IBM VisualAge DataAtlas Multiplatform

Modeler User's Guide

Version 2.5

SC26-9041-01

IBM VisualAge DataAtlas Multiplatform

Modeler User's Guide

Version 2.5

SC26-9041-01

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page vii.

Second Edition (September 1997)

This edition applies to IBM VisualAge DataAtlas Mulitplatform, Version 2.5, Program Number 5648-A48, and to any subsequent releases and modifications until otherwise indicated in new editions or technical newsletters. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality.

If you have comments about this publication, please go to the DataAtlas Web site, select “Browse or participate in the DataAtlas threaded discussion group,” and add a comment to the “Online Information and Publications” thread. The Web site is located at *http://www.software.ibm.com/ad/datatlas*.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood NY 10594, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Deutschland Informationssysteme GmbH, Department 3982, Pascalstrasse 100, 70569 Stuttgart, Germany. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Trademarks

The following terms in this publication are trademarks of the IBM Corporation in the United States, or other countries, or both:

AIX	MVS
DataAtlas	OS/2
DB2	OS/390
DB2 Universal Database	TeamConnection
Electronic Showcase	VisualAge
IBM	XGA

Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation.

Other company, product, and service names may be trademarks or services of others.

About This Book

Read this book to find out how to use the Modeler component of IBM VisualAge DataAtlas Mulitplatform, referred to as DataAtlas Modeler in this book. DataAtlas Modeler is a data modeling tool that supports the entity-relationship approach for conceptual data modeling. It is one of three DataAtlas components; the others are DataAtlas Dictionary and DataAtlas Designer.

This book is intended for those people within an organization who are responsible for data modeling. Typically, these can be application analysts, application programmers, and database administrators.

You should know how to work with objects in OS/2 or Windows NT. You should also be familiar with the concepts of your installed relational database management system, and how to use it.

This book assumes you have installed and started DataAtlas on your desktop. Instructions for installing and starting DataAtlas can be found under the **Installation** topic on the CD-ROM's Electronic Showcase.

How This Book Is Organized

Read Chapter 1 to find out about the constructs and the notation used in DataAtlas Modeler. The remaining chapters describe DataAtlas Modeler and how to create data models, carry out forward and reverse engineering, and create documents.

The appendixes contain reference information. If you are responsible for installing DataAtlas Modeler, read Appendix A for information about the DataAtlas Modeler client/server environment and the relationship to IBM VisualAge TeamConnection Version 2, referred to as TeamConnection in this book.

Related Publications

In addition to this book, the DataAtlas library includes the following books:

Installation Overview, SC26-9042

Dictionary and Designer User's Guide, SC26-9134

This book does not cover how to use TeamConnection. For information about user administration, versioning, and access control, refer to the *TeamConnection User's Guide*, SC34-4499.

For information on how to use IBM VisualAge Exchange, referred to as VisualAge Exchange in this book, refer to:

Bridge Builder's Guide, SC34-4508

User's Guide 1, SC34-4506

User's Guide 2, SC34-4507

To order these publications, contact your local IBM representative. In the United States, you can order these publications directly by calling 1–800–879–2755.

DataAtlas Web Site

The DataAtlas Web site provides the latest product information. It also lets you communicate directly with members of the product development team and provides product service updates.

If you have comments about this publication, please go to the DataAtlas Web site, select "Browse or participate in the DataAtlas threaded discussion group," and add a comment to the "Online Information and Publications" thread. The Web site is located at *http://www.software.ibm.com/ad/datatlas*.

Terminology Note

This book and the DataAtlas user interface contain the following abbreviations for DB2 products that might be unfamiliar to you:

- DB/390, which refers to any of these DB2 enterprise servers: DB2 Version 3, DB2 for MVS Version 4, and DB2 for OS/390 Version 5.
- DB/390, which refers to DB2 Universal Database Version 5.

Summary of Changes

This summary describes how enhancements to Version 2.5 of IBM VisualAge DataAtlas Mulitplatform have affected this book:

- **DataAtlas Modeler available on Windows NT**

  DataAtlas Modeler is now available on OS/2 and Windows NT. “Appendix A. DataAtlas Modeler Environment” on page 117 describes some of the possible configuration scenarios and the TCP/IP settings required for each.

- **Support for partial transformation of a data model**

  DataAtlas Modeler now supports the transformation of selected objects in a data model. The selected objects can also be validated for the target database management system (DBMS) before you do the transformation. “Transforming Objects in a Data Model into a Relational Design” on page 74 includes information on partial transformation.

- **More flexibility in the attributes view**

  DataAtlas Modeler’s attributes view lets you specify what attribute information is displayed and in what order. In addition, you can display the technology names and the Oracle data type. “Specifying What Attribute Information Is Shown” on page 38 describes how you can customize your attributes view.

Chapter 1. Introduction to DataAtlas Modeler

Databases continue to become increasingly complex. To design and maintain these complex databases, you need to understand the structure of the data your enterprise uses. DataAtlas Modeler combines techniques from conceptual and logical data modeling to help you capture the structure of your data in a way that can be understood by both the designer and user of the data.

The data models you build in DataAtlas Modeler provide the basis for both the logical and the physical design of your data. DataAtlas Modeler transforms the data model to data structures that are specific to the target database management system (DBMS). The resulting relational design contains table definitions and instances of tables.

DataAtlas Modeler uses TeamConnection for storing and maintaining data models, relational designs, and shareable data elements and for controlling access to all of these objects.

This chapter introduces you to the data modeling, documentation, and transformation concepts used in DataAtlas Modeler.

Data Model Objects and Concepts

DataAtlas Modeler supports the entity-relationship (ER) method for conceptual data modeling. This approach uses ER models to visualize the structure of the data. In DataAtlas Modeler, data models are made up of entity groups, entities, relationships, attributes, constraints, and data elements. These objects are the basic building blocks of a data model.

The constructs and the notation used by DataAtlas Modeler are described here.

Entity Group

An *entity group* is a named group of entities and their relationships. A data model always has at least one entity group, its root entity group.

In DataAtlas Modeler, an ER diagram is the graphical representation of an entity group—its diagram view. Figure 1 on page 2 shows a DataAtlas Modeler diagram view:

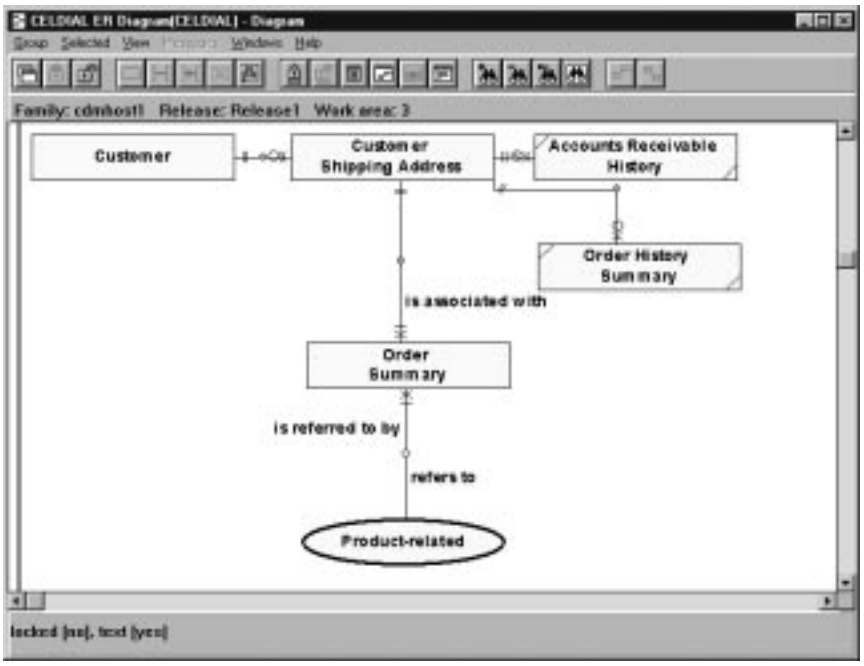


Figure 1. Diagram View of an Entity Group

Use entity groups to consolidate entities that belong together and to organize your model hierarchically—this makes the model easier to read and work with. For example, in a model of an enterprise, you might want to group together those entities that have to do with purchasing activities into a group called purchasing.

An entity group can contain other entity groups.

Entity

An *entity* represents a thing in the real world, usually a person, a place, an object, an event, or a concept. A single occurrence of this entity is called an *entity instance.* Person is an example of an entity; “the person with person ID 553-2729” is an example of an entity instance.

DataAtlas Modeler supports fundamental, associative, and attributive entities. In addition, if an entity has some special significance within an entity group or it is the main entity in an entity group, you can specify that the entity is a *focal entity*. This graphically highlights the entity in the ER diagram.

Fundamental Entity

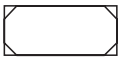
A *fundamental entity* is an entity that represents a class of objects in the real world, for example, employee, job, passenger, flight. Typically, most entities in a data model are of this type.

In an ER diagram, a fundamental entity is shown as:

Associative Entity

An *associative entity* is another way to represent a relationship that has local attributes or is a many-to-many relationship. You can also use an associative entity to model a relationship with a degree greater than 2, or a relationship to a relationship. An associative entity does not usually represent a real-world object.

In an ER diagram, an associative entity is shown as:



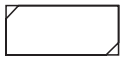
Attributive Entity

In DataAtlas Modeler, the attributes of an object must all be single-valued and atomic. You can use an *attributive entity* to model a composite attribute or a multivalued attribute.

An attributive entity is a dependent entity; its existence depends on the existence of another entity, the *identifying owner*. The relationship from the attributive entity to the identifying owner is the *identifying relationship*. This relationship must be a mandatory one-to-one or one-to-many relationship. An attributive entity can take part in more than one relationship, but it can have only one identifying owner.

For example, if a person can have several addresses, the attribute address of the entity person might consist of the attributes street, city, and state. The attribute address is then composite and multivalued. You can model address as an attributive entity with a one-to-many relationship between person and address.

In an ER diagram, an attributive entity is shown as:



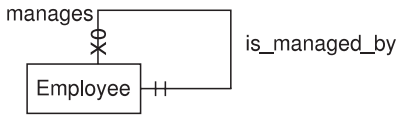
Relationship

A *relationship* is an association between one or more entities. Relationships have a degree, a cardinality, and they can also have verbs.

Degree of a Relationship

The number of different entities that participate in a relationship determine its *degree*. In DataAtlas Modeler, relationships are unary (degree 1) or binary (degree 2). To model relationships with a degree greater than 2, use an associative entity.

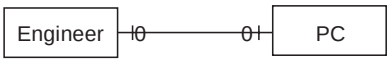
Unary Unary relationships are represented in an ER diagram as follows:



For example, an employee manages at least zero and at most many employees. An employee is managed by at least one and at most one employee.

Binary

Binary relationships are represented in an ER diagram as follows:



For example, an engineer is allocated zero and at most one PC. A PC is allocated to zero and at most one engineer.

Cardinality

The *cardinality* of a relationship specifies the number of instances of one entity that can or must be associated with each instance of another entity. DataAtlas Modeler uses the following notation for cardinalities:

- *Minimum cardinality* defines the existence rules for the entity in the relationship:

0 The relationship with the entity is *optional*.

| The relationship with the entity is *mandatory*.

- *Maximum cardinality* defines the maximum number of instances of the entity that *can* take part in the relationship:

  | A maximum of one instance of the entity can take part in the relationship.

  X Many instances of the entity can take part in the relationship.

Each relationship has a minimum and maximum cardinality in each direction. Figure 2 shows how minimum and maximum cardinalities are represented in DataAtlas Modeler:

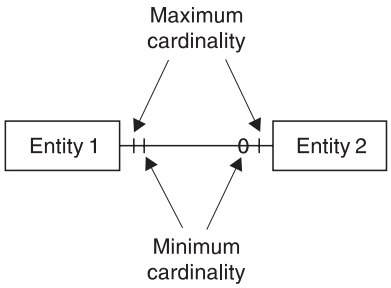


Figure 2. How Maximum and Minimum Cardinalities Are Represented

When describing the cardinality of a relationship, usually only the maximum cardinalities are referred to. For example, relationships are referred to as being *many-to-many*, meaning that the maximum cardinality in each direction is many.

Table 1 lists the possible cardinalities and shows how they are represented in an ER diagram in DataAtlas Modeler:

Table 1. Cardinalities in DataAtlas Modeler

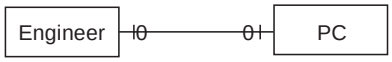
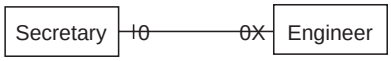
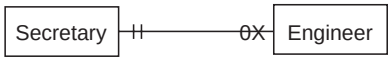
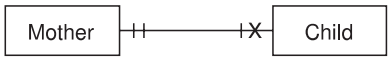
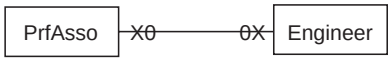
Cardinality	Example
Optional one-to-one	
	A PC is allocated to an engineer or to no one but not all engineers have a PC.
Mandatory one-to-one	
With one entity mandatory: 	Every department must have a manager but a manager can manage at most one department.

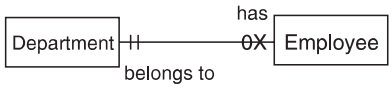
Table 1. Cardinalities in DataAtlas Modeler (continued)

Cardinality	Example
With both entities mandatory: 	A state has exactly one governor and a governor can govern only one state.
Optional one-to-many	
	An engineer can have at most one secretary but a secretary can work for several engineers or none.
Mandatory one-to-one	
With one entity mandatory: 	Every engineer is assigned to exactly one secretary but a secretary can work for several engineers or none.
With both entities mandatory: 	A mother has at least one child. A child has exactly one mother.
Optional many-to-many	
	A professional association can have members who are engineers. An engineer can belong to many professional associations or none.
Mandatory many-to-many	
With one entity mandatory: 	A product is made up of one or more parts. A part can be included in zero or more products.
With both entities mandatory: 	A drug is manufactured by one or more manufacturers. A manufacturer can manufacture one or more drugs.

Verbs

A *verb* describes how entities connected by a relationship relate to one another. Verbs can help to make relationships in an ER diagram easier to read.

For example, assume you want to model the statement: “A department has zero or more employees, and an employee must belong to exactly one department”. The relationship looks like this:



has and belongs to are verbs. You can read the relationships: “a department has zero or more employees”, and “an employee belongs to exactly one department.”

Attribute

An *attribute* is an object that describes a characteristic of an entity or a relationship. Each attribute can have a data element assigned to it that determines its data type. If you are using the data model as the basis for a relational design, you must assign a data element to each attribute.

In DataAtlas Modeler, attributes must be single-valued and atomic; the value can be mandatory or optional. To model a multivalued or composite attribute, use an attributive entity.

Attributes can be either keys or nonkeys. The keys can be primary keys or foreign keys. A primary key is mandatory; a foreign key and the nonkey attributes can be mandatory or optional.

Primary Key

A *primary key* is an attribute or group of attributes that uniquely identifies each instance of an entity or relationship. The values of a primary key must be unique and non-null.

A primary key that consists of more than one attribute is called a *composite primary key*. Each of the attributes that make up a composite primary key is called a *partial primary key*. Each partial primary key must have a value of non-null, and each combination of the partial primary keys must be unique.

Foreign Key

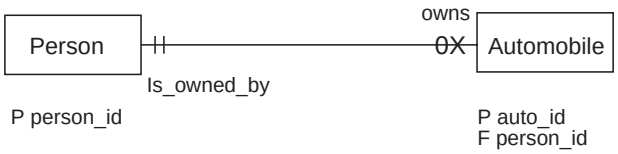
A *foreign key* is an attribute generated from a primary key of another entity along a relationship. It identifies which instance of an entity is associated with an instance of that relationship. A foreign key that is comprised of more than one attribute is called a *composite foreign key*. Each of the attributes that make up a composite foreign key is called a *partial foreign key*.

Note: Foreign keys are not conceptual data modeling constructs. Because DataAtlas Modeler is a relational database modeling tool, foreign keys are automatically generated in the data model. If you are designing a conceptual model only, you can ignore the foreign keys.

Foreign keys are generated automatically by DataAtlas Modeler in one of the following ways:

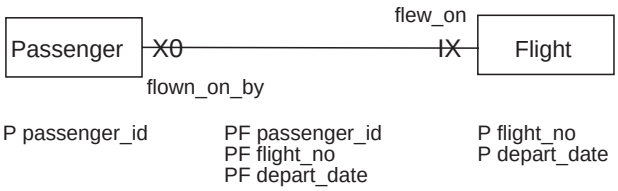
- From one entity to another along a relationship

  For example, one person owns zero or more automobiles, and one automobile is owned by exactly one person. Each automobile has as its foreign key the primary key of exactly one person, and this foreign key uniquely identifies the owner. The foreign key propagation looks like this:



- Into a relationship from the participating entities

  For example, a passenger flew on at least one flight, and a flight was flown on by zero or more passengers. The relationship receives as foreign keys the primary keys of passenger and flight. For each instance of the relationship, these keys together uniquely identify the corresponding entity instances. The foreign key propagation looks like this:



"Appendix B. How DataAtlas Modeler Propagates Keys" on page 127 describes the rules that determine how a foreign key is generated.

Constraint

A constraint in DataAtlas Modeler is a rule that governs the validity of the data manipulation operations, such as insert, delete, and update, associated with an entity or a relationship. You can document constraints in DataAtlas Modeler. This information is then available for the programming specifications.

For example, if you are working on a data model for a bank, you might specify as a constraint on the entity *withdrawal* that the value of its attribute *amount* must be less than or equal to the balance of the corresponding account. This constraint would prevent someone from being able to withdraw more money than they have in their account.

Data Element

In DataAtlas Modeler, a *data element* defines the data type, size, and other characteristics of an attribute. The *data type* identifies the physical representation of the data, for example, *real*, *integer*, *character*, and *binary* in the target DBMS.

Each attribute can have exactly one data element associated with it, but one data element can apply to any number of different attributes. For example, you have several entities that have the attribute *age*. If *age* is always an integer in your data modeling context, you can create one data element with the name *age* and the data type *integer*, and assign it to each of the *age* attributes in the various entities.

Data elements are shareable throughout TeamConnection.

Generalization and Specialization

Generalization and specialization are techniques for classifying entities:

- Generalization

  Groups together entities with common attributes and associates them with a generic entity, a supertype. The individual entities then become subtypes of the supertype.

  For example, if your model contains the entities car, truck, and motorcycle, you might recognize that they all share certain attributes and create the supertype vehicle. You can then move the common attributes to the supertype, and connect the supertype to its subtypes via an isa bundle. The attributes of vehicle would apply to car, truck, and motorcycle.

- Specialization

  Splits off, via subtypes, entities that have attributes that only apply to them.

  For example, if your model contains the entity *person*, you might recognize that some persons are tennis players and others are hockey players, and that these classes of people have attributes that are unique to them. The attribute *racquet_size* applies only to tennis players, and the attribute *goals_scored* applies only to hockey players. You can create new entities called *tennis_player* and *hockey_player*, and move the attributes *racquet_size* and *goals_scored* to the appropriate entities.

Figure 3 on page 10 shows the generalization and specialization constructs used by DataAtlas Modeler:

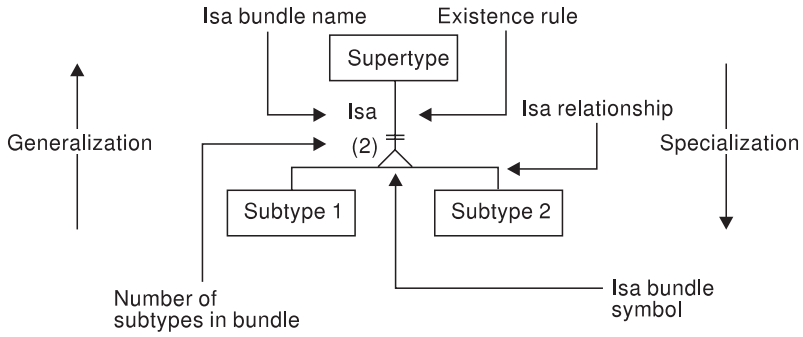


Figure 3. Generalization and Specialization Constructs

Supertype

A *supertype* is an entity that describes a superset of other, less global entities (its subtypes). For example, the supertype *athlete* might have the subtypes *runner* and *swimmer*. All of a supertype's attributes and relationships apply to each of its subtypes.

A supertype is connected to each of its subtypes by an isa relationship.

Subtype

A *subtype* is an entity that describes a subset of another, more global entity (its supertype). For example, the supertype *athlete* might have the subtypes *runner* and *swimmer*.

A subtype receives the primary key of its supertype as a foreign key. The attributes and relationships of the supertype apply to the subtype. Typically, a subtype has additional attributes and relationships that apply only to it and not to the supertype or to other subtypes of the same supertype.

A subtype is connected to its supertype by an isa relationship.

Isa Relationship

An *isa relationship* is a relationship between a supertype and one of its subtypes. In DataAtlas Modeler, isa relationships are not independent objects—they exist only in the context of an association of a supertype with one of its subtypes.

An isa relationship always has the following cardinality:

- 0,1 in the direction from the supertype to the subtype (one supertype instance can have zero or one relationship to one instance of the subtype)
- 1,1 in the direction from the subtype to the supertype (one subtype instance must have exactly one relationship to one instance of the supertype).

These cardinalities are not shown in an ER diagram; the existence rules are shown instead.

Isa Bundle

An *isa bundle* is a group of related isa relationships that you use for generalization or specialization. It is the object used to identify a generalization or specialization.

Existence Rules

In DataAtlas Modeler, there are two rules that determine the existence of an isa bundle:

- Mandatory or optional
- Exclusive or nonexclusive

Mandatory
Each instance of the supertype must have at least one relationship to an instance of one of its subtypes. Mandatory isa bundles are typified by the phrase *must be*, as in "A person *must be* exactly one of (male, female)."

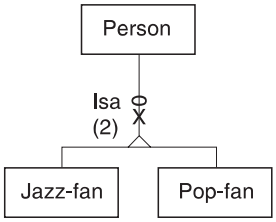
Optional
Each instance of the supertype need not have a relationship to an instance of one of its subtypes. Optional isa bundles are typified by the phrase *can be*, as in "A mammal *can be* at most one of (dolphin, dog, bear, hog)."

Exclusive
Each instance of the supertype has at most one relationship to one instance of one of its subtypes. For example, a person is of the subtype male or female, but not both. Exclusive isa bundles are typified by the phrase *at most one of*, as in "An animal is *at most one of* (reptile, bird, mammal, insect)".

Nonexclusive
Each instance of the supertype has relationships to instances of one or more of its subtypes. For example, an athlete is of the subtype swimmer, runner, cyclist, or any two, or all three. Nonexclusive isa bundles are typified by the phrase *one or more of*, as in "An athlete is *one or more of* (swimmer, runner, cyclist)."

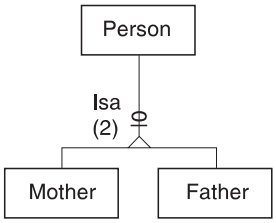
In DataAtlas Modeler, these constructs are shown as follows in an ER diagram:

Optional, nonexclusive



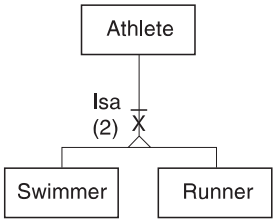
For example, a person can be either a jazz fan, a pop fan, or both. There is an optional, nonexclusive isa bundle from Person (the supertype) to Jazz-fan and Pop-fan (the subtypes).

Optional, exclusive



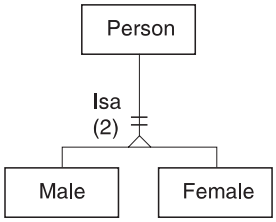
For example, a person can be either a father or a mother, but not both. There is an optional, exclusive isa bundle from Person to Father and Mother.

Mandatory, nonexclusive



For example, an athlete must be one or more of swimmer or runner. There is a mandatory, nonexclusive isa bundle from Athlete to Swimmer and Runner.

Mandatory, exclusive



For example, a person must be either a male or a female, but cannot be both. There is a mandatory, exclusive isa bundle from Person to Male and Female.

Naming Objects

Data model objects in DataAtlas Modeler have two names; a conceptual name and a technology name. The conceptual name is a descriptive name that communicates the function of the object within the model and is meaningful to the user of the data. The technology name is used in the forward transformation of the model to create the name of the corresponding physical object. It enables you to use company-specific naming conventions in a conceptual model that apply to the physical implementation. Technology names are available for entity groups, entities, relationships, attributes, and constraints.

If you are designing a conceptual model only, you can ignore the technology name—DataAtlas Modeler will automatically generate a technology name for you. Then, if you decide at a later date to use the model as the basis for a physical implementation, you can use either the generated technology names or specify your own.

Document Objects and Concepts

You can produce documents containing the data model objects and their views directly from DataAtlas Modeler. To enable you to create documents, DataAtlas Modeler also has document and document profile objects.

Document

In DataAtlas Modeler, a *document* contains the graphical and textual descriptions of some or all of the objects contained in a data model. A document also contains, for each object, a cross-reference to each related

object. For example, the section that describes an attribute also refers to the section that describes its corresponding data element. The content, structure, and format of a document is controlled by elements in a *document profile*.

You can either print the document directly from DataAtlas Modeler or save it in one of several different graphic formats to include in other documents, for example, word processor documents.

Document Profile

Document profiles are like templates; they define the content, structure, and format of documents you create in DataAtlas Modeler. A document profile consists of one or more title pages, headers, footers, document structures, and text blocks:

- Title page

  Text that appears on the front page of the document.
- Header and footer

  Text that appears at the top and bottom of each page. This can include, for example, the date the document was produced, the page number, or the title of the document.
- Document structure

  A *document structure* defines the content of the chapters and the order in which they appear in the document. It also includes settings for headings, chapter and section numbering, and text and graphic formatting information.
- Text block

  A *text block* is additional text included in a document that is independent of a data model and its objects. Typical uses for text blocks are introductory chapters, copyright statements, acknowledgments, and lists of reviewers, but you can use them for anything. Text blocks can be specific to a document, or common to all documents created using the profile.

Document profiles are not stored in TeamConnection. They are stored as files either locally or on a LAN.

Database Engineering

The DataAtlas components support the complete database engineering process as shown in Figure 4 on page 15.

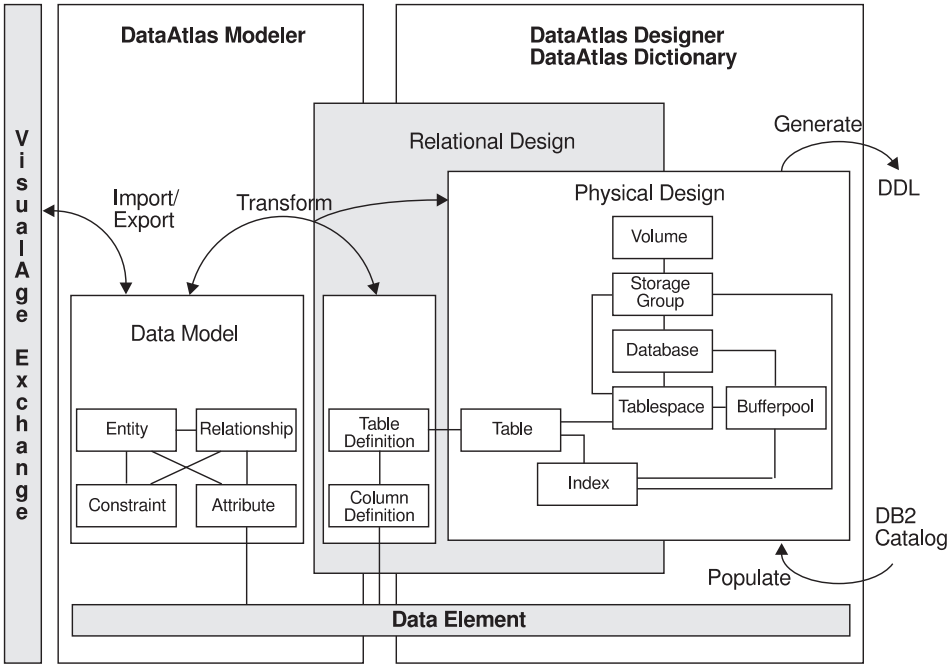


Figure 4. DataAtlas and Database Engineering

DataAtlas Modeler supports both the forward and reverse engineering of data model objects and relational designs. This data engineering process is called *transformation*.

During forward engineering, DataAtlas Modeler transforms objects in a data model into a relational design. A relational design contains the objects that describe a relational database, namely its table definitions, plus one or more physical designs. Each physical design contains the objects specific to a target DBMS, for example, the tables and indexes. DataAtlas Designer uses the relational design to create the physical design, and DataAtlas Dictionary uses the physical design to generate DDL statements.

DataAtlas Dictionary provides a way to populate tables and table definitions with information contained in the database catalog. DataAtlas Modeler can take this relational design and transform it into a conceptual data model. This is called reverse engineering. In addition, if you have IBM VisualAge Exchange installed, you can import data models into DataAtlas Modeler that you have created in other data modeling tools.

Chapter 2. DataAtlas Modeler Road Map

This chapter describes the views you work with when modeling data and creating documents using DataAtlas Modeler. It also describes how to start and end a DataAtlas Modeler session.

Starting a DataAtlas Modeler Session

To start DataAtlas Modeler from the DataAtlas main folder, double-click on the **DataAtlas Modeler** icon.

Tip

If your TeamConnection database is password protected, ensure that you are logged on to TeamConnection before starting DataAtlas Modeler.

The data models view of DataAtlas Modeler opens. Figure 5 shows a typical data models view:

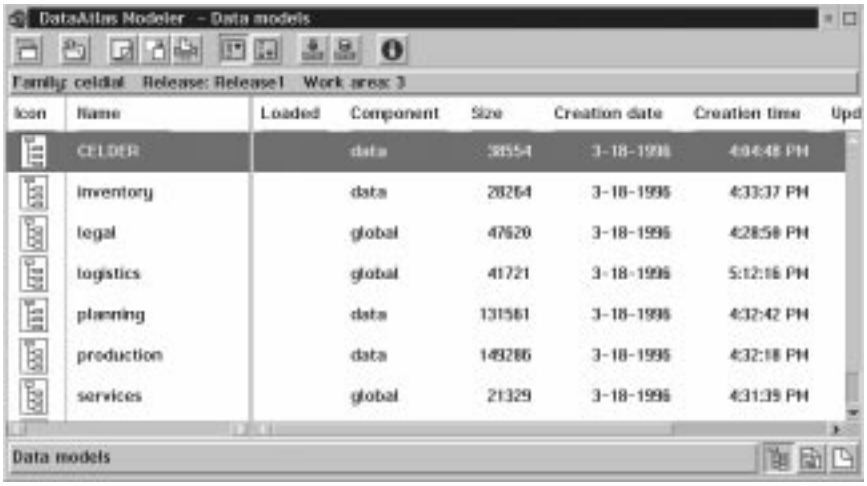


Figure 5. DataAtlas Modeler Data Models View

To open a data model, double-click on it.

Tip

If you open the data models view and the **Release** and **Work area** fields contain question marks (?), check your TeamConnection settings. You must have specified a valid release name, and the work area must have the status **working**. For information on how to change these settings in DataAtlas Modeler, see "Changing TeamConnection Family, Release, or Work Area" on page 31.

DataAtlas Modeler Views

A view shows different aspects of an object. DataAtlas Modeler has the following views:

Data models
Lists the data models you have access to.

Document profiles
Lists the document profiles you have access to.

Documents
Lists the documents created in DataAtlas Modeler that you have access to.

You can move between each of these views by clicking the buttons in the bottom right-hand corner of the view.

To open a...	Click...
Data models view	
Document profiles view	
Documents view	

Every view contains a *tool bar*. The tool-bar choices provide a shortcut to some of the actions in the menu bar or object pop-up menu. "Appendix C. DataAtlas Modeler Tool Bar Choices" on page 129 lists the DataAtlas Modeler tool-bar choices. The data models view also shows the current TeamConnection family, release, and work area underneath the tool bar.

All views also have an Information Area at the bottom of the view. This shows progress information and status information about the current view or selected object.

Data models, document profiles, and documents each have objects that in turn have their own views.

Data Models

DataAtlas Modeler has the following views of a data model and its objects:

Tree view
: Shows the entity groups, entities, relationships, and isa bundles in a data model. It also shows the data elements added to a data model from TeamConnection.

Diagram view
: Shows the entity-relationship (ER) diagram of an entity group.

Attributes view
: Shows the attributes and constraints for an entity or a relationship.

Text view
: Describes an object.

Settings view
: Defines the characteristics of an object, for example, its name. Every object has a settings view. The settings views are not discussed any further in this chapter.

Table 2 shows which views each data model object can have:

Table 2. Data Model Objects and Their Views

Object	Tree View	Diagram View	Attributes View	Text View	Settings View
Data model	x			x	x
Entity group		x		x	x
Entity			x	x	x
Relationship			x	x	x
Attribute				x	x
Constraint				x	x
Isa bundle				x	x
Data element				x	x

Figure 6 shows how you might navigate through these views. Open the next view by double-clicking on an object in the current view.

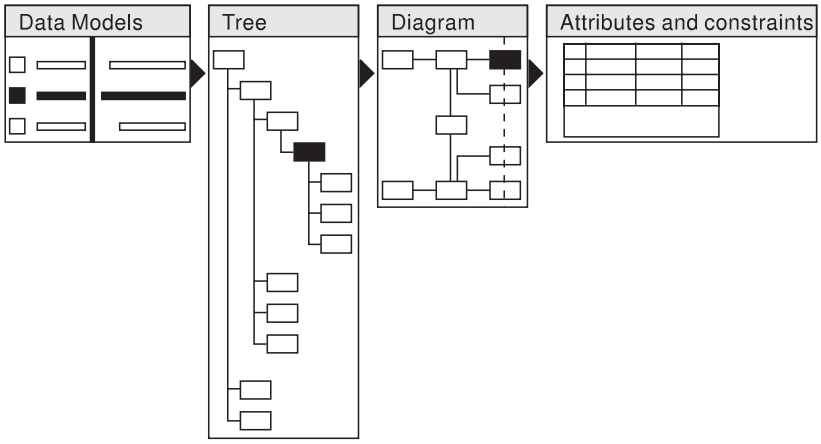


Figure 6. Data Modeling Road Map

Tree View

The tree view is the view that opens when you double-click on a data model in the data models view.

A tree view is divided into the following sections:

Diagram elements
Shows the entity groups, entities, relationships, and isa bundles in the data model.

Data elements
Shows the data elements added to the model from TeamConnection.

Wastebasket
Holds descriptive text from objects that have been deleted during this DataAtlas Modeler session.

You can change the order of the sections in the tree by dragging the section nodes, for example, **Diagram elements**, or the group nodes, for example, **Entities**, to a new position in the tree. Figure 7 on page 21 shows part of a tree view:

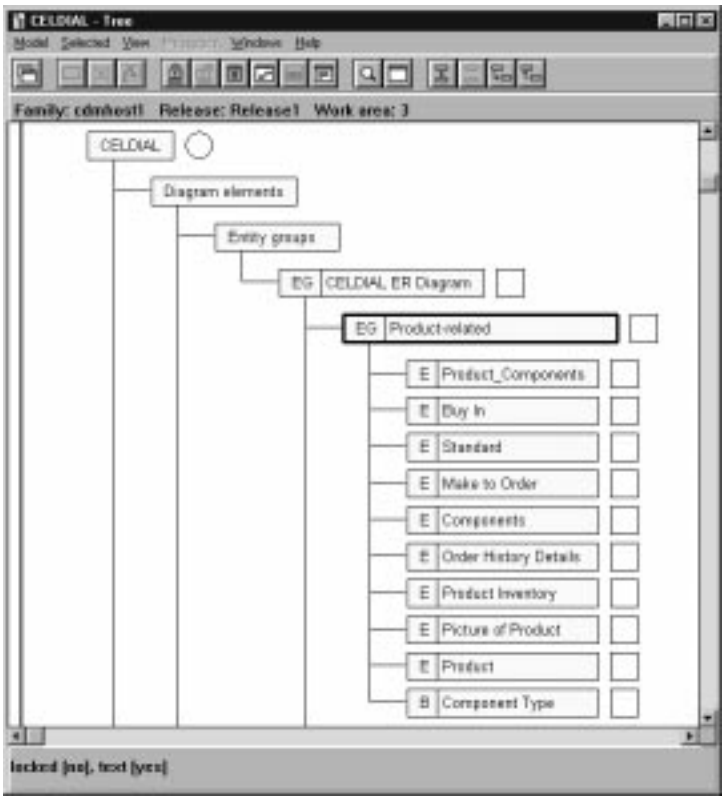


Figure 7. Tree View of a Data Model

The symbols in the tree view have the following meaning:

□	The object is unlocked.
■	The object is locked. This symbol can be either gray or yellow. If the symbol is gray, the object is locked by you. If the symbol is yellow, the object is locked by someone else. When you select a locked object, the Information Area shows who has locked the object. For further information about locking, see "Locking and Unlocking an Object" on page 42.
○	The data model has descriptive text.

The descriptive text for the data model is locked.

This symbol can be either gray or yellow. If the symbol is gray, the object is locked by you. If the symbol is yellow, the object is locked by someone else.

Double-clicking on objects in the tree view opens the following views:

Object	**View**
Entity group	Diagram view
Entity	Attributes view
Relationship	Attributes view
Isa bundle	Settings view
Data element	Settings view

Diagram View

The diagram view shows the ER diagram of an entity group. It contains its entities, relationships, and if applicable, isa bundles. If a relationship involves an entity in another entity group, the entity is shown as:

Such an entity is called a *ghost entity*. If the entity group contains other entity groups, these are shown as:

Figure 8 on page 23 shows part of a diagram view:

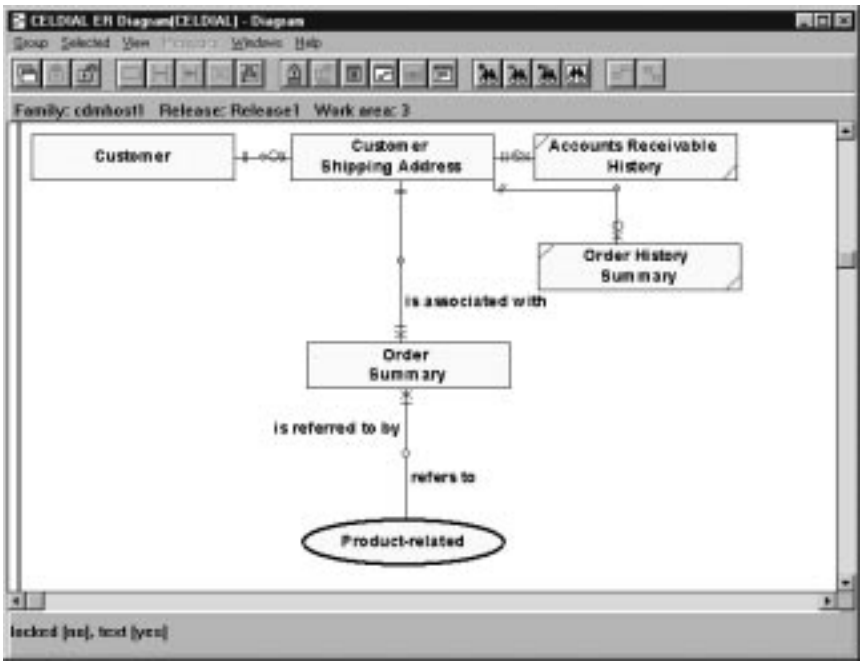


Figure 8. Diagram View of an Entity Group

The diagram view is divided into *pages* indicated by the dashed lines. A page corresponds to a printed page.

Attributes View

The attributes view shows the attributes and constraints belonging to an entity or relationship. The order of the attributes determines the order of the columns in tables in the relational design. Figure 9 on page 24 shows an attributes view:

Figure 9. Attributes View

The attributes view contains the following information for attributes and constraints:

- For an attribute its:
  - Attribute type
  - Attribute name
  - Data element
  - Data type

  By default, the DB2 data type is shown. You can also include the Oracle data type and the attribute technology name. For further information on how to specify what attribute information is shown in the attributes view, see "Specifying What Attribute Information Is Shown" on page 38.
- For a constraint:
  - Constraint name
  - When the constraint is to be checked
  - After which elemental access methods it is to be checked

Text View

The text view is the view you use to describe objects in the data model. All objects in a data model (including the data model itself) can have a text view. Figure 10 shows a text view of an object:

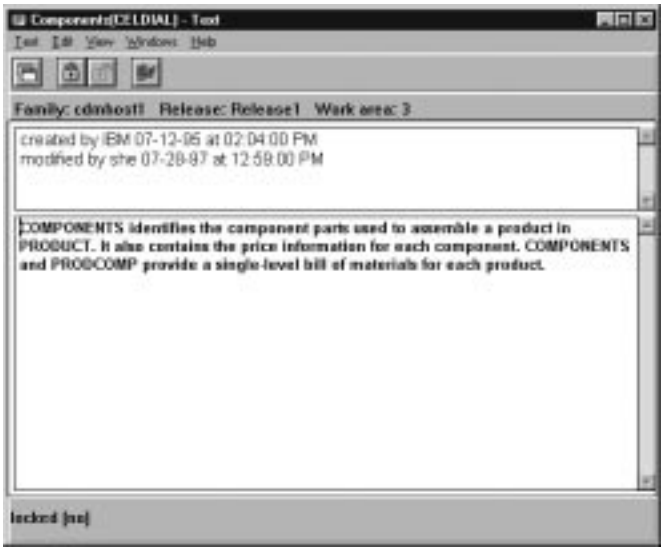


Figure 10. Text View

Document Profiles

Document profiles are like document templates; they define the content, structure, and format of documents you create using DataAtlas Modeler.

DataAtlas Modeler has the following views of document profiles:

Profile tree
: Shows the document elements a document profile contains. It lists the title pages, headers, footers, text blocks, and document structures.

Structure tree
: Shows the structure and content of a document.

Document Profile Tree

Figure 11 on page 26 shows part of a document profile tree:

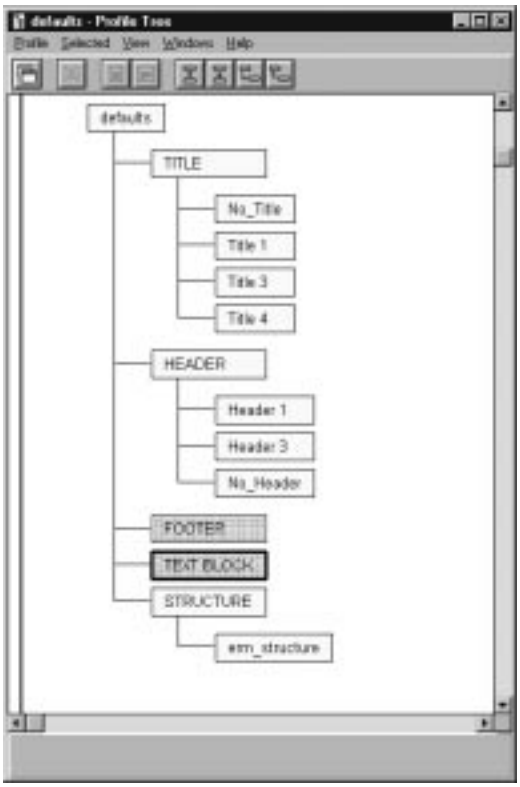


Figure 11. Document Profile Tree

The title page, header, and footer elements determine the content, structure, and format of these elements in a document. A profile can contain several of these elements. When you create a document, you select the appropriate elements for the document from the profile. Double-clicking on any of these elements opens a settings view for that element.

You can include additional text in a document that is not part of the data model. To do this, you must create a text block for the information in the document profile and then include the text block in a document structure. If you double-click on a text block, its text view opens.

The document structure element determines which sections (including sections made up of text blocks) the document contains, and the order in which they appear. Double-clicking on a document structure opens the document structure tree.

The symbols in the document profile tree have the following meaning:

○ The text is closed.

● The text is open.

Document Structure Tree

The document structure tree represents a standard document structure based on all the objects in a data model. Figure 12 shows part of a document structure tree:

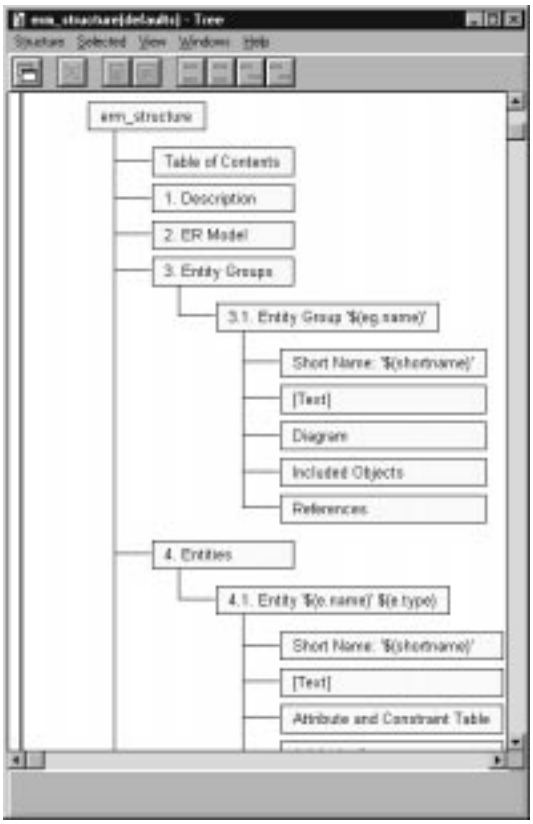


Figure 12. Document Structure Tree

A document structure contains the following chapters and sections that you can include and exclude from your document:

- Table of Contents
- Chapter 1: Description
- Chapter 2: ER Model

  Includes the tree view of the data model.
- Chapter 3: Entity Groups

  Includes all the entity groups in the data model. For each entity group, it includes its technology name, description, diagram view, a list of objects in the entity group, and the entity group on the next higher level that contains this entity group (references).
- Chapter 4: Entities

  Includes all the entities in the data model. For each entity, it includes its technology name, description, attributes, constraints, and in which entity group the entity is located. Each attribute and constraint includes its technology name and description, and each constraint also has information about its elemental access methods.
- Chapter 5: Relationships

  Includes all relationships in the data model. For each relationship, it includes its technology name, description, attributes, constraints, and the entities participating in the relationship. Each attribute and constraint includes its technology name and description, and each constraint also has information about its elemental access methods.
- Chapter 6: Isa Bundles

  Includes all isa bundles in the data model. For each isa bundle, it includes its description, participating entities, existence rules, and the entity group where the isa bundle is located.
- Chapter 7: Data Elements

  Includes the data elements added to the data model. For each data element, it includes its description, the DBMS representation, and a list of the attributes that use the data element.

  Note: If you create a document containing only data elements or the document structure contains only data elements, the resulting document will not contain the where-used list of attributes.
- Chapter 8: Index

The headings for each section containing objects are shown as variables prefixed by the object type. For example, in the entity group chapter the heading for the individual entity group sections is shown as: `Entity Group '$(eg.name)'`. When you create a document, the variable is replaced with the name of the entity group. You can change the name used for the object type, but you cannot change the variable or use the variable elsewhere.

Documents

The documents view lists the PostScript documents created in DataAtlas Modeler that you have access to. From here you can copy, delete, and print documents.

User Profile

The user profile allows you to specify settings for the way you work with DataAtlas Modeler that apply only to your workstation. You can get to the user profile from the data models view, the document profiles view, and the documents view.

You can set the following settings in your user profile:

- Deletion confirmation
- Minimum font size for a diagram view
- Whether the following is shown:
  - A data model tree view as a tree graphic, that is, each element in the tree view is contained in a box that is connected by lines to its parent and siblings
  - History information
  - A frame around a view
  - Page breaks using dashed lines in the diagram view

The changes to the profile become effective when you open a model; they have no effect on models that are already open.

Ending a DataAtlas Modeler Session

To end a DataAtlas Modeler session:

1. Close the tree view of the data model.
2. If you are using DataAtlas Modeler in a single-user environment, or you are the only person using the model in a multiuser environment, the Unload Data Model window opens. Figure 13 on page 30 shows the Unload Data Model window.

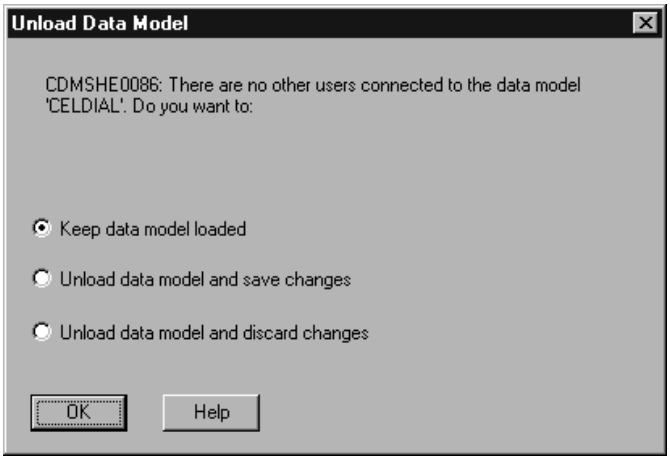


Figure 13. Unload Data Model Window

3. Click:
   - **Keep model loaded** to end a data modeling session but keep the data model loaded; the data model is not checked into TeamConnection.
   - **Unload and save changes** to end a data modeling session and check the changes to the data model into TeamConnection.
   - **Unload and discard changes** to end a data modeling session and discard all changes made to the data model since the model was checked out of the TeamConnection database. This discards **all** changes made by **all** users except for changes made to shareable data elements.

     Note: This selection is not available if the DataAtlas Modeler control server was started with the parameter /N (no unload and discard changes). For information on how to set this parameter, see "Starting a DataAtlas Modeler Control Server" on page 122.

Chapter 3. Working with DataAtlas Modeler Views

This chapter describes general tasks that are applicable to several views and some of the more common tasks applicable to a specific view.

Changing TeamConnection Family, Release, or Work Area

All data model views show the current TeamConnection family, release, and work area.

To change the TeamConnection family, release, or work area:

1. In the data models view, do one of the following:
   - Move the mouse pointer to an empty area in the view, for example, to the column headings. Click mouse button 2 to display the pop-up menu for the view. In the pop-up menu, click **Settings**.
   - Click the Change View and TeamConnection Settings choice in the tool bar.

   The Settings notebook opens.
2. Go to the **TeamConnection** page:
   - Make your changes to the release or work area.
   - To change the family, click **Change**.

     Note: You cannot change the family if you have data models open.

   Close the notebook when you have finished.

   A list of models in the new family, release, or work area appears.

Changing the Default Position and Size of Windows

You can change the default position and size of the windows used to display object views. This can be useful, for example, if you change from an XGA screen to a VGA screen.

To change the default position and size of the windows used to display the object views:

1. Move and size the window.
2. In the object menu (the one on the far left of the menu bar), click **Layout**.

3. In the Layout window:
   - Click **Apply** to change the window position and size for this session.
   - Click **Save** to save the window position and size as the default.

Hiding and Showing the Tool Bar

All views in DataAtlas Modeler have a tool bar. The tool bar provides a shortcut to some of the actions in the menu bar or the object pop-up menu.

You might prefer to work without the tool bar in a particular view, for example, to increase the amount of space available for a graphic. You can choose to work without the tool bar in all views except the data models view, the document profiles view, and the documents view.

To hide or show the tool bar in a view, do one of the following:

- In the **View** menu, click **Hide tool bar** or **Show tool bar**.
- Click the small, square button in the lower right-hand corner of the window, between the horizontal and vertical scroll bars.

  Note: This is not available in the text view.

"Appendix C. DataAtlas Modeler Tool Bar Choices" on page 129 contains a list of all the available tool-bar choices.

Navigating among Windows

When you are in any tree view, diagram view, attributes view, or text view, you can navigate between windows you already have open. To list and go to a window you already have open:

1. In the **Windows** menu, click **Window list**.
2. Click a window in the list and then **Show**.

Tree View

The following describes some of the tasks that you can do in a tree view.

Navigating in a Tree View

To help navigate around the objects in a tree view of a data model, document profile, or document structure, you can use the following choices in the **View** menu:

- **Select next sibling**
- **Select previous sibling**
- **Select first child**
- **Select parent**

You can also use the cursor keys or the corresponding choices in the tool bar to move around the tree view. For a list of tool-bar choices, see "Appendix C. DataAtlas Modeler Tool Bar Choices" on page 129.

Restricting the Tree View

You might only be interested in one node of a tree view and its dependents, for example, an entity group and its dependents.

To restrict a tree view of a data model, document profile, or document structure to just these objects, do one of the following:

- Click the node in the tree. In the **View** menu, click **Display selected only**.
- Place the pointer over the node in the tree, and press and hold down Ctrl+mouse button 2.

To return to the complete tree view, do one of the following:

- In the **View** menu, click **Display all**.
- Place the pointer over the node in the tree, and press and hold down Ctrl+mouse button 2.

DataAtlas Modeler displays all objects in the tree view.

Hiding and Showing Objects

A tree view of a data model or a document profile might show more detail than you need. You can temporarily reduce the level of detail by hiding the dependents of one or more nodes. This is called *collapsing* a node. Showing the hidden dependents of a collapsed node is called *expanding* a node.

To collapse nodes in a tree view, do one of the following:

- Place the pointer over each node you want to collapse, and press and hold down Shift+mouse button 2.
- Click the nodes that you want to collapse, and in the **Selected** menu, click **Collapse**.

To expand collapsed nodes in a tree view:

- Place the pointer over each node you want to expand, and press and hold down Shift+mouse button 2.

- Click the nodes that you want to expand, and in the **Selected** menu, click **Expand**.

Sorting the Order of Objects

In the tree view of a data model, you can sort the order of the objects in the following groups: **Entity groups**, **Entities**, **Relationships**, **Isa bundles**, and **Data elements**.

To sort a group of objects:

1. Click the appropriate group node, for example, **Entity groups**.
2. In the **View** menu, click **Sort order**.
   The Sort Order window opens.
3. Select the sort method; **User defined** and **By creation date** are not available for data elements.
4. Click **OK** to sort the objects according to the selected sort method.

If you specify **User defined** as the sort order, you can move objects within the same object group. For example, you can move an entity to a new position in the **Entities** group, but you cannot move it to the **Relationships** group.

Tip

If you specify **User defined** or **Alphabetically** as the sort order, this order is kept in documents you create in DataAtlas Modeler. If you specify **By creation date** or **By modification date** as the sort order, the objects are sorted alphabetically in documents you create in DataAtlas Modeler.

Diagram View

The following describes some of the tasks that you do in a diagram view.

Navigating among Diagram Views

A larger data model can have many entity groups. There are several ways to navigate among these entity groups.

To open any entity group:

1. In the **Windows** menu, click **Object**.

The Object window opens. Here you can list all the objects of a certain type.

2. Click **Entity group**.

   A list of all the entity groups in the data model opens.

3. Click an entity group in the list and then **OK**.

   DataAtlas Modeler opens the diagram view of the selected entity group.

To go to the entity group that is the parent of a ghost entity:

1. Click the ghost entity.
2. In the **Selected** menu, click **Go to**.

   DataAtlas Modeler opens the entity group that contains the entity that corresponds to this ghost entity.

If the current entity group is the child of another entity group, you can go to the parent entity group in one of the following ways:

- Double-click on an empty area in the diagram view.
- In the **Windows** menu, click **Parent entity group**.

DataAtlas Modeler opens the parent entity group.

Zooming In and Out of a Diagram

When you open a diagram view of an entity group, it is sized so that all the objects in the entity group are visible in one window. The setting for the minimum font size in the user profile determines whether the text is shown. You can change the minimum font size as described in "Specifying the Minimum Font Size for a Diagram View" on page 36.

To zoom in on a diagram view, do one of the following:

- In the **View** menu, click **Zoom in**.

  DataAtlas Modeler zooms in on the diagram by 20%. The top left corner of the diagram is used as the reference point for the zoom action.

- Use a drag box to select an area to zoom in on. In the **View** menu, click **Zoom in**.

  DataAtlas Modeler zooms in on the diagram as far as it can and still display the area marked by the drag box.

To zoom out of a diagram view, do one of the following:

- In the **View** menu, click **Zoom to fit in window**.

  DataAtlas Modeler zooms so that the entire diagram just fits in the current window.

- In the **View** menu, click **Zoom out**.

  DataAtlas Modeler zooms out of the diagram by 20%. The top left corner of the diagram is used as the reference point for the zoom action.

To undo your last zoom action, do the following:

- In the **View** menu, click **Undo zoom**.

Instead of using the menu-bar choices, you can use the corresponding tool-bar choices For a list of tool-bar choices, see "Appendix C. DataAtlas Modeler Tool Bar Choices" on page 129.

Specifying the Minimum Font Size for a Diagram View

Sometimes when viewing a diagram, you might want to see all the objects in the entity group. However, when you zoom out, the text in the diagram can become too small for you to read. To avoid this, set the minimum font size to the smallest size you can comfortably read.

To change the minimum font size for a diagram view:

1. Open the data models view, document profiles view, or document view.
2. In an empty area of the view, click mouse button 2. In the pop-up menu, click **User profile**.

   The User Profile notebook opens.
3. Specify the new minimum font size and close the notebook.

   The changes to the profile become effective when you open a model; they have no effect on models that are already open.

Changing the Size of an ER Diagram

You can change the size of an ER diagram by stretching and compressing it:

1. Press and hold down Shift+mouse button 2 and move the mouse pointer.

   DataAtlas Modeler displays a grid on top of the diagram view. The grid stretches and compresses as you move the mouse pointer.
2. When you have resized the diagram, release the Shift key and the mouse button.

Adding and Deleting Pages

The pages defined in a diagram view are used when you print the diagram view. They can also help you to visually divide up a model when you are creating it on the workstation.

Often, the diagram view of an entity group contains too many entities for it to fit easily on one page. To improve the readability of the diagram, you can add additional rows and columns of pages. Ensure that you have the entity group locked. For information on locking objects, see “Locking and Unlocking an Object” on page 42.

- To add a column of pages, in the **View** menu, click **Add column of pages**.
- To add a row of pages, in the **View** menu, click **Add row of pages**.

You can delete rows or columns of unused pages, or delete rows and columns together so that the set of defined pages remains a rectangle. DataAtlas Modeler deletes columns from right to left and rows from bottom to top.

- To delete a column of pages, in the **View** menu, click **Delete column of pages**.
- To delete a row of pages, in the **View** menu, click **Delete row of pages**.
- To delete all unused rows and columns of pages, in the **View** menu, click **Remove pages**.

Aligning Entities

To align entities horizontally or vertically with respect to a reference object:

1. Lock the entity group.
2. Select the objects you want to align.

   DataAtlas Modeler highlights the selected objects.
3. In the **View** menu, select **Align horizontally** or **Align vertically**.
4. Choose one of the highlighted objects as the reference object.

   The selected objects are aligned with respect to the reference object.

Instead of using the choices in the menu bar, you can use the corresponding choices in the tool bar to align entities. For a list of tool-bar choices, see “Appendix C. DataAtlas Modeler Tool Bar Choices” on page 129.

Attributes View

The following describes some of the tasks that you can do in an attributes view.

Navigating among Attributes Views

You can use the primary and foreign keys to navigate among attributes views.

Finding the Source of a Foreign Key

To find the source of a foreign key:

1. Open the attributes view of the entity or relationship.
2. Select a foreign key.
3. In the **Selected** menu, click **Show primary key**.

   DataAtlas Modeler displays the attributes view of the entity containing the corresponding primary key and highlights the primary key in that view.

Listing Entities That Refer to a Primary Key as a Foreign Key

To list the entities that refer to a primary key as a foreign key:

1. Open the attributes view of the entity.
2. Select the primary key.
3. In the **Selected** menu, click **List foreign keys**.

   DataAtlas Modeler displays a list of the entities that contain foreign keys referring to the primary key. You can select an entity from the list and open its attributes view. DataAtlas Modeler highlights the foreign key in that view.

Locating Relationships Associated with a Foreign Key

To locate the relationship in the diagram view of an entity group that is associated with a foreign key in the attributes view of an entity:

1. Open the attributes view of the entity.
2. Select the foreign key.
3. In the **Selected** menu, click **Show relationship**.

DataAtlas Modeler displays the diagram view of the entity group. The relationship associated with the foreign key is highlighted.

Specifying What Attribute Information Is Shown

By default, the attributes view shows the attribute type, name, data element, and DB2 implementation. You can customize the view to include the attribute technology name and the Oracle implementation, to change the order in which the information is shown, and to remove information from the view.

To specify what information is shown in the attributes view, click the **View** menu, then **Show columns**. The Show Columns window for attributes opens.

- To add information, click the corresponding entries in the **Available Columns** list, and then >>.

- To remove information, click the corresponding entries in the **Visible Columns** list, and then <<.

  Note: You cannot remove the attribute name or the attribute type.
- To change the position of a column in the attributes view, click the entry in the **Visible Columns** list, and then click either **Up** or **Down**.

Specifying Where New Attributes and Constraints Are to Be Placed

The order of the attributes in an attributes view determines the order of the columns in the corresponding table in the relational design. For each attributes view, you can specify whether new attributes are to be placed before or after the selected attribute. The current setting is shown in the Information Area at the bottom of the window.

To define where you want to place new attributes:

1. Open the attributes view of the entity or relationship.
2. In the **View** menu, click **New item placement**, then **Before** or **After**.

Chapter 4. Working with Objects

This chapter describes common tasks that apply to all of the objects types in DataAtlas Modeler.

Selecting and Deselecting Objects

Table 3 shows how to select objects in DataAtlas Modeler. When you select objects as described in Table 3, all previously selected objects are deselected, unless you press and hold down the Ctrl key while you click on the objects with mouse button 1.

Table 3. Selecting Objects in DataAtlas Modeler

To Select...	**Do the Following...**
A single object in any view	Click mouse button 1 on it.
Several objects in any view	Press and hold down the Ctrl key while you click on the objects with mouse button 1.
An object and its dependents in a tree view	Press and hold down the Shift key, and click mouse button 1 on the object.
The root node in a tree view, or a restricted tree view	Press the space bar.
Several objects in a diagram view	Move the mouse pointer to an empty area near the objects you want to select, press and hold down mouse button 1, and drag a box over the objects.
An entity and all its partners in a diagram view, plus the relationships that connect them	Press and hold down the Shift key, and click mouse button 1 on the entity.
All the attributes of the same type in an attributes view	Press and hold down the Shift key and click mouse button 1 on any one of the attributes.

Table 4 shows how to deselect objects in DataAtlas Modeler.

Table 4. Deselecting Objects in DataAtlas Modeler

To Deselect...	**Do the Following...**
An object in any view without deselecting other objects	Press and hold down the Ctrl key, and click mouse button 1 on the object.
All selected objects in any view	Click mouse button 1 on an empty area in the view.
The root node in a tree view, or a restricted tree view	Press the space bar.

Locking and Unlocking an Object

DataAtlas Modeler lets several users work simultaneously on the same model. However, to ensure that two users do not change the same object at the same time, DataAtlas Modeler requires that a user locks an object before changing it.

An object can be locked by only one user at a time. When you select an object or open a view of an object, the Information Area at the bottom of the window shows whether the object is already locked and by whom.

Lock an object when:

- You want to prevent others from changing it.
- You want to change it yourself.

If you still have objects locked when you end your DataAtlas Modeler session, these objects are automatically unlocked when you close the tree view of a data model.

Table 5 shows which objects must be locked by you to carry out some of the more common actions in DataAtlas Modeler. For multiuser installations, additional locking requirements might apply. See the individual task descriptions for further information about these additional requirements.

Table 5. Locking and Actions on Objects

Action	Entity	Relation-ship	Attribute	Constraint	Isa Bundle	Data Element
Create object	EG	EG	E or R	E or R	EG	
Modify object	E	R	E or R	E or R	B	DE
Delete object	EG + E	EG + R	E or R	E or R	EG + B	DE
Create or modify text	E	R	E or R	E or R	B	DE
Add or remove subtype	-	-	-	-	EG and B	-
Group objects	EG	-	-	-	-	-
Ungroup objects	Next higher EG	-	-	-	-	-

Legend:

EG Entity group
E Entity
B Isa bundle

DE Data element
R Relationship

To lock or unlock the object corresponding to a view, for example, an entity group in its diagram view:

- In the object menu (the one at the far left of the menu bar), click **Lock** or **Unlock**.

To lock or unlock an object in a view:

1. Select the object.
2. In the **Selected** menu, click **Lock** or **Unlock**.

If you are using DataAtlas Modeler in a single-user installation, you might find it more convenient to lock or unlock all the objects in a data model at once:

1. Open the tree view of the data model.
2. In the **Model** menu, click **Lock all objects** or **Unlock all objects**.

Opening an Object

Objects in DataAtlas Modeler can have several views. How you open an object depends on what type of object it is and the view you want to open.

To open a view of a data model, document profile, or document:

1. Open the DataAtlas Modeler view containing the object, for example, the data models view to open a data model.
2. Select the data model, document profile, or document you want to open, and do one of the following:

To open a...	**Click...**
Tree view	**Open**, then **Tree view** in the object's pop-up menu. This is only available for data models and document profiles. It is the default (double-click) action for these objects.
Settings view	**Settings** in the object's pop-up menu.

To open a view of an entity group, entity, relationship, attribute, constraint, isa bundle, or data element:

1. Open a view that contains the object. For example, if you want to open a view of an entity, open a tree view of the data model, or a diagram view of the entity group containing it.
2. Select the object.

3. Do one of the following:

To open a...	Click...
Settings view	**Selected, Open as**, then **Settings**. This is the default (double-click) action for isa bundles, data elements, attributes, and constraints.
Diagram view	**Selected, Open as**, then **Diagram**. This is the default (double-click) action for entity groups.
Attributes view	**Selected, Open as**, then **Attributes**. This is the default (double-click) action for entities and relationships.
Text view	**Selected, Open as**, then **Text**.

To open a view of any object from a tree view, diagram view, attributes view, or text view:

1. In the **Windows** menu, click **Object**.

   The Object window opens.
2. Click an object type, an item in the list of objects, and then click **OK** to open the object.

Moving Objects

To move a single object in any view:

1. Lock the object corresponding to the view.

   For example, to move an entity in a a diagram view, you must lock the entity group corresponding to the diagram view.
2. Move the mouse pointer to the object.
3. In OS/2, hold down mouse button 2, drag the object to its new position, and release the mouse button.
4. In Windows NT, hold down mouse button 1, drag the object to its new position, and release the mouse button.

To move several objects in a diagram view at the same time:

1. Lock the entity group corresponding to the diagram view.
2. Select the objects.
3. In OS/2, move the mouse pointer over one of the selected objects, press and hold down Ctrl+mouse button 2, drag the selected objects to a new position, and release the mouse button.
4. In Windows NT, move the mouse pointer over one of the selected objects, press and hold down Ctrl+mouse button 1, drag the selected objects to a new position, and release the mouse button.

Tip

To move objects in a tree view, the sort order for the group must be set to **user-defined**. For information on how to define the sort order for objects, see "Sorting the Order of Objects" on page 34. Using drag and drop, you can move objects in the **Entity groups**, **Entities**, **Relationships**, and **Isa bundles** groups on the same hierarchical level. For example, you can move an entity to a new position within its entity group, but not from one entity group to another.

For information on how to move entities between entity groups, see "Moving Entities between Entity Groups" on page 66 and "Moving a Ghost Entity to an Entity Group" on page 65.

Changing the Settings for an Object

Settings define the properties of an object, for example, its name, its cardinality if it is a relationship, or its type if it is an entity. You define these characteristics when you create an object. You can change the following settings for each of the object types:

Object	Settings
Data model	Name
Document profile	Name
Document	Name
Entity group	Name, technology name
Entity	Name, technology name, type, rank
Relationship	Verbs, cardinalities, name of the relationship, technology name
Isa bundle	Name, existence rules
Attribute	Name, technology name, attribute type, data element
Constraint	Name, technology name, elemental access method, when the constraint is checked
Data element	Name, definition information

To change the settings for a data model, document profile, or document:

1. Open the appropriate view of DataAtlas Modeler.
2. Place the pointer over the object and press mouse button 2.
3. In the pop-up menu, click **Settings**.
   The Settings notebook for the object opens.

4. Make your changes to the settings and close the notebook.

To change the settings for any other data model object:

1. If the object is an entity group, entity, relationship, isa bundle, or data element, lock the object.

   If the change affects the diagram view of the entity group to which the object belongs, ensure that the entity group is not locked by someone else. For example, changing the name of an entity or the cardinality of a relationship affects the entity group to which these objects belong. You cannot change these objects if the entity group to which they belong is locked by someone else.
2. If the object is an attribute or a constraint, lock the entity or relationship to which it belongs.
3. Do one of the following:
   - In an open view of the object, in the object menu (the one at the far left of the menu bar), click **Open as**, then **Settings**.
   - In a view containing the object, select the object, then in the **Selected** menu, click **Open as**, then **Settings**.

   The Settings view for the object opens.
4. Make your changes to the settings and click **OK**.

Specifying Rules for Generating Technology Names

The technology name allows you to specify a name for the physical implementation of the object. For example, the entity technology name is used to generate the table name in the physical implementation of the data model.

You can generate a technology name automatically when you create an object or modify an object name. The technology name is generated from the object name or from an object-specific prefix and a number sequence. You can specify which of these rules DataAtlas Modeler is to use when it automatically generates the technology name.

To specify rules for the automatic generation of the object technology name:

1. Open the tree view.
2. In the **Model** menu, click **Technology name generation rules**.

   The Technology Name Generation Rules window opens. Figure 14 on page 47 shows a Technology Name Generation Rules window.

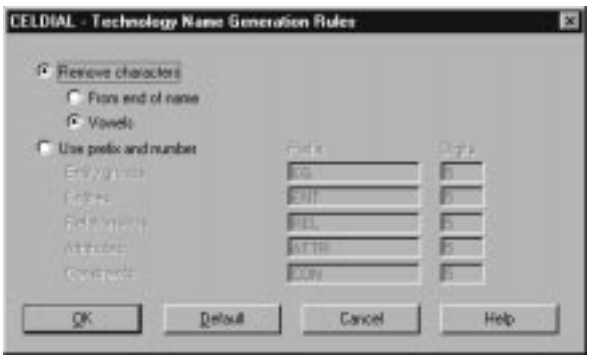


Figure 14. Naming Rules Window

3. Select the technology name generation rules:
   - To generate the technology name from the object name, click **Remove characters**, and then one of the name reduction methods.

     If the resulting name is not unique, DataAtlas Modeler adds a number to the end of the generated name to make it unique.
   - To generate the technology name from an object-specific prefix and a number sequence, click **Use prefix and number**. Specify a prefix for each object type and the number of digits to be used for the number.

     When you create an object, DataAtlas Modeler generates a number with the specified number of digits and adds it to the object-specific prefix to create a unique technology name for the object.
4. Click **OK**.

Specifying the Layout of an Object View

The layout settings determine the appearance of an object on the workstation, in printed graphics views, and in the documents you create in DataAtlas Modeler; the same layout settings are used for each. There are separate layout settings for the views containing graphical information and for the text view.

You can specify the following layout settings:

- Fonts
- Paper size
- Orientation
- Units
- Margins

You can specify layout settings for a view or an object in a view:

1. Do one of the following:
   - For a view, in the object menu (the one at the far left of the menu bar), select **Layout**.
   - For an object in a view, select it, then in the **Selected** menu, click **Layout**.

   The Layout window opens. Figure 15 shows a Layout window.

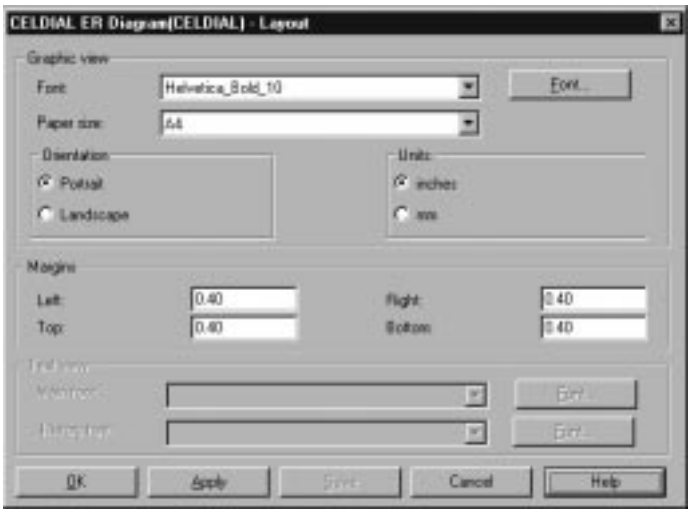


Figure 15. Layout Window

   If the selected object is in a tree view, diagram view, or attributes view, the fields in **Graphic view** and **Margins** are active. If the view is a text view, only the fields in **Text view** are active.

2. Make your changes to the layout. If you are making changes to the font, click **Font** to see a sample of the new font.
3. To see changes to the layout applied to the current view, click **Apply**.
4. To use these layout settings for all newly created objects of the same type, click **Save**.
5. To apply the layout settings and close the Layout window, click **OK**.

   If you changed the layout of an object in a view, you do not see the changes until you open a view of the object. For example, if you changed the layout of an entity in the tree view, you do not see the changes until you open its attributes view.

Specifying the Size of Objects in a Diagram View

To improve the appearance of your ER diagrams, you can specify how objects in an entity group are to be sized. The object size can be:

- Fixed

  All objects in the entity group are the same fixed size. If an object name is too long for the object symbol, DataAtlas Modeler truncates the name so it fits into the symbol. Object names that have been truncated end in periods (...) in the diagram view.

- Dynamic

  All objects in the entity group are the same size. The size of the objects depends on the longest entity or entity group name; the longer the name, the larger the object size. If the length of this name is changed, then the size of all the objects in the ER diagram also changes.

Tip

When you create or modify an entity or an entity group, use an ampersand (&) to indicate where a new line in the name is to begin. For further information on using ampersands in object names, see "Creating an Entity" on page 54 or "Creating an Entity Group" on page 64.

You can specify size settings for an individual entity group or for all entity groups in a model.

1. Do one of the following:
   - To specify size settings for all newly created entity groups, in the tree view, in the **Model** menu, click **Object size**.
   - To specify size settings for an entity group, do one of the following:
     – In the tree view, select the entity group. In the **Selected** menu, click **Object size**.
     – In the diagram view, in the **Group** menu, click **Object size**.

   The Object Size window opens. Figure 16 on page 50 shows an Object Size window.

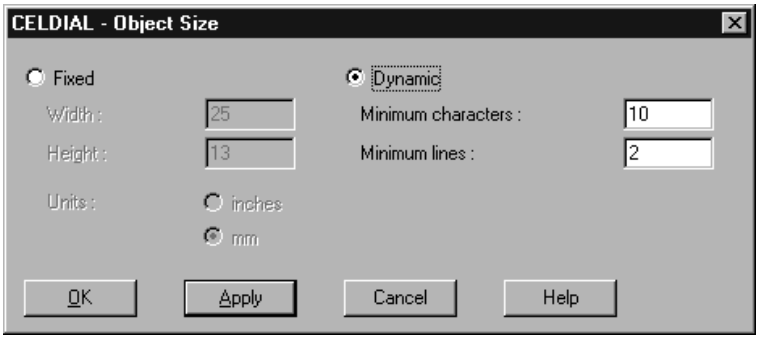


Figure 16. Object Size Window

2. Specify the size settings:
   - If you want all the objects in the entity group to have the same fixed size, click **Fixed**. Specify the object width and height in the appropriate fields.
   - If you want all the objects in the entity group to be dynamically sized, click **Dynamic**. In the **Minimum characters** field, specify the minimum number of characters in each line of the entity name. In the **Minimum lines** field, specify the number of lines for the entity name.
3. Click **OK** to save the settings.

Locating an Object in a Tree View

In a large, complex data model, it can be difficult to find a particular object. To locate a particular object in the tree view of a data model:

1. In the **View** menu, click **Search**.

   The Search window opens.
2. Enter a search pattern, the subtrees you want to search, and click **Search**.

   The **Pattern** field is case-independent. For example, if you enter *prod* as the search pattern, DataAtlas Modeler finds PRODUCT and product.
3. To locate the object in the tree, select one of the objects in the **Results** list, and click **Select**.

   The tree view scrolls to display the object, and the object is selected.

Deleting an Object

To delete a data model, document profile, or document:

1. Open the appropriate view of DataAtlas Modeler.

If the object is a data model, ensure that it is not loaded. If the object is a document profile, ensure that it is not open.

2. Place the pointer over the object you want to delete and press mouse button 2.
3. In the pop-up menu, click **Delete**.

Note: You cannot delete the DEFAULTS document profile.

To delete an object from a document profile, or an object other than a data element from a data model:

1. In a view containing the object you want to delete, select that object.
2. If the object is an entity group, entity, relationship, or isa bundle, lock the object.

   Ensure that the entity group to which the object belongs is not locked by someone else. When you select an object, the Information Area at the bottom of the window shows whether the object is already locked and by whom.
3. If the object is an attribute or a constraint, lock the entity or relationship to which it belongs.
4. In the **Selected** menu, click **Delete**.

   The object is deleted. If the object has a description associated with it, this is put in the wastebasket in the tree view of the data model.

Tip

The **Confirm delete** setting in the user profile determines whether an object is deleted with or without you confirming the deletion. For information about the user profile, see "User Profile" on page 29.

To remove a data element from a data model:

1. Select the data element in the tree view.

   The Information Area shows whether the data element is used in the model; you can remove only those data elements that are not used.
2. Lock the data element.
3. In the **Selected** menu, click **Remove data element**.

   The data element is removed from the data model, but remains in the TeamConnection database.

To remove all the unused data elements from a data model:

1. In the tree view of the data model, select all the data elements.
2. Lock all the data elements.
3. In the **Model** menu, click **Remove unused data elements**.

   A list of all the unused data elements in the model is displayed.
4. Click **OK** to remove all the unused data elements.

   The data elements are removed from the data model, but remain in the TeamConnection database.

Chapter 5. Modeling Data

A data model consists of entities, the relationships between entities, attributes for entities and relationships, isa bundles, and entity groups. Entities and relationships can also have constraints, and each object can have text associated with it. These objects form the basic building blocks of a data model. This chapter describes how to use DataAtlas Modeler to create these basic building blocks.

Creating a Data Model

You can create a data model, in one of the following ways:

- Create a new data model, and then the entities, relationships, and attributes it contains.
- Copy an existing model, then add new objects, and modify or delete existing ones.

Creating a New Data Model

To create a new data model:

1. Open the data models view of DataAtlas Modeler.
2. Do one of the following:
   - Move the mouse pointer to an empty area in the view. Press mouse button 2 to display the pop-up menu for the view. In the pop-up menu, click **Create**.
   - Move the mouse pointer to an existing data model. Press mouse button 2 to display the pop-up menu for the object. In the object pop-up menu, click **Create another**.

   The Create Data Model window opens.
3. Specify a name for the data model and select a TeamConnection component from the list to which the data model is to belong. Click **OK** to create the new model.

   The new model is added to the list of data models.

Copying an Existing Data Model

When you create a data model, you might have an existing data model that you want to use as a basis for the new model.

To copy an existing data model:

1. Open the data models view of DataAtlas Modeler.
2. Move the mouse pointer to the data model you want to copy. Press mouse button 2 to display the pop-up menu for the object. In the object pop-up menu, click **Copy**.

   The Copy Data Model window opens.
3. Specify a name for the data model and select a TeamConnection component from the list to which the data model is to belong. Click **OK** to create the new model.

   If the data model you are copying is already loaded, DataAtlas Modeler copies the last version of the model stored in TeamConnection. The new model is added to the list of data models.

Creating an Entity

You can create entities from the tree view of a data model or from the diagram view of an entity group:

1. Lock the entity group where the entity is to be placed.
2. In the tree view, select the entity group where the entity is to be placed. In the **Selected** menu, click **Create entity**.

   The Create Entity window opens. Figure 17 shows a Create Entity window.

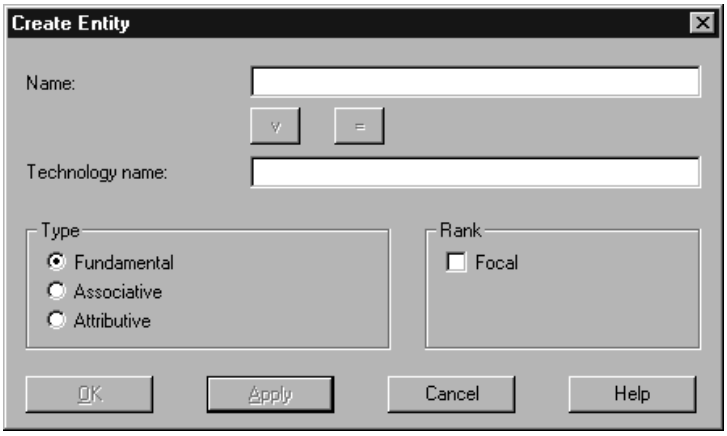


Figure 17. Create Entity Window

3. In the diagram view, do one of the following:

- Click a free position in the diagram where the entity is to be placed, and in the **Group** menu, click **Create entity**.
- Select an existing entity, and in the **Selected** menu, click **Create another**.

The Create Entity window opens.

4. You must specify either a name or a technology name for the entity.

   Click = to copy the name from one name field to another; the target name field must be blank. Click v to generate the technology name from the object name using the predefined technology name generation rules. For further information on specifying these rules, see "Specifying Rules for Generating Technology Names" on page 46.

   You can specify how the entity name appears in the diagram view by using an ampersand (&) to indicate where a new line in the name is to begin. For example, if you want the entity name Service Representative to appear on two lines like this:

   ```
      Service
   Representative
   ```

   In the **Name** field, type: **Service&Representative**.

   If you leave either the entity name or the technology name blank, DataAtlas Modeler automatically generates the missing name when you close the window.

5. Select an entity type.
6. If the entity is the main entity in the entity group, click **Focal**.
7. If you want to create several entities, click **Apply** to keep the Create Entity window open. The new entities do not appear in the view until you close the window.
8. Click **OK** to close the Create Entity window and create the entity.

   The newly created entity is automatically locked. If you created the entity in the tree view or used the **Create another** choice in the diagram view, the new entity is placed at the top of the ER diagram in the left-hand corner. From here, you can move the entity to a new position in the ER diagram. For further information about moving objects in a diagram view, see "Moving Objects" on page 44.

Creating a Relationship

In DataAtlas Modeler you can create unary and binary relationships between entities in the same or in different entity groups. To model a relationship with a degree greater than 2, use an associative entity.

You can also use associative entities to model many-to-many relationships or relationships with attributes. Because these relationships result in additional tables in a relational design, the associative entity can help you to establish a visual link to the relational design.

An isa bundle is a group of relationships between supertypes and subtypes. For information on how to create an isa bundle, see “Creating an Isa Bundle” on page 66.

Between Entities in the Same Entity Group

To create a unary relationship or a binary relationship between two entities in the same entity group:

1. Open the diagram view of the entity group.
2. Lock the entity group.
3. Select at least one of the entities that is to participate in the relationship.
4. In the **Selected** menu, click **Create relationship**.

   If you selected only one entity, DataAtlas Modeler highlights all the entities in the diagram; click the partner entity from those highlighted. If you are creating a unary relationship, click the entity again.

   The Create Relationship window opens.
5. Specify a cardinality for the relationship in both directions.

   The cardinality determines the direction of the foreign key propagation. For information about key propagation in DataAtlas Modeler, see “Appendix B. How DataAtlas Modeler Propagates Keys” on page 127.

   All other information in the window is optional. If you do not specify a name for the relationship, DataAtlas Modeler creates a name by concatenating the names of the entities participating in the relationship and Verb1.

   If you do not specify a technology name, DataAtlas Modeler automatically generates one using the predefined technology name generation rules. For further information on specifying these rules, see “Specifying Rules for Generating Technology Names” on page 46.

   Figure 18 on page 57 shows a completed Create Relationship window.

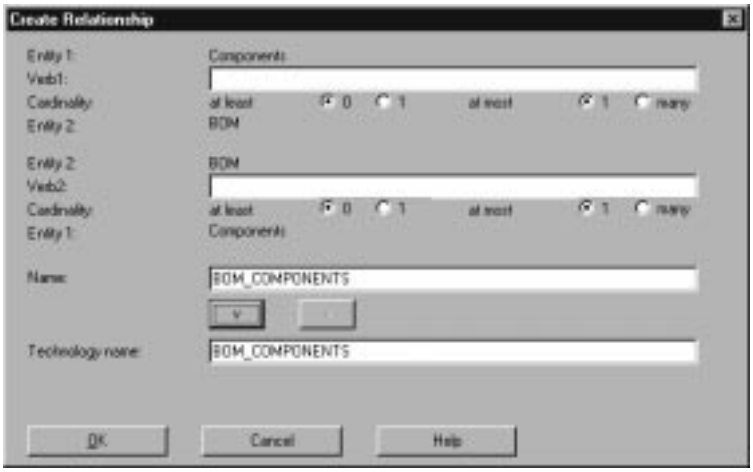


Figure 18. Create Relationship Window

6. Click **OK** to create the relationship.

   The relationship appears in the diagram view. If the relationship has verbs, these are also shown. The newly created relationship is automatically locked.

To create a second relationship between two entities:

1. Open the diagram view of the entity group.
2. Lock the entity group.
3. Select a relationship.
4. In the **Selected** menu, click **Create another**.

   The Create Relationship window opens.
5. Fill in the window and click **OK**.

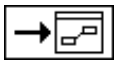
   The relationship and its verbs appear in the diagram view.

Between Entities in Different Entity Groups

To create a relationship between two entities that are in different entity groups:

1. Lock the entity groups that contain the entities that are to participate in the relationship.
2. Open the diagram view of one of the entity groups.
3. Select the entity that is to participate in the relationship.
4. In the **Selected** menu, click **Create relationship**.

DataAtlas Modeler highlights all the entities in the diagram, and displays the **Partner Entity** button at the right-hand side of the diagram:



5. Click the **Partner Entity** button.

   The Partner Entity window opens.
6. Click the entity group containing the partner entity and the partner entity.
7. Click **OK**.

   The Create Relationship window opens.
8. Fill in the window and click **OK**.

   The relationship and its verbs appear in the diagram view. The newly created relationship is automatically locked.

   If both entities belong to different sibling entity groups, the second entity is shown as a ghost entity in the current diagram view.

   If the second entity belongs to a child entity group of the current entity group, the relationship is drawn only to the entity group where the second entity resides.

Adding Data Elements to a Data Model

Data elements are shareable throughout TeamConnection applications. Usually, you already have a pool of data elements in the TeamConnection database that have been defined for your enterprise. You generally only need a small number of these for a data model. You must add the data elements you need to your data model or create your own shareable data elements in DataAtlas Modeler.

To add data elements to a model:

1. Open the tree view of the data model.
2. In the **Model** menu, click **Add data elements**.

   The Add Data Elements window opens. Figure 19 on page 59 shows the Add Data Elements window.

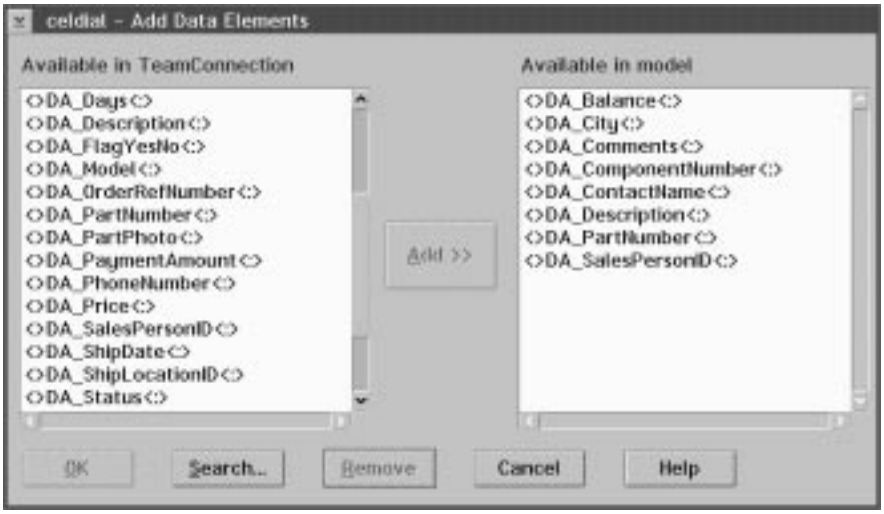


Figure 19. Add Data Elements Window

3. Click **Search** to search for data elements in TeamConnection.

   The **Search Data Elements in TeamConnection** window opens.
4. Enter your search criteria; the fields are case sensitive. You can use wildcard characters in any of the fields; use an underscore (_) to represent one character and a percent sign (%) to represent several characters.
5. Click **Search**.

   The search results appear in the Add Data Elements window in the **Available in TeamConnection** list.
6. Select a data element from the search results list, then click **Add >>** to add it to the data model.
7. When you have finished, click **Cancel** to leave the window.

   The tree view of the data model is redisplayed with the data elements assigned to the model.

Tip

The data elements added to the data model are not automatically updated in DataAtlas Modeler when changes are made to them elsewhere in TeamConnection. To see these changes, you must refresh the tree view.

Creating a Data Element

If you cannot find a suitable data element in TeamConnection, you can create data elements in DataAtlas Modeler. The new data elements are added to the data model and to the TeamConnection database.

To create a data element:

1. Open the tree view of the data model.
2. Do one of the following:
   - In the **Model** menu, click **Create data element**.
   - Click either the **Data elements** node or a data element, and in the **Selected** menu, click **Create data element**.

   The Create Data Element window opens. Figure 20 on page 61 shows the Create Data Element window.

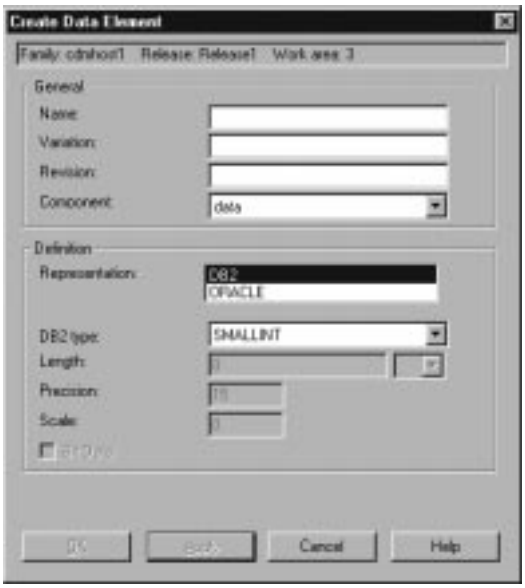


Figure 20. Create Data Element Window

3. Fill in the general information about the data element. When you select a representation type, fields specific to the type are shown in the **Definition** group. In addition, the data type you select determines the availability of the other fields.
4. Click **OK** to create the data element and add it to TeamConnection. The newly created data element is automatically locked.

Creating an Attribute

To create an attribute for an entity or a relationship:

1. Open the attributes view of the entity or relationship.
2. Lock the entity or relationship.
3. Do one of the following:
   - In the **Entity** or **Relationship** menu, click **Create attribute**.
   - Select an attribute, and in the **Selected** menu, click **Create another**.

   The Create Attribute window opens. Figure 21 on page 62 shows the Create Attribute window.

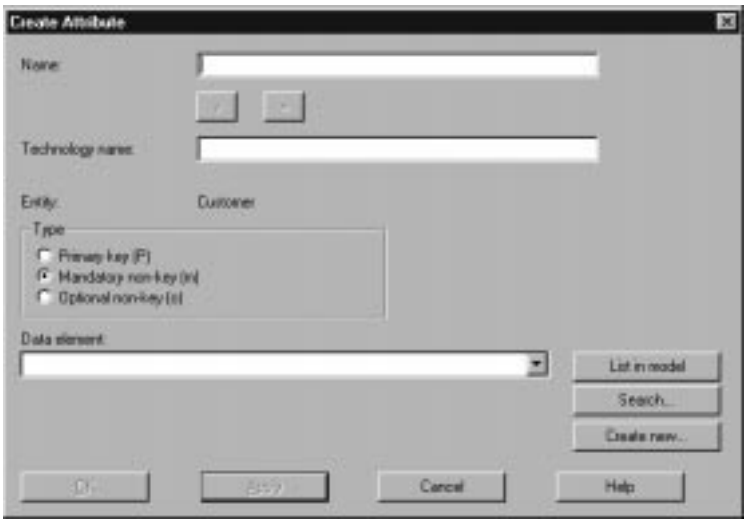


Figure 21. Create Attribute Window

4. You must specify either a name or a technology name for the attribute.

   Click = to copy the name from one name field to another; the target name field must be blank. Click v to generate the technology name from the object name using the predefined technology name generation rules. For further information on specifying these rules, see "Specifying Rules for Generating Technology Names" on page 46.

5. Select an attribute type.
6. Select a data element from the list.

   To reduce the number of data elements in the list, enter a search pattern in the **Data element** field. This field is case sensitive. You can include the wildcard characters % (percent) or * (asterisk) to represent several characters. Only those data elements matching the search pattern are included in the list. For example, if you know the data element you need begins with DA_C, you can type `DA_C%` or `DA_C*` to limit the data elements in the list to those that begin with DA_C.

   If you cannot find a suitable data element in the list of data elements in the model, click **Search** to search for data elements in TeamConnection. If you still cannot find a suitable data element, click **Create new**.

   You do not need to assign a data element to the attribute when you create it.

7. Click **OK** to create the attribute.

   The attributes view is redisplayed with information about the new attribute added to it.

To model a multivalued attribute, create an attributive entity for the attribute. For information about multivalued attributes, see to “Attributive Entity” on page 3 .

Assigning a Data Element to an Attribute

When you create an attribute, you do not have to assign a data element to it straight away. To assign a data element to an attribute at a later stage:

1. Open the attributes view of the entity or relationship.
2. Lock the entity or relationship.
3. Select an attribute.
4. In the **Selected** menu, click **Open as**, then **Settings**.

   The Settings window for the attribute opens.
5. Select a data element from the list of data elements.

   If you cannot find a suitable data element in the list of data elements in the model, click **Search** to search for data elements in TeamConnection. If you still cannot find a suitable data element, click **Create new**.
6. Click **OK** to assign the data element to the attribute.

   The attributes view is redisplayed with the data element information.

Creating a Constraint

To create a constraint for an entity or a relationship:

1. Open the attributes view of the entity or relationship.
2. Lock the entity or relationship.
3. Do one of the following:
   - In the **Entity** or **Relationship** menu, click **Create constraint**.
   - Select a constraint, and in the **Selected** menu, click **Create another**.

   The Create Constraint window for a constraint opens. Figure 22 on page 64 shows the Create Constraint window.

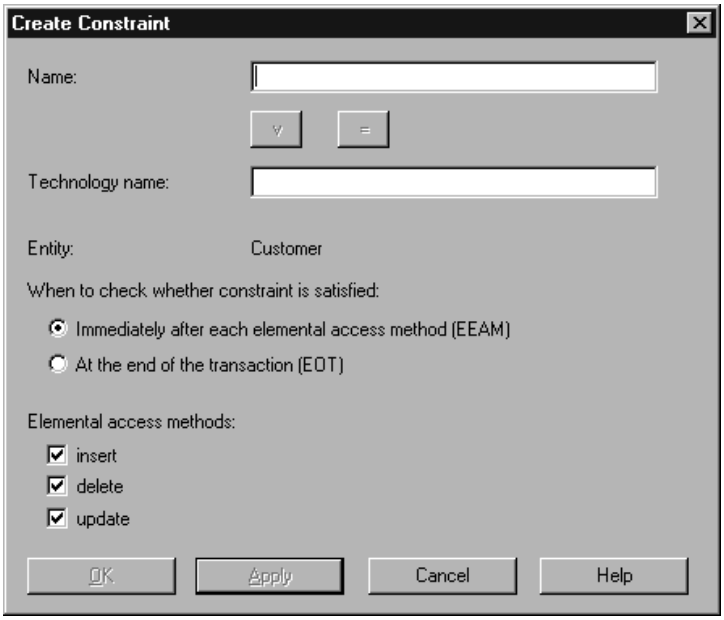


Figure 22. Create Constraint Window

4. Fill in the window and click **OK** to create the constraint.

   The attributes view of the entity is redisplayed with the constraint information.

Working with Entity Groups

Entity groups provide a way of grouping together entities that belong together. For example, you might want to group together all the entities involved in order processing into an entity group called OrderProcessing. You can create entity groups in the tree view of a data model and in the diagram view of an entity group.

Creating an Entity Group

To create an entity group:

1. Lock the entity group containing the entities you want to group together.
2. Select the entities.
3. In the **Selected** menu, click **Group**.

   The Create Entity Group window opens.
4. You must specify either a name or a technology name for the entity group.

Use the = button to copy the name from one name field to another; the target name field must be blank. Use the v button to generate the technology name from the object name using the predefined technology name generation rules. For further information on specifying these rules, see “Specifying Rules for Generating Technology Names” on page 46.

You can specify how the name appears in the diagram view by using an ampersand (&) to indicate where a new line in the name is to begin. For example, if you want the entity group name Order Processing to appear on two lines like this:

```
  Order
Processing
```

In the **Name** field, type: **Order&Processing**.

5. Click **OK** to create the entity group.

   In the tree view, the entity group is added as a new hierarchical level. In the diagram view, the entities are replaced by the entity group.

Ungrouping an Entity Group

You can ungroup an entity group from the tree view of a data model or from the diagram view of the next higher entity group.

To ungroup an entity group:

1. Lock the entity group that contains the entity group you want to ungroup.
2. Select the entity group.
3. In the **Selected** menu, click **Ungroup**.

   The objects in the entity group are moved to the entity group on the next higher level.

Moving a Ghost Entity to an Entity Group

To move a ghost entity to an open entity group:

1. Lock the target entity group and the parent entity group of the ghost entity.

   Ensure that the ghost entity is not locked by anyone else. The locking information is shown in the Information Area when you select the ghost entity.
2. Select the ghost entity.
3. In the **Selected** menu, click **Get entity**.

   The ghost entity is moved to the current entity group.

Moving Entities between Entity Groups

To move entities between entity groups:

1. Open the diagram view of the target entity group.
2. Lock the target entity group.
3. Click a free position in the diagram.
4. In the **Group** menu, click **Get entity**.

   A list of the entities that you can move to this entity group appears. Only entities that are in entity groups that are not locked by other users appear in this list.
5. Select the entities you want to move and then click **OK**.

Generalizing and Specializing

Generalization and specialization involve relating entities using an isa bundle. For general information about generalization and specialization, see "Generalization and Specialization" on page 9.

Creating an Isa Bundle

To create an isa bundle:

1. Open the diagram view of the entity group.
2. Lock the entity group.
3. Click the entity that is to be the supertype.
4. In the **Selected** menu, click **Specialize**.

   The Create Isa Bundle window opens.
5. Fill in the window:
   - Specify the name of the isa bundle.
   - Click the existence rules that apply.
   - Select a subtype from the list of available entities, and then click >> to include it in the isa bundle. Repeat this step for each of the subtypes you want to add to the isa bundle.

   Figure 23 on page 67 shows a Create Isa Bundle window.

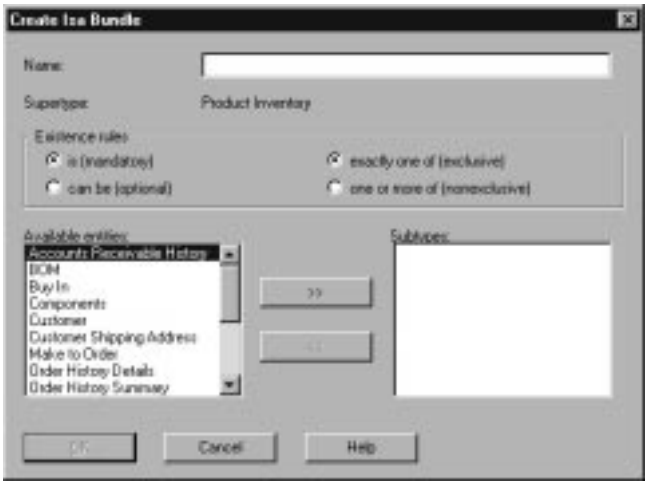


Figure 23. Create Isa Bundle Window

6. Click **OK** to create the isa bundle.

   The newly created isa bundle is automatically locked.

Adding a Subtype to an Isa Bundle

To add a subtype to an existing isa bundle:

1. Open the diagram view of the entity group.
2. Lock the isa bundle and entity group to which it belongs.
3. Click the isa bundle.
4. In the **Selected** menu, click **Add subtype**.

   The Add a Subtype window opens. Figure 24 on page 68 shows the Add a Subtype window.

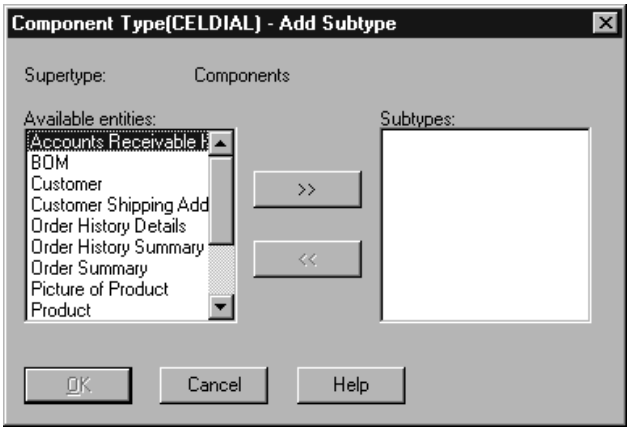


Figure 24. Add a Subtype Window

5. Click entity in the list of available entities that you want to add to the isa bundle, and then click >> to include it in the subtypes list.
6. Click **OK**.

   The entity is added as a subtype to the isa bundle.

Removing a Subtype from an Isa Bundle

To remove a subtype from an isa bundle:

1. Open the diagram view of the entity group.
2. Lock the isa bundle and the entity group to which it belongs.
3. Click the isa relationship connected to the subtype you want to remove.
4. In the **Selected** menu, click **Delete**.

   The subtype is removed from the isa bundle.

Describing Objects in the Data Model

You can create a description for any of the following objects:

- Data model
- Entity group
- Entity
- Relationship
- Attribute
- Constraint
- Isa bundle

- Data element

This description can be included in any document created in DataAtlas Modeler.

To create a description for an object, do the following:

1. If the object is:
   - An entity group, entity, relationship, isa bundle, or data element, lock the object.
   - An attribute or a constraint, lock the entity or relationship to which it belongs.
2. Do one of the following:
   - In an open view of the object, in the object menu (the one at the far left of the menu bar), click **Open as**, then **Text**.
   - Click the object, and in the **Selected** menu, click **Open as**, then **Text**.

   The text view of the object opens. If the object is the data model, lock the text view.
3. Enter the description of the object.

   Use the following formatting controls to specify how the text is to be formatted when it is included in a document. Use capital letters for the formatting controls and place them at the beginning of a line. Do not include any other text in the same line as the control.

   .B Justifies text to the right and left.

   .C Centers text.

   .H The text following this control does not appear in a document.

   .L Justifies text to the left.

   .P Begins a new page.

   .R Justifies text to the right.

   .S The text following this control is unformatted; long lines are split. The text appears in a document in a proportional font.

   .U The text following this control is unformatted; long lines are split. The text appears in a document in a monospace font.

   For all of these controls except .H, .P, and .U, DataAtlas Modeler runs the text together removing any additional blanks or new-line characters.

Tip

When you include a text block with formatting controls in a document structure, change the settings for the section containing the text block to **Unformatted and proportional**. For information on how to change the settings for a text block, see "Specifying Settings for a Chapter or Section" on page 109.

4. You can also include text that you have prepared in another text editor and saved as an ASCII file. To do this, do one of the following:
   - Include the following formatting control:

     ```
     .include path\filename.ext
     ```

     For example:

     ```
     .include C:\OS2\README.TXT
     ```

     When you create a document that includes the object, the text in the file appears in the document.
   - In the **Edit** menu, click **Get**.

     The text in the file is added to the text view.
5. Close the text view when you have finished.

Retrieving Text from the Wastebasket

When you delete objects in the data model that have a description, the text is put into the wastebasket in the tree view. The text remains there as long as the data model is loaded.

To use the text in the wastebasket for another object, do the following:

1. Double-click on the text in the wastebasket to open its text view.
2. Do one of the following:
   - Copy the text to the clipboard. Open the text view of the object where you want to use the text. In the **Edit** menu, click **Paste**.
   - Save the text to a file. Open the text view of the object where you want to use the text. In the **Edit** menu, click **Get**.

   The text from the wastebasket is added to the current text view.

Validating Objects in a Data Model

DataAtlas Modeler helps you check whether your data model is complete and follows the rules imposed by the entity-relationship modeling method. You can specify which rules are to be invoked during validation, for example, that all entity groups must have a description or that all entities must have attributes assigned.

You can validate your data model as often as you like and at any stage of its development. You can also validate a model before you transform it into a relational design. For a description of the validation rules for transformation, see "Validating Objects in a Data Model for Transformation" on page 73.

You can validate the following objects:

- Data model
- Entity group
- Entity
- Relationship
- Isa bundle
- Data element

1. To validate all the objects in a view, in the object menu (the one at the far left of the menu bar), click **Validate**.
2. To validate one or more objects in a view, select the objects, and in the **Selected** menu, click **Validate**.
3. In the Validate window, click the rules DataAtlas Modeler is to use to validate the selected objects.

   For example, you want to validate all the entities in an entity group to ensure that:

   - All entities have descriptive text.
   - All entities have attributes assigned to them.
   - All attributes have data elements assigned to them.
   - Each entity has a primary key defined, it is sufficient to uniquely identify each instance of the entity, and it is not redundant.

   The Validate window looks like Figure 25 on page 72:

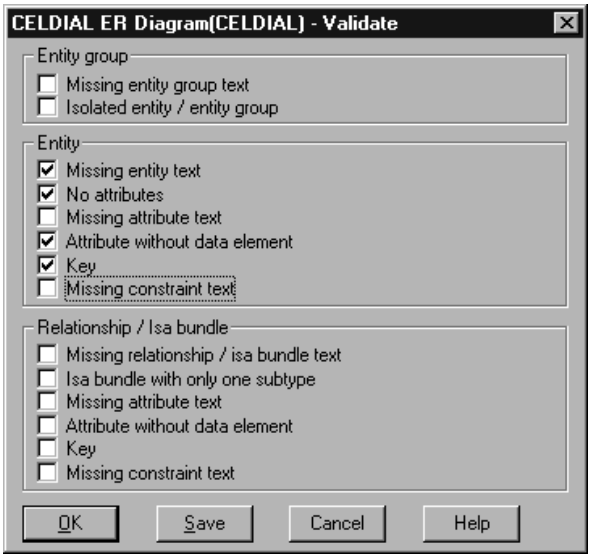


Figure 25. Completed Validate Window for Entity Group Validation

4. Click **OK** to start the validation.

   If objects are found that do not comply with the selected validation rules, the Validation Object List containing these objects opens.

5. Click an object in the list and then **OK**.

   If the object is an entity group, entity, or relationship, DataAtlas Modeler opens a view containing the object and displays a list of validation messages associated with the object. For example, if you select an entity from the list, the attributes view of the entity opens.

   For any other object, the object is selected in the view, and the list of validation messages associated with the object is displayed.

6. Click **OK**.

   DataAtlas Modeler closes the message window.

7. Use entries in the **Messages** menu to display the messages for the view again.

8. To work on messages for other objects, return to the view where you started the validation, and in the **Messages** menu, click **Message list**.

   The Validation Object List is shown again.

Chapter 6. Forward and Reverse Engineering

DataAtlas Modeler supports both the forward engineering of objects in data models to relational designs and the reverse engineering of relational designs to objects in a data model. In DataAtlas Modeler, this data engineering process is called *transformation*.

During forward engineering, DataAtlas Modeler creates a relational design and instances of tables from a data model. During reverse engineering, DataAtlas Modeler creates a data model from a relational design.

This chapter describes how to carry out forward and reverse engineering in DataAtlas Modeler and how objects are transformed in both cases.

Forward Engineering

DataAtlas Modeler supports transformation into relational designs for DB2 Universal Database, DB2 for OS/390, and Oracle. The transformation produces a relational design containing table definitions, and instances of tables for the specified target database management system (DBMS).

You can transform all of the objects in a data model, or select some of the objects in the model and do a partial transformation on only these objects. This partial transformation can be useful, for example, if your databases run on both DB2 for OS/390 and DB2 Universal Database. You can use the same data model to create system-specific relational designs and table instances.

Before you transform objects in a data model you can validate them to ensure that they meet the requirements of the target DBMS.

Validating Objects in a Data Model for Transformation

If you validate objects in a data model before you transform them, it can help you to resolve design problems that might otherwise cause problems during transformation. The validation checks that:

- Object names in the data model will produce valid table names according to SQL naming conventions for the target DBMS. This includes checking the name length, the valid characters, and the reserved words.
- Every entity has a primary key that is sufficient to uniquely identify each instance of the entity, and that it is not redundant.

- Every entity has attributes defined so that columns and column definitions can be created for the table instance and the table definition.
- Every attribute has a data element assigned to it, otherwise a complete definition of the column cannot be created.

Validate objects in a data model from the tree view of the data model:

1. Do one of the following:
   - To validate all the objects in the model, in the **Model** menu, click **Transform into**, then **Validate for transform**, and then the target DBMS.
   - To validate selected objects, in the **Selected** menu, click **Transform into**, then **Validate for transform**, and then the target DBMS.

   The Validate window for the target DBMS opens.
2. Click the rules DataAtlas Modeler is to use to validate the objects, and then click **OK** to start the validation.

   If objects are found that do not comply with the selected validation rules, the Validation Object List containing these objects opens.
3. Select an object from the list and then click **OK**.

   If the object is an entity group, entity, or relationship, DataAtlas Modeler opens a view containing the object and displays a list of validation messages associated with the object. For example, if you select an entity from the list, the attributes view of the entity opens.

   For any other object, the object is selected in the view and the list of validation messages associated with the object is displayed.
4. Click **OK**.

   DataAtlas Modeler closes the message window.
5. Use entries in the **Messages** menu to display the messages for the view again.
6. To work on messages for other objects, return to the view where you started the validation, and in the **Messages** menu click **Message list**.

   The Validation Object List is shown again.

Transforming Objects in a Data Model into a Relational Design

You can transform objects in a data model into a new relational design or into an existing relational design that resulted from a previous transformation.

To transform objects in a data model a relational design:

1. Open the tree view of a data model.
2. Do one of the following:

- To transform all the objects in the model, in the **Model** menu, click **Transform into**, then **Relational design**, and then the target DBMS.
- To transform only some of the objects:
  a. Select the entities. For example, to transform all the entities in an entity group, press and hold down the Shift key, and click on the entity group to select all its entities.

     You do not need to select relationships; all relationships between selected entities are automatically transformed. However, if you use associative entities to represent many-to-many relationships or relationships with attributes, you must also select the associative entities.

     Tip

     If you are carrying out a partial transformation into an existing relational design, the updates made to the relational design depend upon the objects you select. Table 9 on page 83 shows the objects you must select in the data model so that the relational design is correctly updated.

  b. In the **Selected** menu, click **Transform into**, then **Relational design**, and then the target DBMS.

The Transform into Relational Design window for the target DBMS opens. Figure 26 on page 76 shows an example of a Transform into Relational Design window.

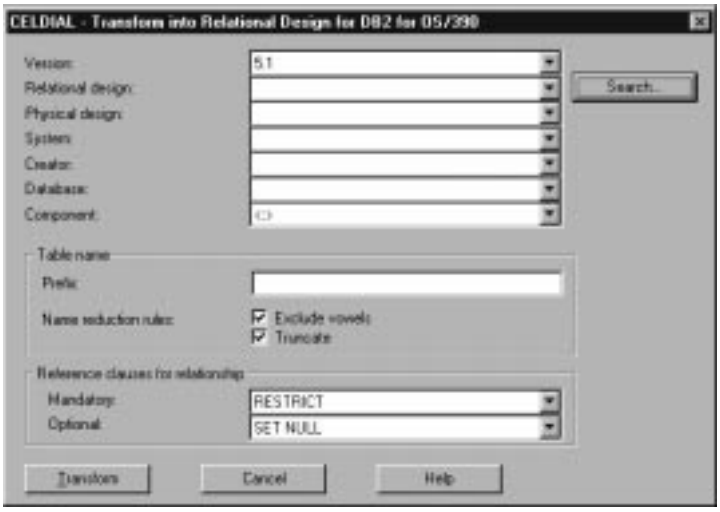


Figure 26. Example of a Transform into Relational Design Window

If the objects have already been transformed into a relational design, or the data model resulted from the transformation from a relational design, the settings specified for the last transformation are shown in the window.

3. Specify the transformation options.

   The version, relational design, system, component, and creator or schema are mandatory fields. If you are transforming into a relational design for DB2 Universal Database, the database field is also mandatory. If you specify an object name that does not already exist in the TeamConnection database, a new object is created. This does not apply to the version or the component.

   - Relational design

     If the objects in the data model have already been transformed into a relational design, the **Relational design** field contains a list of these designs. Select a design from this list or click **Search** to search all the relational designs in TeamConnection. When you select a relational design, the settings for the last transformation into this relational design are shown in the window.

     If you specify a new relational design name, a relational design with its corresponding collection of table definitions and tables is created.

   - Physical design

     The physical design associated with the selected relational design is shown in the **Physical design** field. If several physical designs are

associated with the relational design, the physical design resulting from the last transformation is shown first.

If you retransform objects into an existing relational design but you specify a new physical design name, the tables created during the previous transformation are linked to the new physical design.

- System

  The systems available for the selected version.

  If you retransform objects into an existing relational design but you specify a different system, a new set of tables is created. These tables are linked to the table definitions created during the previous transformation.

  For example, a data model contains the entity *Department* with the technology name *DEPT*. If you transform the data model into a relational design for the DB2 for OS/390 system *SYS1*, DataAtlas Modeler creates a table definition *Department* and a DB2 for OS/390 table *SYS1:ADM:::DEPT*. If you retransform the data model into the same relational design but change the system name to *SYS2*, DataAtlas Modeler creates a new DB2 for OS/390 table *SYS2:ADM:::DEPT* and links this to the existing table definition *Department*.

- Creator or Schema, Database

  These fields list, where appropriate, the creators, schemas, and databases available for the selected system.

4. Click **Transform**.

   DataAtlas Modeler starts the transformation. The objects are transformed as described in"Transformation into a New Relational Design" on page 78 and "Transformation into an Existing Relational Design" on page 80. If you do a partial transformation on selected objects, the following also apply:

   - All relationships between selected entities are automatically transformed.
   - All attributes of selected entities are transformed.
   - If you select a subtype in an isa bundle but not its corresponding supertype, both the subtype and the supertype are transformed.

   You can stop the transformation at any time, and then resume it or cancel it altogether. When the transformation is finished, a report documenting the transformation is displayed. You can save this report to a file for printing.

5. If the transformation of an object results in a table or column name that violates the naming rules or in a table name that is too long, a Name Conflict window opens. Do one of the following:

   - Assign a new name to the table or column and click **OK**.

- Click **Cancel** to end the transformation.

Transformation into a New Relational Design

When you transform objects into a new relational design, DataAtlas Modeler creates a collection of table definitions and tables for the specified target DBMS. It also creates a set of mapping objects with information about how each object is transformed. Table 6 shows the mapping information for each data model object and its corresponding relational design object:

Table 6. Mapping Objects

Data Model Object	Relational Design Object
Entity	Table definition, table
Attribute	Column definition
One-to-one or one-to-many relationship without attributes	Foreign key definition
Many-to-many relationship or relationship with attributes	Table definition, table, foreign key definition, column definitions

If you transform objects into an existing relational design, DataAtlas Modeler uses the mapping objects as the basis for the transformation. During transformation, the mapping objects are updated with changes to the data model objects.

When you transform objects into a new relational design, DataAtlas Modeler uses the following rules to transform the objects.

Entities

Entities are transformed into new table definitions and new tables. The entity name becomes the table definition name and the entity technology name becomes the target DBMS table name. For example, when you transform a data model containing the entity *Department* with the technology name *DEPT* into a relational design for DB2 for OS/390, DataAtlas Modeler creates a table definition *Department* and a DB2 for OS/390 table *DEPT*.

If a table definition name or table name already exists in the TeamConnection database, DataAtlas Modeler makes the name unique by adding a number to it. For example, if the data model contains the entity *Employee* and the table definition *Employee* already exists in the TeamConnection database, during transformation DataAtlas Modeler creates the table definition *Employee1*.

Attributes of Entities and Relationships

Attributes of entities and relationships are transformed into new column definitions:

- The attribute technology name becomes the column definition name.
- The shared data element associated with the attribute is associated with the column definition.
- The entity primary key becomes the primary key definition of the table definition.

Unary and Binary Relationships

Unary and binary relationships are transformed into new foreign key definitions or additional new table definitions and tables:

- For one-to-one or one-to-many relationships without attributes, the relationship is transformed into a foreign key definition according to the DataAtlas Modeler key propagation rules. For further information on key propagation, see "Appendix B. How DataAtlas Modeler Propagates Keys" on page 127.
- For many-to-many relationships without attributes, and all relationships with attributes, the relationship is transformed into a new table definition and a new table (a correlation table).

  Foreign keys are created in the new table definition from the primary keys of the entities participating in the relationship. If the primary key of an entity is a composite key, the corresponding foreign key is also a composite key.
- For a mandatory one-to-one relationship, the unique attribute is set to YES. For all other types of relationships, the unique attribute is set to NO.

Constraint: DataAtlas Modeler also creates a constraint name based on the relationship technology name. The maximum length of the constraint name is 8 characters. The relationship technology name is shortened as follows:

1. Remove characters that are not valid SQL characters for the target DBMS.
2. Remove vowels starting at the end of the technology name.
3. Truncate the technology name.

If the resulting constraint name is not unique within the table definition, DataAtlas Modeler adds a number to the name.

Reference Clauses: During transformation, DataAtlas Modeler sets the delete rules and update rules for the referential integrity of the foreign key. The default setting depends on the minimum cardinality of the relationship of the dependent entity to the parent entity. The *dependent entity* is the entity to where the foreign key is passed from the parent entity.

DataAtlas Modeler sets the default delete rules and update rules for the target DBMS as shown in Table 7 on page 80. You can change these default settings

in the Transform to Relational Design window before you transform the data model.

Table 7. Default Settings for References Clauses

Relationship in Data Model	DB2 Universal Database	DB2 for OS/390	Oracle
Dependent entity has optional relationship to its parent.	Delete rule: SET NULL Update rule: NONE	Delete rule: SET NULL Update rule: N/A	N/A
Dependent entity has a mandatory relationship to its parent.	Delete rule: RESTRICT Update rule: RESTRICT	Delete rule: RESTRICT Update rule: N/A	Delete rule: NONE Update rule: N/A
Relationship with attributes or many-to-many relationship.	Delete rule: RESTRICT Update rule: RESTRICT	Delete rule: RESTRICT Update rule: N/A	Delete rule: RESTRICT Update rule: N/A

Isa Bundles

Isa bundles are transformed into table definitions and tables. DataAtlas Modeler creates a table definition and a table for the supertype and for each of the subtypes. The isa relationships are transformed in the same way as binary relationships.

Transformation into an Existing Relational Design

When you transform objects into an existing relational design, DataAtlas Modeler uses the mapping objects to determine the changes made to the data model since the last time you transformed it.

Note: If you have made changes to the relational design in DataAtlas Designer or DataAtlas Dictionary, these changes are not affected by the transformation. For example, if you have added a new column to a table, the column is unaffected when you retransform the data model containing the corresponding entity into the same relational design.

If you added new objects to the data model, DataAtlas Modeler creates new relational design objects as described in “Transformation into a New Relational Design” on page 78. If you modified objects, Table 8 on page 81

shows how DataAtlas Modeler transforms these modified objects:

Table 8. Actions in the Data Model and Their Results in the Relational Design

Action in Data Model	Result in Relational Design
On entities	
Change name	Table definition renamed.
Change technology name	Table renamed.
Delete **Note:** When you delete an entity, the attributes and relationships that belong to the entity are also deleted.	Corresponding table definition and all tables that use the table definition are deleted. All parts that belong to the table definition, such as the column definition and the foreign key definition, are also deleted with the table definition. **Note:** This is applicable only when you transform all the objects in the model.
On attributes	
Change name	No effect.
Change technology name	Column renamed.
Change attribute type from optional to mandatory	Column definition set to NOT NULL.
Change attribute type from mandatory to optional	Column definition set to NOT NULL.
Delete	Column definition deleted.
Assign new data element	Link from the column definition to the original data element deleted and link to the new data element created.
On primary keys	
Add attribute	Column definition added to primary key definition.
Remove attribute	Column definition removed from primary key definition. If this column is the only column in the primary key definition, the primary key definition is deleted.
On relationships	
Change name	No effect.
Change technology name	Constraint renamed.
Add attribute to primary key of parent entity	Column definition added to foreign key definition of the dependent table and to the primary key definition of the parent table. **Note:** For a partial transformation, the column definition is deleted only if the dependent entity is also selected.

Table 8. Actions in the Data Model and Their Results in the Relational Design (continued)

Action in Data Model	Result in Relational Design
Remove attribute from primary key of parent entity	Column definition removed from foreign key definition in the dependent table and from the primary key definition of the parent table. If this column is the only column in the foreign key definition, the foreign key definition is deleted.
Delete	For one-to-one and one-to-many relationships, the column definition of the foreign key definition of the dependent table and the foreign key definition are deleted. For many-to-many relationships and relationships with attributes, the correlation table is deleted.
On relationship cardinality	
Change one-to-one to one-to-many	Unique attribute set to NO.
Change one-to-many to mandatory one-to-one	Unique attribute set to YES.
Set relationship to the parent entity to optional	Default delete rule for the foreign key set to ON DELETE SET NULL.
Set relationship to the parent entity to mandatory	Default delete rule for the foreign key set to ON DELETE RESTRICT.
Change a one-to-one or one-to-many relationship to a many-to-many relationship or to a relationship with attributes	1. Column definitions of the foreign key definition of the dependent table deleted. 2. Foreign key definition of the dependent table deleted. 3. New correlation table added. 4. Foreign key definitions added to the correlation table that references the participating tables.

If you are carrying out a partial transformation into an existing relational design, the updates made to the relational design depend upon the actions you carried out in the data model and the objects you select for transformation. To ensure that the relational design correctly reflects the actions on the data model, it is recommended that for each action in the data model, you include the objects shown in Table 9 on page 83 in your set of

selected objects.

Table 9. Objects to Be Selected for Partial Transformation

Action in Data Model	Selected Objects
On entities	
Change name	Entity
Change technology name	Entity
Create	Entity
Delete	All objects. Start transformation from the **Model→Transform into** menu choice.
On primary-key attributes	
Change name	Entity
Change technology name	Entity and all dependent entities[1]
Change attribute type to optional non-key	Entity and all dependent entities[1]
Change attribute type to mandatory non-key	Entity and all dependent entities[1]
Change the order of the primary-key attributes	Entity and all dependent entities[1]
Change data type	Entity and all dependent entities[1]
Create	Entity and all dependent entities[1]
Delete	Entity and all dependent entities[1]
On optional non-key attributes	
Change name	Entity
Change technology name	Entity
Change attribute type to primary key	Entity and all dependent entities[1]
Change attribute type to mandatory non-key	Entity
Change the order of the optional non-key attributes	Entity
Change data type	Entity
Create	Entity
Delete	Entity
On mandatory non-key attributes	
Change name	Entity
Change technology name	Entity
Change attribute type to primary key	Entity and all dependent entities[1]
Change attribute type to mandatory non-key	Entity
Change the order of the mandatory non-key attributes	Entity
Change data type	Entity
Create	Entity

Table 9. Objects to Be Selected for Partial Transformation (continued)

Action in Data Model	Selected Objects
Delete	Entity
On relationships	
Change name	Source and target entity
Change technology name	Source and target entity
Change cardinality from many-to-many to one-to-many or one-to-one	All objects. Start transformation from the **Model→Transform into** menu choice.
Change cardinality from one-to-one to one-to-many or many-to-many	Source and target entity
Change cardinality from one-to-many to one-to-one or many-to-many	Source and target entity
Create	Source and target entity
Delete one-to-one or one-to-many relationship	Source and target entity
Delete many-to-many relationship	All objects. Start transformation from the **Model→Transform into** menu choice.
Remove the last non-key attribute from a relationship with attributes	All objects. Start transformation from the **Model→Transform into** menu choice.
Remove a non-key attribute from a many-to-many relationship	Source and target entity
Add a non-key attribute to a relationship	Source and target entity
On isa bundles	
Change name	Supertype and all subtypes
Change existence rules	Supertype and all subtypes
Create	Supertype and all subtypes
Delete	Supertype and all subtypes
Add subtype	Subtype
Remove subtype	All objects. Start transformation from the **Model→Transform into** menu choice.

Legend:

[1] *All dependent entities* refers to the entities that depend directly on the parent entity, their dependent entities, the dependent entities of those dependent entities, and so on.

Reverse Engineering

If you did not use DataAtlas Modeler to design your existing relational database, you can use it to help you when you want to add new tables. Use DataAtlas Dictionary to populate the TeamConnection database with the existing database catalog. You can then use DataAtlas Modeler to transform the resulting relational design into a data model.

For information on using DataAtlas to populate TeamConnection, refer to the *Dictionary and Designer User's Guide.*

Transforming a Relational Design into a Data Model

To transform a relational design into a data model:

1. Create a new data model object.
2. Open the tree view of the data model.
3. In the **Model** menu, select **Transform from**, then **Relational design**.

   The Transform from Relational Design window opens.
4. Select the relational design that is to be transformed, the source database system, and then click **Transform**.

   DataAtlas Modeler starts the transformation. You can stop the transformation at any time, and then resume it or cancel it altogether.

   When the transformation is finished, a report documenting the transformation is displayed. You can save this report to a file for printing.

Transformation into a Data Model

When you transform a relational design and its corresponding tables into a data model, DataAtlas Modeler transforms:

- A table definition into an entity or a many-to-many relationship

  A table definition is transformed into a many-to-many relationship when the primary key definition of the table definition is made up of two foreign key definitions and there are no other foreign key definitions.

  In all other cases, a table definition is transformed into an entity. The table definition name is transformed into the entity name and the table name in the source DBMS is transformed into the entity technology name.
- A column definition into an attribute of an entity or a relationship

  The column name becomes the attribute technology name. If the relational design you are transforming resulted from a forward transformation of a data model, the attribute retains its original object name. If the relational design was created in any other way, for example, by populating the TeamConnection database with the database catalog, the column name is

also transformed into the attribute name. The data element associated with the column definition is associated with the attribute.

- The primary key definition of the table definition into the primary key for the entity

  The primary key columns are transformed into attributes; the attribute type is set to **P**.

- Foreign key definitions into relationships:
  - When the foreign key definition is also the primary key definition, a one-to-one relationship is created where the dependent entity has a mandatory relationship to the parent entity.
  - When the primary key definition is made up of two foreign key definitions and there are no other foreign key definitions, a many-to-many relationship is created where the dependent entity has an optional relationship to the parent entity.
  - In all other cases, the transformation depends on whether NULL values are allowed for the foreign key and whether the foreign key is unique. Table 10 shows the possible combinations.

Table 10. Rules for the Transformation of Foreign Keys

NULL Value Allowed	Unique Attribute Value	Data Model Relationship
Yes	No	A one-to-many relationship where the dependent entity has an optional relationship to the parent entity.
No	No	A one-to-many relationship where the dependent entity has a mandatory relationship to the parent entity.
No	Yes	A one-to-one relationship where the dependent entity has a mandatory relationship to the parent entity.

Chapter 7. Importing and Exporting Models Using VisualAge Exchange

If you have already modeled your data using a different data modeling tool, VisualAge Exchange provides a bridge from your modeling tool to DataAtlas. The *VisualAge Exchange User's Guides* describe in detail the translation to and from other modeling tools.

Importing a Data Model

If you want to use DataAtlas Modeler to work with models that you created with other data modeling tools, you must:

1. Translate the data model into a CASE model using VisualAge Exchange.
2. Import the CASE model into TeamConnection.
3. Transform the CASE model into a data model using DataAtlas Modeler.

Translating a Data Model into a CASE Model

Ensure that you set the following for the target CASE model in VisualAge Exchange:

COMPONENT
The TeamConnection component to which the translated data model is to belong.

MODEL
The name for the translated model; the name must be unique within TeamConnection.

STRVER
The TeamConnection version context. This has the following format:

```
family|release|work area
```

The family name must be unique for every model you translate.

MAPNAME
The map name; this must be unique for every model you translate.

MAPDOM
The source used to populate the data elements in the CASE model. This parameter determines whether VisualAge Exchange uses defining attributes or domains in the source data model to create data elements in the CASE model. Specify one of the following:

N VisualAge Exchange uses defining attributes in the source data model to create data elements.

If the source modeling tool does not have defining attributes, VisualAge Exchange generates a defining attribute for every attribute in the source model. It then creates the data elements from the generated defining attributes.

Y VisualAge Exchange uses domains in the source data model to create data elements.

VisualAge Exchange generates a domain from the data type and size of every attribute. It then creates the data elements from the generated domains.

VARIATION and REVISION

It is recommended that you set the VARIATION, the REVISION, or both to ensure that the translated data element names are unique in the TeamConnection database. If the names are not unique, you cannot import the translated model into TeamConnection.

TARFILNAM

The name of the file produced by the translation process.

NOTESWANTED

If your data modeling tool supports comment text as well as description text, you can append the comments to the descriptive text in the translated data model.

N Translate only the descriptive text.

Y Append comment text to the descriptive text.

Importing a CASE model into TeamConnection

Before you import a model, ensure that a suitable family, release, and work area exist in the TeamConnection database for the imported model. Use the TeamConnection **work area import** command to import the file produced by the translation process into TeamConnection.

For example, if you want to import the file d:\celdial.imp to work area 3 in Release1, the syntax for the **workarea import** command is:

```
teamc workarea -import 3 -release Release1 -file d:\celdial.imp
```

Transforming a CASE Model into a Data Model

A CASE model is the TeamConnection internal representation of a data model. It is the form used for data models that were created with other data

modeling tools and then imported into TeamConnection. To work with an imported model in DataAtlas Modeler, you must transform it into a DataAtlas Modeler data model.

To transform a CASE model into a data model:

1. Create a new data model object.

   Ensure that you create the new model in the same TeamConnection family, release, and work area as the translated model.
2. Open the tree view of the data model.
3. In the **Model** menu, click **Transform from**, then **CASE model**.

   The Transform from CASE Model window opens.
4. Select the CASE model that is to be transformed and then click **Transform**. You can stop the transformation at any time, and then resume it or cancel it altogether.

   DataAtlas Modeler transforms the constructs in a CASE model into the equivalent DataAtlas Modeler constructs with the following exceptions:

   - N-ary relationships

     N-ary relationships are transformed into associative entities with binary relationships.
   - Relationship constraints

     Relationship constraints specify whether the occurrence of one relationship affects the occurrence of another relationship. DataAtlas Modeler does not transform relationship constraints.

The imported model has only one entity group. Information about how objects were positioned in ER diagrams in the original model is not retained in the imported model.

Imported models are easier to work with in diagram views if you divide the main entity group into several smaller entity subgroups before you start modeling in DataAtlas Modeler. You can then assign default positions to the objects in each of the entity subgroups.

1. Open the tree view of the data model and create entity subgroups.

   For information on how to create entity groups, see“Creating an Entity Group” on page 64.
2. Open the diagram view of each entity group in turn.
3. To assign default positions to the objects in the view, press Ctrl+Alt+F twice.
4. If necessary, you can compress the diagram by pressing and holding down Shift+mouse button 2 and moving the mouse pointer.

5. Make any further manual adjustments to the position of objects.
6. If you have any empty pages in the diagram view, click **Remove pages** in the **View** menu to remove them.

Exporting a Data Model

If you want to use the ER data model you created in DataAtlas Modeler in another data modeling tool, you must:

1. Transform the data model into a CASE model.
2. Export the CASE model from TeamConnection.
3. Translate the exported CASE model into a data model for the source data modeling tool using VisualAge Exchange.

Transforming a Data Model into a CASE Model

To transform a CASE model into a data model:

1. Open the tree view of a data model.
2. In the **Model** menu, click **Transform into**, then **CASE model**.

   The Transform into CASE Model window opens.
3. Specify the name of the CASE model; the name must be unique within the CASE models in TeamConnection.
4. Click **Transform**.

   DataAtlas Modeler starts the transformation. You can stop the transformation at any time, and then resume it or cancel it altogether.

Exporting the CASE Model

Use the TeamConnection **part export** command to export the CASE model to a file.

For example, if you want to export the CASE model celdial in release Release1 and work area 3 to the file d:\celdial.exp, the syntax for the **part export** command is:

```
teamc part -export "<>celdial<:>"
           -file d:\celdial.exp
           -release Release1
           -workarea 3
           -type DSDataModel
           -viewtype CDMPublicCASEModel
```

Translating the CASE Model into a Data Model

Translate the exported file to the format for the target data modeling tool using VisualAge Exchange. The *VisualAge Exchange User's Guides* describe how to translate the exported file.

Chapter 8. Producing Documentation

DataAtlas Modeler allows you to produce documentation for a data model. You can either print a data model view, for example, a diagram view of an entity group, or generate documents containing information about the objects in a model.

To create a document, you need a document profile that defines the content, structure, and format of the document. DataAtlas Modeler contains a default document profile called DEFAULTS, which you can use to create a standard document. For information on how to create customized documents, see "Chapter 9. Customizing Documents" on page 99.

Printing a View of an Object

You can print a view of an object, for example, the diagram view of an entity group. Use the Layout window for the view to specify the orientation, the paper size, and the font used for printing.

To print a view:

1. Do one of the following:
   - Open the view you want to print, and click **Print** in the object menu (the one at the far left of the menu bar).
   - Open a view that contains the object you want to print, select the object, and click **Print** in the **Selected** menu.

   The Print window opens.
2. Specify the number of copies, the print range, and the printer, and click **Print**. The printer can also be a fax printer.

Creating a Document

You can create documents for the complete model or part of the model. The settings you specified in the Layout window for each object or text block are used when you create the document. A default font is used for all other text in the document. You can change these font settings from the Document Preview window after you create the document. For example, in the preview window you can change the font used for the main text of the document.

To create a document:

1. Open the tree view of the data model.

> **Tip**
>
> To ensure that you include the latest data element information in a document, refresh the tree view of the data model before you create the document.

2. Do one of the following:
   - If you want to create a document for all the objects in a model, in the **Model** menu, click **Create document**.

> **Tip**
>
> If you restricted the view using **Display selected only**, or excluded objects from the tree view using the **Exclude** choice, the document includes only those objects contained in the current view.

   - If you want to create a document for only some of the objects in the data model, select the objects you want to include in the document, and in the **Selected** menu, click **Create document**.

     The Create Document window opens.

> **Tip**
>
> You can also restrict the objects described in the document by customizing the document profile. For information on how to restrict the objects in a document using the document profile, see “Excluding and Including Chapters and Sections” on page 111.

Figure 27 on page 95 shows the Create Document window.

Figure 27. Create Document Window

3. Select a document profile and the title page, header, footer, and document structure from this document profile that you want to use for this document. You must select one document element from each type.

   If you decide that you do not want to include a particular document element in the document, you must still select a document element for this type. In this case, the document element is an empty document element. For information on how to create empty document elements, see "Creating an "Empty" Title Page" on page 100 and "Creating an "Empty" Header or Footer" on page 104.

   If a suitable profile does not already exist, click **Create new** to create a new profile or **Copy** to copy and change an existing profile.

4. Documents are not automatically saved. If you want to save the generated document, click **Save**.
5. To preview the document before printing it, click **Preview**.
6. To print the document from this window using the printer defaults, click **Use default values for print window**, and click **Print**.

   To select a printer and the output type before printing the document, click **Print**.

Previewing a Document

DataAtlas Modeler lets you check and fine-tune the format of a document before you print it. For example, you can change the font used for the main text throughout the document.

When you preview a document, you can do the following:

- Navigate through the document to a section or page.
- Change the page settings, fonts, or scaling factors in the document.
- Get information about the document.
- Change the settings for the preview window.
- Save the document.
- Print the document.

Navigating through a Document

A document is divided into the following sections: title page, table of contents, body, and appendix. You can use these sections to navigate through a document, or you can go to a specific page number.

To navigate through a document, click **View** in the menu bar, and do one of the following:

- To go to another page in the document, click **Page**. You can specify the number of the new page relative to the first page in the section or the absolute page in the document.

  The current page number in the section and the absolute page number in the document are shown in the Information Area at the bottom of the window.
- To go to the first page in another section, click the section.

You can also use choices in the tool bar to navigate through the document. For a list of tool-bar choices, see "Appendix C. DataAtlas Modeler Tool Bar Choices" on page 129.

Changing Page Settings, Fonts, and Scaling Factors

In a document preview, you can change certain aspects of the layout before you print the document, for example the font size used for headings, or the size of the graphics. These changes then apply throughout the document.

To change the page settings, fonts, or scaling factors:

1. In the **Document** menu, select **Layout**, then click one of the following:

- **Page**, to change page settings, for example the page size and margins.

  The Page window opens.
- **Font**, to change the font for a particular text type, for example, the font used for the running footers.

  The Fonts window opens.
- **Scaling Factors**, to rescale the graphics.

  The Scaling Factors window opens.

2. Make your changes, and click **OK** to activate them.

Getting Information about a Document

To get information about a document, for example, when the document was created, and the total number of pages it has:

1. In the **Document** menu, click **Information**.

   DataAtlas Modeler displays information about the document.

Changing the Settings for the Preview Window

You can change the way the Preview window looks without affecting the appearance of your finished document. You can change the following settings for the Preview window:

- Borders
- Graphics
- Scroll position

To change the settings for a Preview window:

1. In the **Document** menu, click **Open as settings**.

   The Preview Settings window opens.
2. Make the changes.
3. Click **OK** to change the settings for the window.

Saving a Document

When you create a document and print it, the document is not saved automatically. You might not want to print your document directly from the Preview window, or you might want to save the document in a format that you can use elsewhere.

1. To save a document:
   - In the Create Document window, click **Save**.
   - In the Preview window, click **Save to document file** in the **Document** menu.

The Save Document File window opens.

2. In the Save Document File window, specify the:
   - File path where the file is to be saved to

     If the output format is PostScript, the default path shown in the window enables you to work with the document in the documents view.
   - File name

     If the output type is PCX or TIF, DataAtlas Modeler shortens the name you enter here to 5 characters.
   - Output type

     If you selected a format other than PostScript, DataAtlas Modeler creates a separate file for each page in the document. It adds a number starting with 000 for the title page to the file name for each file it creates.
3. Click **OK** to save the document.

Printing a Document

You can print PostScript documents in DataAtlas Modeler when you create them from the Create Document window, or view them from the Document Preview window or from the documents view.

To print a document from the:

- Create Document window, click **Print**.
- Preview window, click **Print** in the **Document** menu.
- Documents view:
  1. Move the mouse pointer to the document and press mouse button 2 to display the object pop-up menu.
  2. In the object menu, click **Print**.

     The Print window opens.
  3. Specify the number of copies, the print range, and the printer, and click **Print**.

Chapter 9. Customizing Documents

Document profiles define the content, structure, and format of documents created within DataAtlas Modeler. You can create a document at any time using the DEFAULTS document profile delivered with DataAtlas Modeler. You can also create your own document profiles to produce customized documents.

This chapter describes how to design the document profile elements to customize the content, structure, or format of a document.

Creating a Document Profile

When you create a document profile, you can either copy an existing profile or create a new profile.

To create a new document profile:

1. Open the document profiles view of DataAtlas Modeler.
2. Do one of the following:
   - Move the mouse pointer to an empty area in the document profiles view. Press mouse button 2 to display the pop-up menu for the view. In the pop-up menu, click **Create**.
   - Move the mouse pointer to an existing document profile. Press mouse button 2 to display the pop-up menu for the object. In the object pop-up menu, click **Create another** or **Copy**.

   The Create Document Profile window opens.
3. Specify a profile name and click **OK** to create the new document profile.

You can also create a new document profile when you create a document. To do this, from the Create Document window, click **Create new** or **Copy**.

If you created the document profile using any of the **Create** choices, the new profile is a copy of the DEFAULTS document profile delivered with DataAtlas Modeler.

Defining Title Pages

A title page is divided into three text areas: top, center, and bottom. The text in any of these areas can be several lines long; you can justify the text separately for each area. All title pages have a default font. When you create a document, you can change the font for the title page from the document preview.

To define a title page:

1. Open the tree view of the document profile.
2. Do one of the following:
   - Double-click on the **Title** node.
   - Click the **Title** node. In the **Selected** menu, click **Create element**.
   - Select an existing title, and in the **Selected** menu, click **Copy**.

   If you double-clicked on the node or clicked **Create element**, the Create Title window opens. If you clicked **Copy**, the Settings window for the title opens.
3. Specify a name for the title page.
4. Enter the text for the title page and specify how the text is to be justified.

   You can use document variables in any of the fields. For a list of these variables, see "Appendix D. Document Variables" on page 133.

   If you want the text in any of the title page areas to appear on several lines, use the new-line variable *$(nl)* to indicate where you want a new line to start.
5. Click **OK** to create the title page.

Creating an "Empty" Title Page

If you do not need a title page in your document, you must still select a title page in the Create Document window. In this case, the title page element is an "empty" element. The Settings window for an empty title page might look like Figure 28 on page 101:

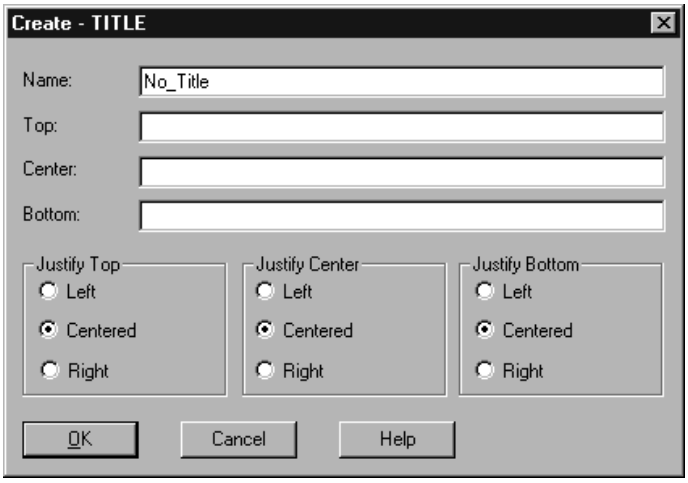


Figure 28. Settings for an "Empty" Title Page

A Title Page Example

To create a title page that looks like Figure 29 on page 102:

Figure 29. Example Title Page

Enter the following in the Create Title window:

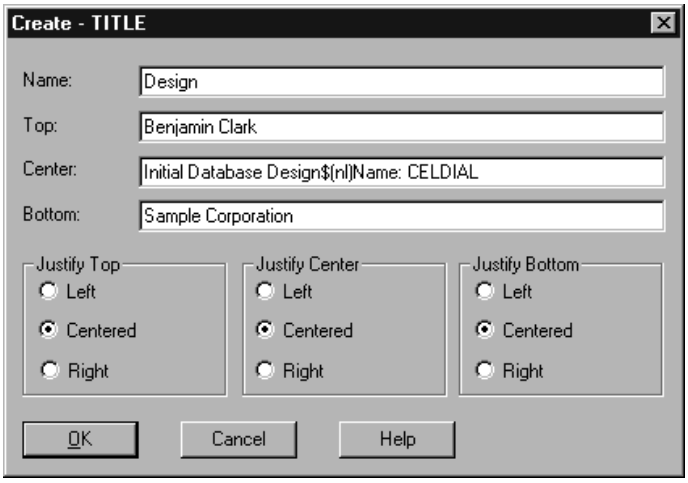


Figure 30. Example of Create Title Window

Defining Headers and Footers

Headers and footers are divided into three text areas: left, center, and right. The text in any of these areas can be several lines long. Headers and footers have a default font. When you create a document, you can change the font used in the document from the document preview.

To define a header or a footer:

1. Open the tree view of the document profile.
2. Do one of the following:
   - Double-click on the document element node, for example, **Header**.
   - Click the document element node. In the **Selected** menu, click **Create element**.
   - Click an existing document element of the type you want to define. In the **Selected** menu, click **Copy**.

   If you double-clicked on the node or clicked **Create element**, the Create window for the selected document element type opens. If you clicked **Copy**, the Settings window for the document element opens. Figure 31 on page 104 shows a Create Header window.

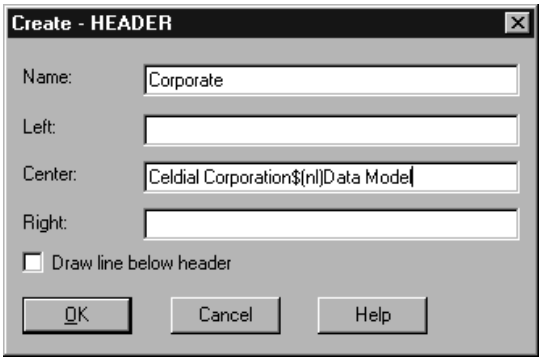


Figure 31. Create Header Window

3. Fill in the window.

   You can use document variables in any of these fields. For a list of the variables, see "Appendix D. Document Variables" on page 133.

4. Click **OK**.

Creating an "Empty" Header or Footer

If you do not need headers or footers in your document, you must still select the appropriate document element in the Create Document window. In this case, the document element is an "empty" element. The settings for an empty footer might look like Figure 32:

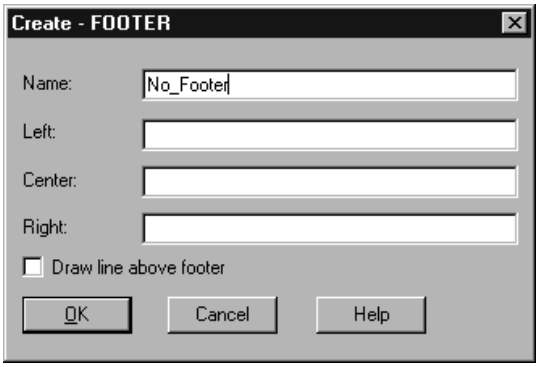


Figure 32. Settings for an "Empty" Footer

A Header and Footer Example

For example, you want the running headers and footers in your document to look like those in Figure 33:

Figure 33. Example of Running Headers and Footers

To do this in DataAtlas Modeler, the Create Header window looks like Figure 34 on page 106, and the Create Footer window looks like Figure 35 on page 106 . Use document variables in the footer for the current date, $(date), and the page numbering, $(page).

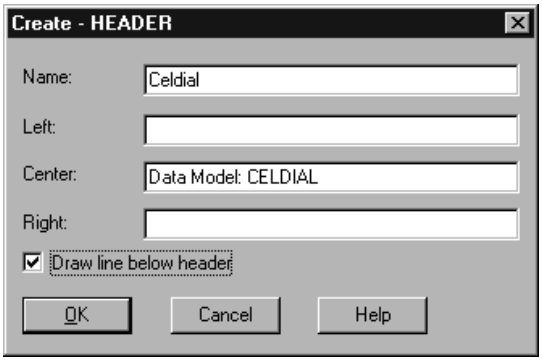


Figure 34. Example of Create Header Window

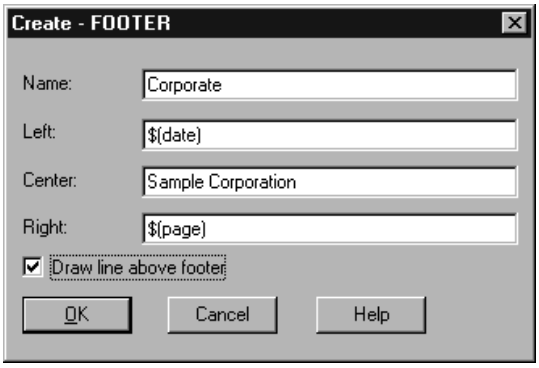


Figure 35. Example of Create Footer Window

Creating a Text Block

You can create additional text for documents that is independent of the data model and its objects. These text blocks can be specific to a particular document, or common to all documents created using the document profile. To include the text in the document you must include it in a document structure. For information on how to include text blocks in a document structure, see "Including a Text Block in a Document Structure" on page 113.

To create a text block:

1. Open the tree view of the document profile.
2. Do one of the following:

- Double-click on the **Text blocks** node.
- Click the **Text blocks** node. In the **Selected** menu, click **Create element**.
- Click an existing text block. In the **Selected** menu, click **Copy**.

If you double-clicked on the node or clicked **Create element**, the Create Text Block window opens. If you clicked **Copy**, the Settings window for the text block opens.

3. Specify the name of the text block.
4. To create the text for the text block, click **Open**.

   DataAtlas Modeler opens the text view of this text block.
5. Enter the text.

   Use the following formatting controls to specify how the text is to be formatted in the document. Use capital letters for the formatting controls and place them at the beginning of a line. Do not include any other text in the same line as the control.

   .B Justifies text to the right and left.

   .C Centers text.

   .H The text following this control does not appear in a document.

   .L Justifies text to the left.

   .P Begins a new page.

   .R Justifies text to the right.

   .S The text following this control is unformatted; long lines are split. The text appears in a document in a proportional font.

   .U The text following this control is unformatted; long lines are split. The text appears in a document in a monospace font.

   For all of these controls except .H, .P, and .U, DataAtlas Modeler runs the text together removing any additional blanks or new-line characters.

Tip

When you include a text block with formatting controls in a document structure, change the settings for the section containing the text block to **Unformatted and proportional**. For information on how to change the settings for a text block, see "Specifying Settings for a Chapter or Section" on page 109.

6. You can also include text that you have prepared in another text editor and saved as an ASCII file. To do this, do one of the following:
   - Include the following formatting control:

     ```
     .include path\filename.ext
     ```

     For example:

     ```
     .include C:\OS2\README.TXT
     ```

     When you create a document that includes the object, the text in the file appears in the document.
   - In the **Edit** menu, click **Get**.

     The text in the file is added to the text view.
7. Close the text view when you have finished.

Finding Where a Text Block Is Used

You can include the same text block in several document structures. It can be useful to find out where a text block is used, for example, when you want to change or delete a text block and you want to check whether it is used anywhere else.

To find where a text block is used:

1. Open the tree view of a document profile.
2. Click a text block.

   The Information Area shows how many times the text block is used.
3. In the **Selected** menu, click **Where used**.

   DataAtlas Modeler displays a list of the document structures that use the selected text block.

Defining a Document Structure

The document structure controls the content of a document. It defines the parts of a data model and the text blocks that are included in a document and the order in which they are appear.

Note: A document contains an appendix only when you specify in the settings for a document section that its graphics are to appear in an appendix.

To define a document structure:

1. Open the tree view of the document profile.
2. Do one of the following:
   - Double-click on the **Structure** node.
   - Click the **Structure** node. In the **Selected** menu, click **Create element**.
   - Click an existing document structure. In the **Selected** menu, click **Copy**.

   If you double-clicked on the node or clicked **Create element**, the Create Document Structure window opens. If you clicked **Copy**, the Settings window for the document structure opens.
3. Specify the name of the document structure.
4. To define the structure, click **Open** to open the document structure tree.

   DataAtlas Modeler displays the tree view of the structure, from which you can:
   - Specify settings for a chapter or section.
   - Change the order of sections in a document section.
   - Exclude and include document sections.
   - Include text blocks.
   - List text blocks used in the structure.
   - Delete a text block.
   - Display the document outline.

   When you have finished, close the tree view.

Specifying Settings for a Chapter or Section

You can define the following settings for the chapters and sections in a document structure:

- The text of a heading
- Whether the heading is to appear for a particular section

- Whether the heading is to be numbered and what sort of numbering is to be used
- Page breaks
- For text chapters and sections, the text format
- For chapters and sections containing graphics, the graphic position and the scaling factor for the graphics

To define document settings:

1. Do one of the following:
   - To define settings for all the elements in the document, for example, page breaks, select all the chapters and all the sections in the structure tree.
   - To define settings for a chapter or a section, click the chapter or the section.
2. In the **Selected** menu, click **Open as settings**.

   The Settings window section opens. Figure 36 shows the Document Section Settings window.

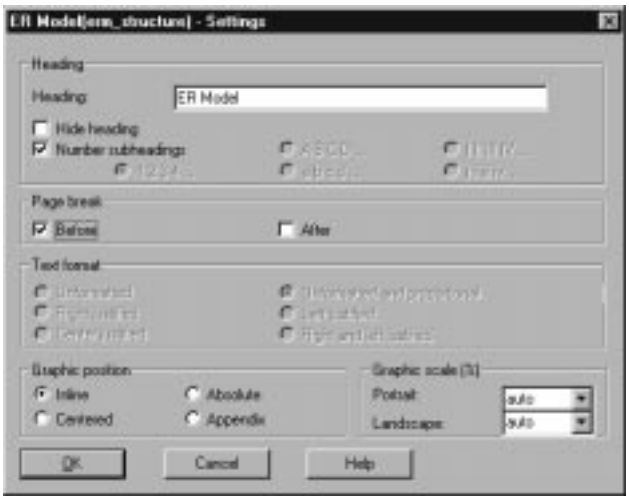


Figure 36. Document Section Settings Window

   If the selected section contains graphics, the fields in **Graphic position** and **Graphic scale** are available. If the selected section is added text or text belonging to an object, the fields in **Text format** are available.

3. Make your changes to the settings.

Tip

If a text block or a description of an object contains formatting controls, click **Unformatted and proportional** as the text format.

4. Click **OK**.

Changing the Order of Chapters and Sections

The document structure in the document structure tree is the default document structure. You can change this structure by dragging a node and dropping it in a different place in the structure tree. You can move nodes only within the same hierarchical level in the tree.

Tip

If you have specified sort order settings for entity groups, entities, relationships, isa bundles, or data elements in the tree view of the data model, these settings affect the order of the objects in the corresponding chapters in a document. For further information on the sort order settings, see "Sorting the Order of Objects" on page 34.

Excluding and Including Chapters and Sections

The default document structure contains all the possible sections a document can have. You might not need all these sections in your document.

To exclude sections from a document structure:

1. Select the sections that you want to exclude.
2. In the **View** menu, click **Exclude**.

   The excluded sections are highlighted in the document structure tree, and the name of the section appears in square brackets. If the section has subsections, DataAtlas Modeler collapses the section to hide the subsections.

To include sections in a document structure that you have previously excluded:

1. Click the section.
2. In the **View** menu, click **Include**.

The sections are no longer highlighted. If the section has subsections, the section is expanded to show them.

For example, if you want to include information only about entity groups and their related objects in a document, exclude all chapters except the Entities chapter as shown in Figure 37:

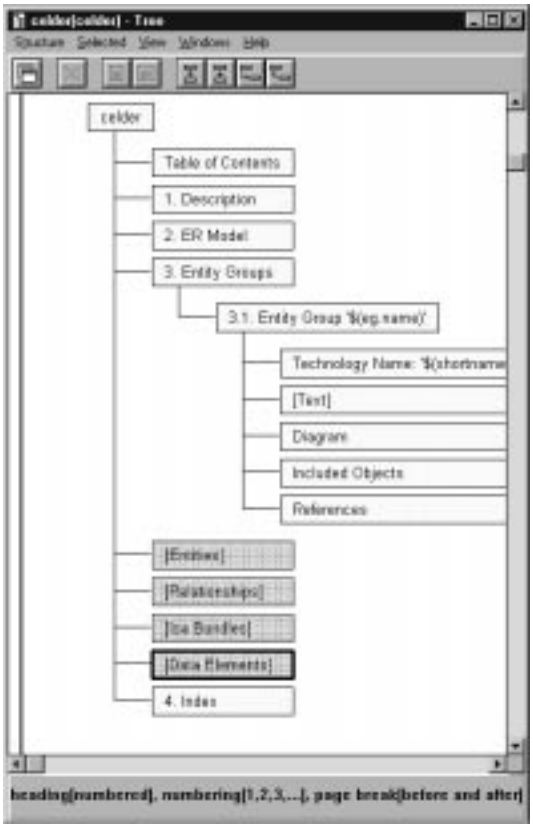


Figure 37. Example of a Document Structure Tree

When you create a document using this structure, the table of contents for the document would look like Figure 38 on page 113:

Figure 38. Example Table of Contents

Including a Text Block in a Document Structure

You can include text blocks defined in a document profile tree in any structure in the document profile. You cannot add text blocks to chapters generated from the contents of the data model; you can only add them as new chapters or to existing text chapters.

Tip

If the added text block contains text formatting controls, update the settings for the new document section. Change the text format to **Unformatted and proportional**.

To include a text block as a new chapter in a document structure:

1. Open the tree view of the document structure.
2. Click an existing chapter, for example, **Entity Groups**.

3. In the **Selected** menu, click **Insert text block**.

   The Insert Text Block window opens. Figure 39 shows the Insert Text Block window.

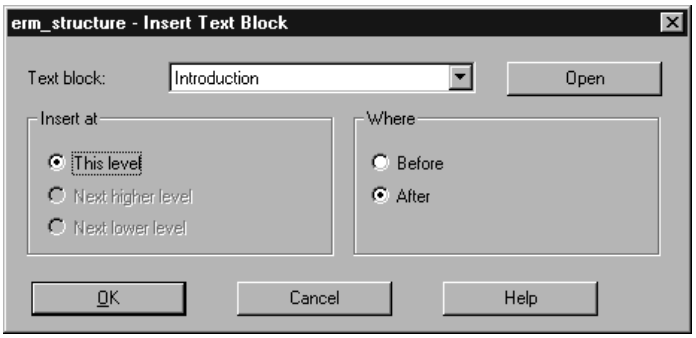


Figure 39. Insert Text Block Window

4. Click a text block in the list, and specify whether the new chapter is to come before or after the selected chapter.
5. If you want to view or change the text block, click **Open**.

   Note: Because text blocks are shared between document structures, if you change the text block here, it is also changed for all document structures that include this text block.
6. Click **OK** to add the text block as a new chapter to the document structure.

To add a text block to an existing text chapter in the document structure:

1. Click an existing text block.
2. In the **Selected** menu, click **Insert text block**.

   The Insert Text Block window opens.
3. Click a text block in the list.
4. Do one of the following:
   - To add the text block on the same structure level as the selected text block, click **This level**, then specify whether the added element is to come before or after the selected element.
   - To add the text block on the structure level above that of the selected element (supersection), click **Next higher level**.
   - To add the text block on the structure level below that of the selected element (subsection), click **Next lower level**.
5. Click **OK** to add the text block to the document structure.

Listing Text Blocks Used in a Document Structure

To list the text blocks used in a document structure:

1. In the **Structure** menu, click **List text blocks**.

   A list of the text blocks used in the structure appears. It shows the name of the text block as defined in the document profile and the name used in the document structure.

Deleting a Text Block from a Document Structure

To delete a text block from a document structure:

1. Click a text block.
2. In the **Selected** menu, click **Delete**.

   The text block is removed from the structure tree. If the text block has dependent text blocks, these move up a level in the document structure.

Displaying the Document Outline

If you have made a lot of changes to the default document structure, it can be difficult to see what your current document contains.

To display the current document outline, click **Outline** in the **Structure** menu. DataAtlas Modeler displays all the chapters and sections that are included in the document structure.

Appendix A. DataAtlas Modeler Environment

This appendix describes the DataAtlas Modeler components and how to configure them for different installation scenarios. It also describes how to start and stop the DataAtlas Modeler control server. How DataAtlas Modeler works together with TeamConnection is also briefly described here. For more detailed information about the TeamConnection functions mentioned here, refer to the *TeamConnection User's Guide*.

DataAtlas Modeler Components

DataAtlas Modeler consists of a server component and a client component. The server component is further split into a control server and a data model server:

- DataAtlas Modeler Control Server

  There must be a DataAtlas Modeler control server running for each TeamConnection family. Each control server is represented by a program icon on the DataAtlas Modeler server machine.

  When you start DataAtlas Modeler on a client, the client connects to the control server for the family and requests a list of data models for the current TeamConnection version context (family, release, and work area). When you open a data model for this family, the control server automatically checks out the data model and starts a DataAtlas Modeler data model server.

- DataAtlas Modeler Data Model Server

  When the data model server is started, it checks whether the previous unload action on the data model was successful or whether a data model recovery is required. "Data Model Recovery" on page 125 describes how DataAtlas Modeler recovers a data model. DataAtlas Modeler then loads the data model into its online cache. If several users are working on the same model, the checking out of the model, and the starting of the data model server are done only for the first user. Subsequent users are connected to the same data model server, and therefore the data model opens faster than for the first user.

  When a data model is unloaded, all objects are unlocked, the data model server is shutdown, and the data model is checked back into TeamConnection.

Configuring the DataAtlas Modeler Environment

You can install the DataAtlas Modeler components in one of the following ways:

- In a single machine installation with the TeamConnection family server and the TeamConnection client components
- In a client/server configuration with the DataAtlas Modeler control server running on the same machine as the TeamConnection family server
- In a distributed client/server configuration with the DataAtlas Modeler components on different machines to the TeamConnection family server

For each of these configurations, you must include an entry in the TCP/IP Hosts file for each machine in the configuration. If you have a name server, you can include the Hosts file entries in the Hosts file on the name server machine instead of defining them locally on each machine in the configuration.

You must also include entries in the TCP/IP Services file for the DataAtlas Modeler control server and the TeamConnection family server. If a DataAtlas Modeler client needs access to more than one TeamConnection family, you must include an entry in the Services file on the client machine for each TeamConnection family server and its DataAtlas Modeler control server.

When a DataAtlas Modeler data model server is started, it automatically searches for the next available free port after the port used by the control server. It is therefore recommended that you reserve a number of ports in the TCP/IP Services file for use by the data model servers. The number of reserved ports depends on the number of models you intend to have open at the same time.

For example, if the port number for the control server *da_testfam* is 5010 and you plan to have 10 data models for the family *testfam* open at the same time, you should reserve the port numbers 5011 through 5020 for the data model servers. The TCP/IP Services file might have the following entries:

```
# Define port for the family server
testfam              5000/tcp
# Define port for DataAtlas Modeler control server
da_testfam          5010/tcp
# Port numbers 5011 - 5020 reserved for DataAtlas Modeler Data Servers
```

If there are not enough reserved port numbers, it might happen that a program requires a port already bound by a data model server and the program cannot be started.

Tip

Some versions of TCP/IP require that the Hosts file ends with a blank line. If you are having problems with TCP/IP settings, ensure that your Hosts file ends with a blank line.

Single Machine Installation

Figure 40 shows the components that must be installed for DataAtlas Modeler to run on a single machine:

DataAtlas Modeler Client
DataAtlas Modeler Server
TeamConnection Client
TeamConnection Server
TCP/IP
OS/2 or Windows NT

Figure 40. DataAtlas Modeler Single Machine Installation

Table 11 shows how you might specify the TCP/IP settings for this installation:

Table 11. TCP/IP Settings for a Single Machine Installation

TCP/IP Settings	Single Machine
Address	9.0.1.1
Name	mymachine
Hosts file	9.0.1.1 mymachine tcserver daserver
Services file	tcserver 5000/tcp daserver 6000/tcp

DataAtlas Modeler Server and Client on Different Machines

In this configuration, the DataAtlas Modeler server and client components are installed on different machines. Figure 41 shows how you might install the components in this configuration:

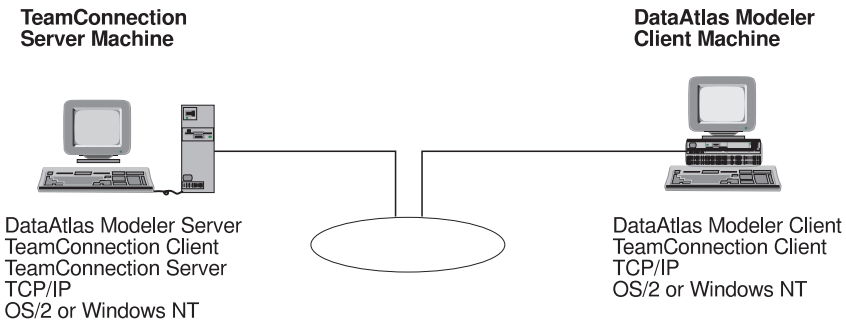


Figure 41. DataAtlas Modeler Client/Server Installation

Table 12 shows how you might specify the TCP/IP settings for this installation:

Table 12. TCP/IP Settings for a Client/Server Installation

TCP/IP Settings	TeamConnection Server Machine	DataAtlas Modeler Client Machine
Address	9.0.2.1	9.0.2.2
Name	tcmachine	mymachine
Hosts file	9.0.2.1 tcmachine tcserver daserver 9.0.2.2 mymachine	9.0.2.1 tcmachine tcserver daserver 9.0.2.2 mymachine
Services file	tcserver 5000/tcp daserver 6000/tcp	tcserver 5000/tcp daserver 6000/tcp

TeamConnection Server, DataAtlas Modeler Server, and Client on Different Machines

In this configuration, the TeamConnection server, the DataAtlas Modeler server, and the DataAtlas Modeler client are all installed on different machines; the TeamConnection server can run on any supported TeamConnection platform. Figure 42 on page 121 shows how you might install the components in this configuration:

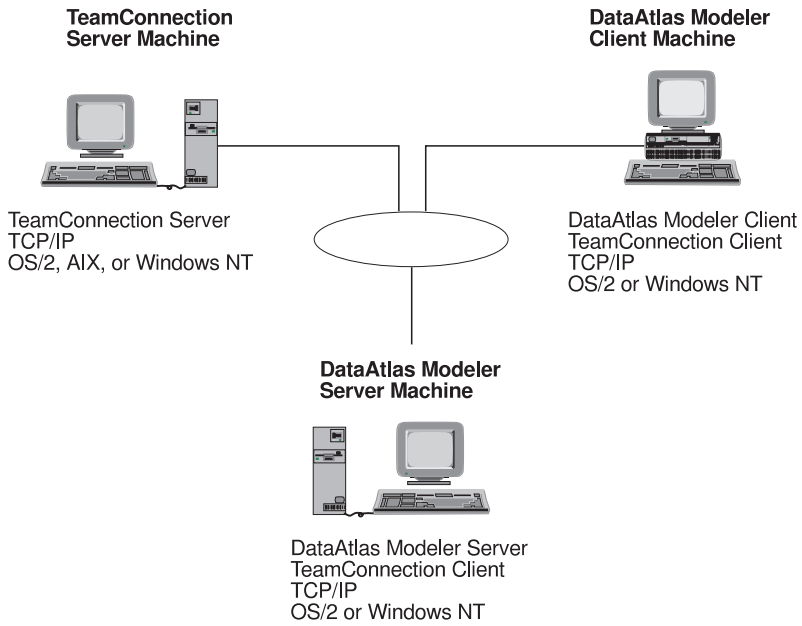


Figure 42. DataAtlas Modeler Distributed Client/Server Installation

Table 13 shows how you might specify the TCP/IP settings for this installation:

Table 13. TCP/IP Settings for a Distributed Client/Server Installation

TCP/IP Settings	**TeamConnection Server Machine**	**DataAtlas Modeler Server Machine**	**DataAtlas Modeler Client Machine**
Address	9.0.3.1	9.0.3.2	9.0.3.3
Name	tcmachine	damachine	mymachine
Hosts file	9.0.3.1 tcmachine tcserver 9.0.3.2 damachine 9.0.3.3 mymachine	9.0.3.1 tcmachine tcserver 9.0.3.2 damachine daserver	9.0.3.1 tcmachine tcserver 9.0.3.2 damachine daserver 9.0.3.3 mymachine
Services file	tcserver 5000/tcp	tcserver 5000/tcp daserver 6000/tcp	tcserver 5000/tcp daserver 6000/tcp

Starting a DataAtlas Modeler Control Server

To start the DataAtlas Modeler control server, your *TC_BECOME* name must be registered as a super user in TeamConnection. If this variable is not set in:

- OS/2, the *TC_USER* variable is used.
- Windows NT, the Windows NT logon user ID is used.

You can start a control server in the following ways:

- From the command line
- From an icon

From the Command Line

To start the control server from the command line, type the following at the command prompt:

```
CDMCSRV  [/F family] [/H hostname] [/N]
```

You can specify the options in any order. The options are case-sensitive.

Options:

/F family
The name of the TeamConnection family you are using. If you do not specify a family name, the environment variable *TC_FAMILY* is used.

/H hostname
The name of the DataAtlas Modeler control server as specified in the TCP/IP Services file. If you do not specify a host name, the environment variable *DA_HOST* is used.

/N The **Unload and discard changes** selection in the Unload Data Model window and in the pop-up menu in the data model view are not available when the last user working on a data model closes it.

From an Icon

When you install the DataAtlas Modeler server component, an icon for a control server is created in the DataAtlas main folder on the desktop (for OS/2) or in the **Programs** entry in the **Startup** menu (for Windows NT).

The default settings for the control server do not include information about the family or the host name. When you start the control server by double-clicking on the control server icon, the control server uses the family and host name specified by the environment variables *TC_FAMILY* and *DA_HOST*.

If you have several families on your server that you want to start from icons, create a DataAtlas Modeler control server icon for each family. Then:

- In OS/2, open the Settings view for each icon, go to the **Program** page in the notebook, and specify the family and the host name of the control server in the **Parameters** field. For example:

  ```
  /H DA_dahost1 /F celdial /N
  ```

- In Windows NT, use the Windows Explorer to find the file CDMCSERV.EXE. Copy it once for each additional control server. Open the Properties notebook for each control server, go to the **Shortcut** page, and add the appropriate parameters after the name of the executable file in the **Target** field. For example:

  ```
  F:\DATATLAS\BIN\CDMCSRV.EXE /H DA_dahost1 /F celdial
  ```

To start a control server from an icon, double-click on the icon.

Shutting Down a DataAtlas Modeler Control Server

You can shut down a control server in the following ways:

- From the OS/2 Window List
- From the Windows NT task bar
- From the command line
- From a minimized window icon

From the OS/2 Window List

To stop or shut down a control server from the OS/2 Window List:

1. Ensure that there are no data models currently loaded. If any are loaded, unload them as described in “Ending a DataAtlas Modeler Session” on page 29 . If you shut down the server without unloading the data models, DataAtlas Modeler might start a data model recovery the next time you open a data model.
2. Shut down the DataAtlas Modeler control server by closing it from the OS/2 Window List.

From the Windows NT Task Bar

To stop or shut down a control server from the Windows NT task bar:

1. Ensure that there are no data models currently loaded. If any are loaded, unload them as described in “Ending a DataAtlas Modeler Session” on page 29 . If you shut down the server without unloading the data models, DataAtlas Modeler might start a data model recovery the next time you open a data model.

2. Shut down the DataAtlas Modeler control server by clicking mouse button 2 on the server entry on the task bar, and then **Close** from the pop-up menu.

From the Command Line

Start another control server from the command line and use this server to shut down the control server for a particular family. Type the following at the command prompt:

```
CDMCSRV [/s normal|forced|save|keep] [/F family] [/H hostname]
```

The options are case-sensitive.

Options:

/s The shutdown behavior of the DataAtlas Modeler control server for the family. Specify one of the following parameters:

normal
Shuts down the control server only when there are no clients connected.

forced Shuts down the control server even when there are clients connected. If there are clients attached, you must manually shut down any running data model servers.

save Shuts down the control server and all the data model servers for this family. Any changes made to the data models are checked into TeamConnection.

keep Shuts down the control server and all the data model servers for this family. The data model is stored on the control server.

/F family
The family for which you want to shut down the control server. Specify this option only if you have several families running on the same server.

/H hostname
The control server you want to shut down. Specify this option only if the control server you want to shut down is different to the environment variable *DA_HOST*.

From a Minimized Window Icon

To shut down a control server from a minimized window icon:

1. Double-click on the minimized window icon.
2. Click **Shutdown**.

This shuts down the control server only when there are no clients attached.

Data Model Recovery

The DataAtlas Modeler server concept allows you to keep models loaded in DataAtlas Modeler for long periods. DataAtlas Modeler keeps log files for each loaded data model. If a model is loaded and a problem occurs with the server so that the loaded models cannot be closed correctly, the log files enable DataAtlas Modeler to recover the models.

When you open a data model after an incorrect shutdown, the data model server recognizes the problems during start and initiates a model recovery. When the recovery is complete, the data model is loaded.

DataAtlas Modeler and TeamConnection

DataAtlas Modeler uses TeamConnection access control and version control to manage data models, data elements, and relational designs.

Access Control

DataAtlas Modeler uses the *TC_BECOME* variable to check the user authorities in TeamConnection. If this variable is not set:

- In OS/2, the *TC_USER* variable is used.
- In Windows NT, the Windows NT logon user ID is used.

DataAtlas Modeler uses the following TeamConnection user authorities to control and manage user access to data models, data elements, and relational designs:

TeamConnection Authority	Allowed Actions
PartAdd	Create objects.
PartCheckOut	Modify objects.
PartExtract	Open a data model, browse its views, and create documentation. You cannot modify the data model or its objects.

DataAtlas Modeler checks the TeamConnection user authorities every time a user attempts to open a data model.

Version Control

You can use TeamConnection version control to manage different versions of your data models. When the DataAtlas Modeler control server checks a data

model out of TeamConnection, it always checks out the most recent version in the TeamConnection database. When a data model that is common to several version contexts is checked back into TeamConnection, the common link is automatically broken. If you modify a data element in DataAtlas Modeler that is common to several version contexts, the common link is also automatically broken when you save your changes.

You can modify a data model in a work area that you can later integrate into a release. For information on changing the TeamConnection version context in a modeling session, see “Changing TeamConnection Family, Release, or Work Area” on page 31.

It is recommended that you set up releases in TeamConnection for serial development. Because DataAtlas Modeler is a multiuser data modeling tool, several users can modify the same data model concurrently from several different client machines. The locking mechanisms in DataAtlas Modeler allow you to synchronize concurrent updates and avoid conflicts. For information on locking in DataAtlas Modeler, see “Locking and Unlocking an Object” on page 42 .

If you use TeamConnection concurrent development and modify the same data model in parallel work areas, when you integrate the work areas into a release you cannot resolve the reconciliation collisions automatically.

Appendix B. How DataAtlas Modeler Propagates Keys

To determine how to propagate keys, DataAtlas Modeler checks each relationship and invokes the first of the following rules that applies:

1. Does the relationship have attributes or constraints of its own?

   If so, the attributes that make up the primary keys of both Entity1 and Entity2 are passed to the relationship, where they form a primary foreign key (PF).

2. Is the maximum cardinality in both directions **many**?
   - (0,X):(0,X)
   - (1,X):(0,X)
   - (0,X):(1,X)
   - (1,X):(1,X)

   If so, the attributes that make up the primary keys of both Entity1 and Entity2 are passed to the relationship, where they form a primary foreign key (PF).

3. Is the cardinality in one direction **0,1** and in the other direction **0,many** or **1,many**?
   - (0,1):(0,X)
   - (0,1):(1,X)
   - (0,X):(0,1)
   - (1,X):(0,1)

   If so, the attributes that make up the primary key of the entity with the maximum cardinality **one** are passed to the other entity, where they form an optional foreign key (oF).

4. Is the cardinality in one direction **1,1** and in the other direction **0,many** or **1,many**?
   - (1,1):(0,X)
   - (1,1):(1,X)
   - (0,X):(1,1)
   - (1,X):(1,1)

   If so, the attributes that make up the primary key of the entity with the maximum cardinality **one** are passed to the other entity, where they form a mandatory foreign key (mF).

5. Is the minimum cardinality in one direction **0**, and in the other direction **1**?

- (0,1):(1,1)
- (1,1):(0,1)

If so, the attributes that make up the primary key of the entity with the minimum cardinality **one** are passed to the other entity, where they form a mandatory foreign key (mF).

6. Is the minimum cardinality in both directions **0**?
   - (0,1):(0,1)

   If so, the attributes that make up the primary key of Entity1 are passed to Entity2, where they form an optional foreign key (oF).
7. Is the minimum cardinality in both directions **1**?
   - (1,1):(1,1)

   If so, the attributes that make up the primary key of Entity1 are passed to Entity2, where they form a mandatory foreign key (mF).

Appendix C. DataAtlas Modeler Tool Bar Choices

The DataAtlas Modeler tool bars contain the following choices:


View data models


View document profiles


View documents


Refresh the container


Open the selected object


Create a new object


Copy the selected object


Delete the selected object


Sort the container ascending


Sort the container descending


Change user profile


Change view and TeamConnection settings


Help for the window


Refresh now


Lock the object corresponding to the current view


Unlock the object corresponding to the current view


Create entity


Create relationship

 Specialize

 Delete

 Group

 Lock selected objects

 Unlock selected objects

 Open as settings

 Open as diagram

 Open as attributes

 Open as text

 Search

 Object

 Select next sibling

 Select previous sibling

 Select first child

 Select parent

 Find and replace

 Zoom in

 Zoom out

 Zoom to fit in window

 Undo zoom

 Align horizontally


Align vertically


Create attribute


Create constraint


Primary


Mandatory


Optional


After each EAM


End of transaction


First page in document


First page in current section


Previous page


Next page


First page in next section


Last page in document


Print document


Display border

Appendix D. Document Variables

You can use the following variables when specifying the contents of headers, footers, and title pages:

$(date)	The current date.
$(family)	The TeamConnection family the data model belongs to.
$(model)	The model name.
$(nl)	New line.
$(page)	The running page number. You cannot use this for a title page.
$(release)	The TeamConnection release the data model belongs to.
$(time)	The current system time.
$(user)	The person who created the document. This is taken from the value of the environment variable *USER* set in the CONFIG.SYS file.
$(workarea)	The TeamConnection work area the data model belongs to.

The following variables are displayed in the document structure tree. They are used by DataAtlas Modeler to substitute object names in document headings. Do **not** use these variables in any other document elements.

$(eg.name)	Entity group
$(e.name)	Entity
$(ea.name)	Entity attribute
$(ec.name)	Entity constraint
$(r.name)	Relationship
$(r.type)	Relationship type
$(ra.name)	Relationship attribute
$(rc.name)	Relationship constraint
$(isab.name)	Isa bundle
$(de.name)	Data element
$(shortname)	Technology name

Appendix E. Transformation Examples

This appendix provides examples of how DataAtlas Modeler transforms objects.

Examples of Transformation of Data Model Objects

Table 14 shows how DataAtlas Modeler transforms entities and their relationships and attributes. The relational design objects are illustrated using SQL CREATE TABLE statements similar to those generated by DataAtlas Dictionary.

Table 14. Transformation of Data Model Objects

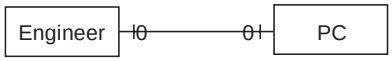
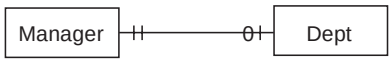
Data Model	SQL Statements
One-to-one relationship where the dependent entity has an optional relationship to the parent entity. For example, a PC is allocated to an engineer or to no one, but not all engineers have a PC. 	CREATE TABLE Engineer (EmpNr INT NOT NULL, Name CHAR(40) NOT NULL, PRIMARY KEY(EmpNr)) CREATE TABLE PC (PCNr INT NOT NULL, EmpNr INT, PRIMARY KEY(PCNr), FOREIGN KEY(EmpNr) REFERENCES Engineer ON DELETE SET NULL)
One-to-one relationship where the dependent entity has a mandatory relationship to the parent entity. For example, every department must have a manager, but a manager can be the manager of at most one department. 	CREATE TABLE Manager (ManNr INT NOT NULL, PRIMARY KEY(ManNr)) CREATE TABLE Dept (DeptNr INT NOT NULL, ManNr INT NOT NULL, PRIMARY KEY(DeptNp), UNIQUE(ManNr), FOREIGN KEY(ManNr) REFERENCES Manager ON DELETE RESTRICT)

Table 14. Transformation of Data Model Objects (continued)

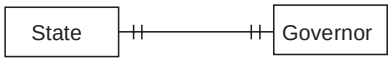
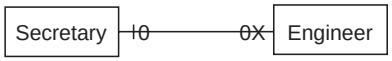
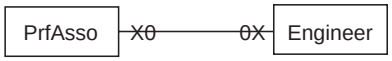
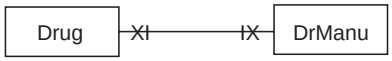
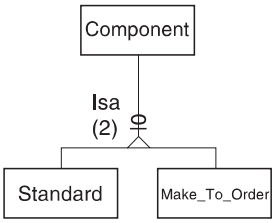
Data Model	SQL Statements
One-to-one relationship with both entities mandatory. For example, a state has exactly one governor, but a governor governs only one state. 	CREATE TABLE State (SName CHAR(40) NOT NULL, PRIMARY KEY(SName)) CREATE TABLE Governor (GName CHAR(40) NOT NULL, SName CHAR(40) NOT NULL, PRIMARY KEY(GName), UNIQUE(SName), FOREIGN KEY(SName) REFERENCES State ON DELETE RESTRICT)
One-to-many relationship where the dependent entity has an optional relationship to the parent entity. For example, an engineer can have at most one secretary, but a secretary can work for several engineers. 	CREATE TABLE Secretary (SecEmpNr INT NOT NULL, PRIMARY KEY(SecEmpNr)) CREATE TABLE Engineer (EmpNr INT NOT NULL, SecEmpNr INT, PRIMARY KEY(EmpNr), FOREIGN KEY(SecEmpNr) REFERENCES Secretary ON DELETE SET NULL)
One-to-many relationship where the dependent entity has a mandatory relationship to the parent entity. For example, every engineer has a secretary, but a secretary can work for several engineers. 	CREATE TABLE Secretary (SecEmpNr INT NOT NULL, PRIMARY KEY(SecEmpNr)) CREATE TABLE Engineer (EmpNr INT NOT NULL, SecEmpNr INT NOT NULL, PRIMARY KEY(EmpNr), FOREIGN KEY(SecEmpNr) REFERENCES Secretary ON DELETE RESTRICT)
Many-to-many relationship. For example, a professional association can have members who are engineers. An engineer can belong to many professional associations or none. 	CREATE TABLE PrfAsso (PaNr INT NOT NULL, PRIMARY KEY(PaNr)) CREATE TABLE Engineer (EmpNr INT NOT NULL, PRIMARY KEY(EmpNr)) CREATE TABLE PrfAsso_Engineer (EmpNr INT NOT NULL, PaNr INT NOT NULL, PRIMARY KEY(EmpNr,PaNr), FOREIGN KEY IsMemb_1 (EmpNr) REFERENCES Engineer ON DELETE RESTRICT, FOREIGN KEY IsMemb_2 (PaNr) REFERENCES PrfAsso ON DELETE RESTRICT)

Table 14. Transformation of Data Model Objects (continued)

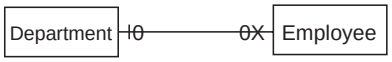
Data Model	SQL Statements
Relationships with attributes. For example, a drug is manufactured under licence by one or more drug manufacturers. A drug manufacturer is licensed to manufacture one or more drugs. The date the licence was issued is an attribute of the relationship. 	CREATE TABLE Drug (DName CHAR(40) NOT NULL, PRIMARY KEY(DName)) CREATE TABLE DrManu (MName CHAR(40) NOT NULL, PRIMARY KEY(MName)) CREATE TABLE Drug_DrManu (DName CHAR(40) NOT NULL, MName CHAR(40) NOT NULL, Date DATE, PRIMARY KEY(DName,MName), FOREIGN KEY lic_1 (DName) REFERENCES Drug ON DELETE RESTRICT, FOREIGN KEY lic_2 (MName) REFERENCES DrManu ON DELETE RESTRICT)
Isa bundles. For example, a component can be either a standard component or a make-to-order component, but not both. 	CREATE TABLE Component (CompID INT NOT NULL, PRIMARY KEY(CompID)) CREATE TABLE Standard (InvNo INT NOT NULL, CompID INT NOT NULL, PRIMARY KEY(InvNo), UNIQUE (CompID), FOREIGN KEY(CompID) REFERENCES Component ON DELETE RESTRICT) CREATE TABLE Make_To_Order (LotNo INT NOT NULL, CompID INT NOT NULL, PRIMARY KEY(LotNo), UNIQUE (CompID), FOREIGN KEY(CompID) REFERENCES Component ON DELETE RESTRICT)

Examples of Transformation of Relational Design Objects

Table 15 contains reverse transformation examples. The relational design objects are illustrated using SQL CREATE TABLE statements similar to those generated by DataAtlas.

Table 15. Transformation of Relational Design Objects

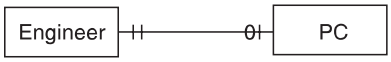
SQL	Data Model

The foreign key is not unique and NULL values are allowed for the foreign key, for example:
`CREATE TABLE Department`
`(DeptId INT NOT NULL,`
`PRIMARY KEY(DeptId))`
`CREATE TABLE Employee`
`(EmpId INT NOT NULL,`
`Name CHAR(40),`
`DeptId INT,`
`PRIMARY KEY(EmpId),`
`FOREIGN KEY(DeptId)`
`REFERENCES Department`
`ON DELETE SET NULL)`
	An optional one-to-many relationship is created.
	 

The foreign key is not unique and the foreign key column is set to NOT NULL, for example:
`CREATE TABLE Department`
`(DeptId INT NOT NULL,`
`PRIMARY KEY(DeptId))`
`CREATE TABLE Employee`
`(EmpId INT NOT NULL,`
`Name CHAR(40),`
`DeptId INT NOT NULL,`
`PRIMARY KEY(EmpId),`
`FOREIGN KEY(DeptId)`
`REFERENCES Department`
`ON DELETE RESTRICT)`
	A one-to-many relationship is created.
	 

Table 15. Transformation of Relational Design Objects (continued)

SQL	Data Model

The foreign key is unique and the foreign key column is set to NOT NULL, for example:
`CREATE TABLE Engineer`
`(EmpNr INT NOT NULL,`
`PCNr INT,`
`PRIMARY KEY(EmpNr))`
`CREATE TABLE PC`
`(PCNr INT NOT NULL,`
`EmpNr INT NOT NULL,`
`PRIMARY KEY(PCNr),`
`UNIQUE(EmpNr),`
`FOREIGN KEY(EmpNr)`
`REFERENCES Engineer`
`ON DELETE RESTRICT)`
	A mandatory one-to-one relationship is created.
	 

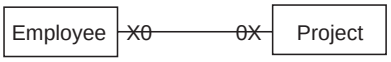
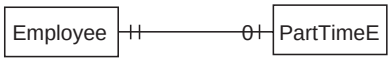
The primary key definition is made up of two foreign key definitions and there are no other foreign key definitions, for example:
`CREATE TABLE Employee`
`(EmpId INT NOT NULL,`
`Name CHAR(40),`
`PRIMARY KEY(EmpId))`
`CREATE TABLE Project`
`(ProjCode INT NOT NULL,`
`Description CHAR(50),`
`PRIMARY KEY(ProjCode))`
`CREATE TABLE ASSIGNMENT`
`(EmpId INT NOT NULL,`
`ProjCode INT NOT NULL,`
`PRIMARY KEY(EmpId,ProjCode),`
`FOREIGN KEY(EmpId)`
`REFERENCES Employee`
`ON DELETE CASCADE,`
`FOREIGN KEY(ProjCode)`
`REFERENCES Project`
`ON DELETE CASCADE)`
	An optional many-to-many relationship is created. The delete rule and the unique attribute value do not affect the transformation.
	 

Table 15. Transformation of Relational Design Objects (continued)

SQL	Data Model

The foreign key definition is the primary key definition, for example:
`CREATE TABLE Employee`
`(EmpID INT NOT NULL,`
`Name CHAR(40),`
`PRIMARY KEY(EmpID))`
`CREATE TABLE PartTimeE`
`(EmpId INT NOT NULL,`
`HourlyRate INT,`
`PRIMARY KEY(EmpId),`
`FOREIGN KEY(EmpId)`
`REFERENCES Employee`
`ON DELETE RESTRICT)`
	A one-to-one relationship is created where the dependent entity is mandatory. The delete rule and the unique attribute value do not affect the transformation.
	 

Glossary

This glossary defines terms and abbreviations that are used in this book. If you do not find the term you are looking for, refer to the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

A

associative entity. A construct used to model a relationship that has local attributes, is a many-to-many relationship, a relationship with a degree greater than 2, or a relationship to a relationship. An associative entity does not usually represent a real-world object.

attribute. A characteristic of an entity. For example, an entity named EMPLOYEE may have an attribute called phone number.

attributive entity. A construct used to model a composite attribute or a multivalued attribute. An attributive entity is a dependent entity; its existence depends on the existence of another entity.

B

binary relationship. A relationship between 2 entities. A binary relationship has degree two.

C

cardinality. The number of instances of an entity that can or must participate in a relationship. See *minimum cardinality*, *maximum cardinality*.

CASE model. The TeamConnection representation of an entity-relationship (ER) model.

collapse. Temporarily reduce the level of detail in a tree view by hiding the dependents of one or more nodes.

component. A TeamConnection object that organizes project data into structured groups, and controls configuration management properties.

conceptual data model. A model that describes the structure of data independently of implementation considerations, such as which type of database management system will be used to implement the database. One type of conceptual data model is an entity-relationship model.

constraint. A DataAtlas Modeler rule that governs the validity of the data manipulation operations, such as insert, delete, and update, associated with an entity or a relationship.

D

data element. A construct that determines the data type of an attribute. Each attribute can have exactly one data element associated with it, but one data element can apply to any number of different attributes. In TeamConnection, data elements are shareable objects.

data model. A model that represents the way data is organized. See *conceptual data model*, *logical data model*, *physical data model*.

data type. A category that identifies the internal representation of the data, for example, *integer*, *numeric*, and *character*.

DBMS. Database management system.

degree of a relationship. The number of different entities that participate in a relationship. DataAtlas Modeler supports unary and binary relationships.

dependent entity. The entity to where the foreign key is passed along a relationship.

document. A file that contains the graphical and textual descriptions of some or all of the objects contained in a data model.

document profile. A collection of elements that define the content, structure, and format of documents created in DataAtlas Modeler. It contains definitions for title pages, headers, footers, additional text, and document structures.

E

entity. In data modeling, a representation of a class of things with common characteristics or properties. The characteristics or properties are called *attributes*. A member of this group is called an *entity instance*. *Person* in an example of an entity; *the person with person ID 553-33-2729* is an example of an entity instance.

entity group. A named group of entities used to consolidate entities that belong together and to organize a data model hierarchically.

entity instance. The single occurrence of an entity.

entity-relationship model (ER model). A data model based on the concept of entities and the relationships among entities, and of the attributes of entities and relationships.

entity type. See *entity*.

ER model. Entity-relationship model.

expand. Show the hidden dependents of a collapsed node in a tree view.

F

family. A logical organization of related development data. A single TeamConnection server can support multiple families. The data in one family cannot be accessed from another family.

foreign key. A primary key that is propagated along a relationship.

G

generalization. A technique for grouping together entities with common attributes and associating them with a supertype.

ghost entity. An entity outside an entity group that participates in a relationship with an entity in the entity group.

I

identifier. See *primary key*.

information area. A part of a window where information about the selected object or action appears.

isa bundle. A group of related isa relationships.

isa relationship. A relationship between a subtype and a supertype.

L

locking. An internal mechanism used to protect objects. Locking ensures that no other user can change the object currently being used.

logical data model. A model that describes the structure of data with regard to the type of database management system that will be used to implement the database. Contrast with *conceptual data model* and *physical data model*.

M

mandatory attribute. An attribute that must have a non-null value. Contrast with *optional attribute*.

many-to-many relationship. A binary relationship between entities A and B where each instance of A may be related to many instances of B and each instance of B may be related to many instances of A. The term *many* refers to the

maximum cardinalities of the relationship in each direction. Contrast with *one-to-many relationship* and *one-to-one relationship*.

maximum cardinality. The highest number of entity instances that can participate in a relationship. A maximum cardinality can be any number, but in DataAtlas Modeler it is either *one* or *many*.

minimum cardinality. The lowest number of entity instances that can participate in a relationship. A minimum cardinality can be any number, but in DataAtlas Modeler it can be either *zero*, indicating that participation in the relationship is optional, or *one*, indicating that participation is mandatory.

O

one-to-many relationship. A binary relationship between entities A and B where each instance of A may be related to many instances of B and each instance of B is related to at most one instance of A. The terms *one* and *many* refer to the maximum cardinalities of the relationship in each direction. Contrast with *many-to-many relationship* and *one-to-one relationship*.

one-to-one relationship. A binary relationship between entities A and B where each instance of A is related to at most one instance of B, and each instance of B is related to at most one instance of A. The term *one* refers to the maximum cardinalities of the relationship in each direction. Contrast with *one-to-many relationship* and *many-to-many relationship*.

optional attribute. An attribute that can but need not have a non-null value. Contrast with *mandatory attribute*.

P

physical data model. A representation of data that exploits the features of a particular DBMS. Typically, the model is tuned to achieve optimal performance for a given data load and work load.

populate. The extraction of data definitions from external sources, such as relational database catalog, and insertion into the TeamConnection database.

primary key. An attribute or group of attributes that uniquely identifies each instance of an entity or relationship.

R

RDBMS. Relational database management system.

relational design. A collection of table definitions plus one or more physical designs.

relationship. An association between two entities (binary) or between an entity and itself (unary).

release. A TeamConnection object defined by a user that contains all the parts that must be built, tested, and distributed as a single entity.

S

subtype. An entity that describes a subset of another, more global entity (its supertype).

supertype. An entity that describes a superset of other, less global entities (its subtypes).

T

table. A named data object consisting of a specific number of columns and some number of unordered rows.

text block. A block of user-defined text that you can include in a document even though it is not part of the data model.

U

unary relationship. A relationship between an entity and itself. A unary relationship has degree one.

V

version. A specific view of a driver, release, or work area.

W

work area. An object in TeamConnection that you create and associate with a release. When the work area is created, you see the most current view of the release and all the parts that it contains. You can check out the parts in the work area, make modifications, and check them back into the work area. You can also test the modifications without integrating them. Other users are not aware of the changes that you make in the work area until you integrate the work area to a release. While you work on parts in a work area, you do not see subsequent part changes in the release until you integrate or refresh your work area.

Index

D

E

F

U

V

W

Z

Printed in the USA

SC26-9041-01