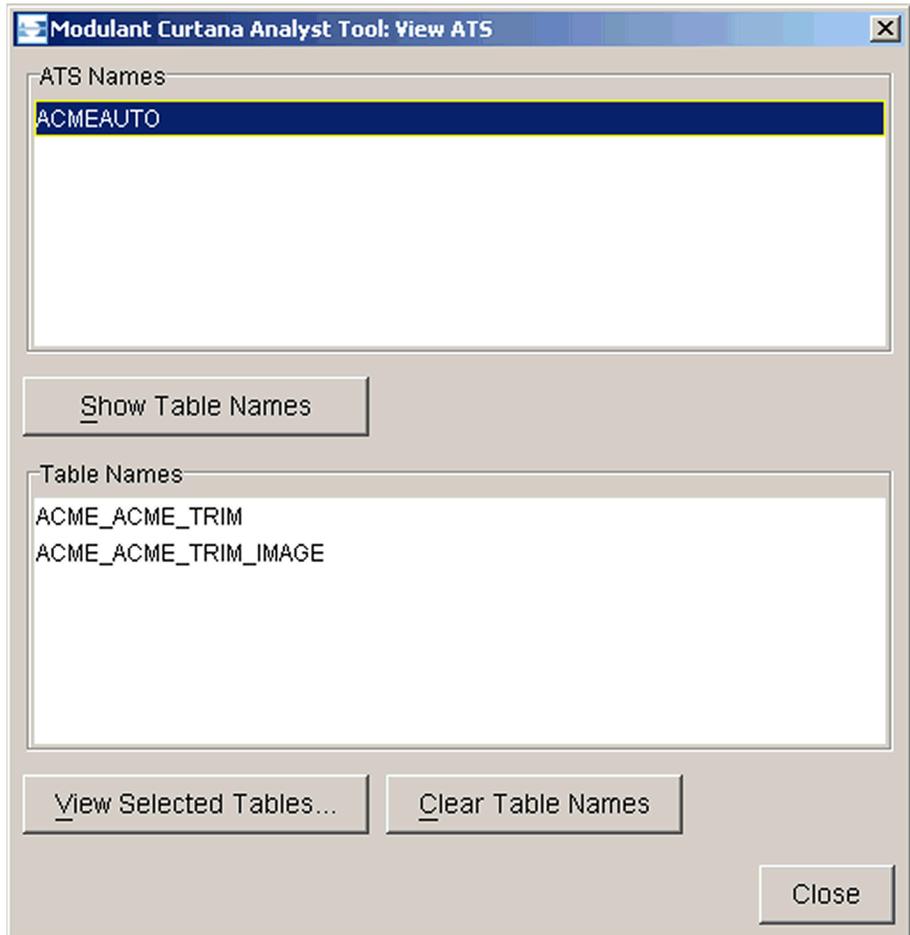
The **Integrate** dialog box now shows the configuration information you specified:



- 4 Click **Execute** to start the transformation run.

## Understanding the Results of the Transformation

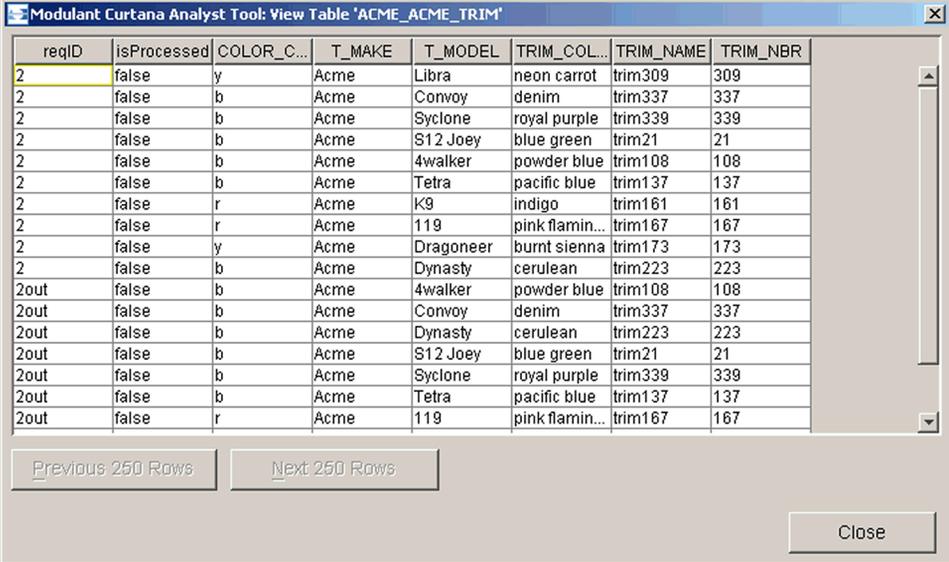
When the transformation run has finished, the Analyst Tool displays the **View ATS** dialog box.



- 1 To view the source and target data, which is still in the internal database:
  - a Under **ATS Names**, select **ACMEAUTO**.
  - b In the **Table Names** list, select **ACME\_ACME\_TRIM** and **ACME\_ACME\_TRIM\_IMAGE**.
  - c Click **View Selected Tables**.

### 3 Creating a Sample Mapping

A copy of the **View Table** dialog box appears for each table you selected, with the name of the table in the title bar:



reqID	isProcessed	COLOR_C...	T_MAKE	T_MODEL	TRIM_COL...	TRIM_NAME	TRIM_NBR
2	false	y	Acme	Libra	neon carrot	trim309	309
2	false	b	Acme	Convoy	denim	trim337	337
2	false	b	Acme	Syclone	royal purple	trim339	339
2	false	b	Acme	S12 Joey	blue green	trim21	21
2	false	b	Acme	4walker	powder blue	trim108	108
2	false	b	Acme	Tetra	pacific blue	trim137	137
2	false	r	Acme	K9	indigo	trim161	161
2	false	r	Acme	119	pink flamin...	trim167	167
2	false	y	Acme	Dragoneer	burnt sienna	trim173	173
2	false	b	Acme	Dynasty	cerulean	trim223	223
2out	false	b	Acme	4walker	powder blue	trim108	108
2out	false	b	Acme	Convoy	denim	trim337	337
2out	false	b	Acme	Dynasty	cerulean	trim223	223
2out	false	b	Acme	S12 Joey	blue green	trim21	21
2out	false	b	Acme	Syclone	royal purple	trim339	339
2out	false	b	Acme	Tetra	pacific blue	trim137	137
2out	false	r	Acme	119	pink flamin...	trim167	167

The **reqID** column shows the number of the transformation run. Entries with the number followed by **out** represent target ATS data—the result of the transformation.

- 2 To verify the results of the transformation:
  - a Check that you have the same number of rows of input data as you do of output data.
  - b Check each column in the table to see that wherever there is input data, there is corresponding output data and that the values match.

For more information about testing your mapping specification using the Analyst Tool in a transformation run, see the *Modulant Curtana Analyst Tool Guide*.

## Cleaning Up the Internal Database

If you want to perform the same transformation more than once, you must run the Clean Up Tool after each transformation run. For information about this tool, see the *Modulant Curtana Analyst Tool Guide*.