

BC Data Modeler



HELP.BCDWBTOODMO

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

BC Data Modeler	9
BC Data Modeler.....	10
Data Modeler: overview and modeling principles.....	11
Data Modeler: overview	12
Entity types	14
Attributes of entity types	16
Relationships	18
Specializations/generalizations	22
Structural concept: data model and data model hierarchy	24
Top-down and bottom-up modeling	26
Consistency checks	27
Integration of the Data Modeler with the Business Navigator and the Business Object Repository.....	29
Overview of the data model.....	30
Accessing the Data Modeler	31
Access from the menu.....	32
Access from the Object Browser of the ABAP/4 Development Workbench	33
Access from the ABAP/4 Repository Information System	34
Accessing the hierarchy of SAP models	35
ABAP/4 Repository Information System.....	36
Repository Information System: overview	37
Repository Information System: access.....	38
Repository Information System: searching for objects	39
Repository Information System: settings.....	40
Repository Information System: selection options	41
Repository Information System: examples for selection options.....	42
Repository Information System: where-used list	43
Data Modeler: handling.....	44
Version management	45
Workbench Organizer	46
Request management.....	48
User settings.....	49
Tool (pre 3.0).....	50
Printing objects	51
Data Modeler: maintaining entity types	52
Entity type: definition.....	53
Time-dependent (entity type)	54
Customizing object	55
Assigned table/view	56
Entity type: Attributes.....	57
Entity type: attributes	59
Relationship role	61

Relationship: category.....62

Relationship: cardinality.....64

Relationship: temporary reference.....65

Entity type: generalizations.....66

Entity type: specializations/specialization categories67

Specialization category: attributes.....69

Entity type: Dictionary assignment70

Entity type: comment.....71

Entity type: example72

Entity type: aliases.....73

Entity type: Data Browser.....74

Entity type: hypertext.....75

Entity type: variants76

Entity type: switching to a different entity type78

Entity type: use in data models (inverse hierarchy).....79

Inverse hierarchy: changing the display80

Entity type: specializations/specialization categories81

Data Modeler: maintaining data models83

Displaying/printing data models.....84

Creating data models.....85

Data Model: Type.....86

Data model: references.....87

Changing a data model.....88

Deleting a data model89

Switching to a different data model.....90

Data model: making consistency checks91

Consistency check: completeness92

Consistency check: existence of predecessors93

Consistency check: connectivity.....94

Consistency check: existence of hierarchy objects.....95

Data model: use in other data models (inverse hierarchy).....96

Inverse hierarchy: changing the display97

Data Modeler: hierarchy98

Hierarchy: access.....99

Hierarchy: printing100

Hierarchy: changing the display.....101

Changing a hierarchy.....102

Changing a hierarchy: inserting objects103

Changing a hierarchy: deleting assignments104

Changing a hierarchy: reassigning modeling objects105

Hierarchy: switching data models.....106

Hierarchy: making consistency checks.....107

Consistency checks: completeness108

Consistency check: connectivity.....109

Consistency check: existence of predecessors110

Consistency check: existence of hierarchy objects.....	111
Hierarchy: accessing the graphics.....	112
Hierarchy: graphics positions.....	113
Data Modeler: graphics.....	114
Graphics: display method (SAP-SERM).....	115
Graphics: accessing	119
Graphics: display mode/maintain mode	120
Graphics: changing the section displayed.....	122
Graphics: Select	123
Graphics: displaying/changing objects	124
Graphics: creating objects	125
Graphics: deleting objects	126
Graphics: positioning mode.....	127
Graphics: Get environment, Suppress, Reload (Utilities).....	128
Graphics: other functions (Edit)	129
Graphics: changing the settings	130
Graphics: printing	132
Using the Data Modeler	133
Accessing the Data Modeler	134
Repository Information System	135
Entity types	136
Attributes.....	137
Relationships	138
Generalization/specialization	139
Dictionary assignment.....	140
Data models	141
Data model hierarchy.....	142
Graphics	143
Other themes	144
navigate to the Business Navigator	145
navigate in the Business Object Repository	146
access the Data Modeler via the menu	147
access the Data Modeler via the Object Browser	148
access the Data Modeler via the ABAP/4 Repository Information System	150
access the hierarchy of the SAP Model.....	151
call the Repository Information System from the Data Modeler	152
search for objects with the Repository Information System	153
define the user settings in the Repository Information System.....	154
set the selection options in the Repository Information System.....	155
use the Repository Information System to find out where objects are used	156
define the user settings for the Data Modeler.....	157
print objects	158
display the definition of an entity type.....	159
create the definition of an entity type	160
change the definition of an entity type	161

delete the definition of an entity type.....162

branch from the definition of an entity type to the graphics function.....163

assign a table or a view to an entity type164

display the attributes of an entity type165

create the attributes of an entity type166

change the attributes of an entity type167

delete the attributes of an entity type168

create tables and transfer attributes169

display relationships.....170

create relationships171

change relationships172

delete relationships.....173

display generalizations.....174

create generalizations.....175

delete generalizations176

display specializations/specialization categories.....177

create specializations/specialization categories178

change the specialization category.....179

delete specialization category/specializations.....180

display the Dictionary assignment of an entity type181

create or change the Dictionary assignment of an entity type.....182

delete the Dictionary assignment of an entity type184

display, create or change a comment for an entity type.....185

delete a comment for an entity type.....186

display, create or change an example for an entity type187

delete an example for an entity type188

delete, create or change an alias for an entity type.....189

delete an alias for an entity190

display the contents of the table/view which is assigned to an entity type.....191

view the hypertext of an entity type192

create a variant for an entity type193

display or change a variant for an entity type.....194

delete a variant for an entity type195

switch to another entity type.....196

determine the assignment of an entity type to data models197

change the display of the inverse hierarchy198

print entity types and their subobjects).....199

display and print data models.....200

create a data model.....201

change a data model.....202

delete a data model.....203

switch to another data model.....205

carry out consistency checks206

find out in which other data models a data model is used207

change the display of the inverse hierarchy208

access the data model	209
print the data model hierarchy	211
change the display of the data model hierarchy	212
insert objects in the data model hierarchy	213
delete assignments in the data model hierarchy	215
reassign objects in the data model hierarchy	216
switch to the hierarchy of another data model	217
carry out consistency checks from the data model hierarchy	218
switch to the graphic function from the data model hierarchy	219
switch to the graphics positions of modeling objects from the data model hierarchy	220
call the graphics function	221
change the section of the screen displayed in the graphics function	223
select objects in the graphics function	224
display/change objects in the graphics function	225
create objects in the graphics function	227
delete objects in the graphics function	229
change the positions in the graphics	230
use the graphics utilities Get environment, Suppress and Reload	231
use the graphics utilities Compress/Expand data model, Adjust, Visibility and Restrict area	232
change the settings in the graphics function	233
print the graphic	234

BC Data Modeler

BC Data Modeler

Introduction

[Data Modeler: Overview and modeling principles \[Page 11\]](#)

[Accessing the Data Modeler \[Page 31\]](#)

[ABAP/4 Repository Information System \[Page 36\]](#)

Maintaining modeling objects

[Data Modeler: handling \[Page 44\]](#)

[Data Modeler: maintaining entity types \[Page 52\]](#)

[Data Modeler: maintaining data models \[Page 83\]](#)

[Data Modeler: hierarchy \[Page 98\]](#)

[Data Modeler: graphics \[Page 114\]](#)

Notes on handling

[Using the Data Modeler \[Page 133\]](#)

Data Modeler: overview and modeling principles

Data Modeler: overview

This contains a short description of the Data Modeler and its functions.

[Data Modeler: overview \[Page 12\]](#)

Data Modeler: modeling principles

You are given information on the basic objects of the Data Modeler, on the general structural concept, on top-down and bottom-up modeling, and on available consistency checks:

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

[Integration of the Data Modeler with the Business Navigator and the Business Object Repository \[Page 29\]](#)

[Overview of the data model \[Page 30\]](#)

Data Modeler: overview

What is the Data Modeler and how can it be used?

The Data Modeler is a development tool from the ABAP/4 Development Workbench, which you can use to create data models according to the SAP SERM method (SERM = Structured Entity Relationship Model). In addition to supporting modeling functions, the Data Modeler allows you to map the models you have created to the ABAP/4 Dictionary.

Owing to the close integration of the Data Modeler with the ABAP/4 Dictionary, both top-down and bottom-up modeling approaches are possible.

The basic objects of the Data Modeler are data models and entity types, and the relationships and specializations existing between the entity types.

The problem to be modeled - generally a small section of the real or abstract world - is mapped to a data model.

The individual (physical or abstract) objects of the section to be modeled are represented by entity types that are connected by relationships.



A data model of a university is to be created. It could be given the name *Unimodell*, for example. To begin with, we will consider just a small part of the model, the professors and students at the university. We define two entity types *Professors* and *Students* and a relationship between the two entity types, which we will label *Teaches*.

In the Data Modeler you can create data models with any number of hierarchy levels and in the process reuse parts of data models. The data model components can be expanded or compressed in a hierarchy list and in the graphics display, thus facilitating work with large data models.

Modeling involves use of a graphical editor, which allows you to design data models as if you were using a drawing board.

Other subjects:

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

[Integration of the Data Modeler with the Business Navigator and the Business Object](#)

[Repository \[Page 29\]](#)

[Overview of the data model \[Page 30\]](#)

Entity types

Entity types

An **entity** is a physical or abstract object (for example, Mr. Douglas, the "Sales Information System" project) that can be distinguished from other objects and for which information is to be stored. Entities are divided into entity types according to their properties.

Each **entity type** is assigned attributes, which describe the entities of the entity type. The properties of entities are described by means of actual values for the attributes.

An **attribute** consists of a name and a definition of the possible values for the attribute (for example, the attribute *color* with the values *white*, *black*, *yellow*, and so on). One or more attributes are labeled as key attributes. The values of the key attributes uniquely identify an entity within its entity type.



Entities, entity types, attributes

Term	Example
Entities	Mr Douglas Ms Roberts Ms Stone 'Sales Information System' project 'Warehouse Management' project
Entity types	employee project
Attributes	<p>Attributes of entity type 'employee': <u>personnel number</u> (key attribute) name address date joined company</p> <p>Attributes of entity type 'project': <u>project number</u>(key attribute) name start date end date project manager</p>

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

Attributes of entity types

Attributes of entity types

Attributes define the possible properties for entities of an entity type.

Attribute type

The attribute type describes the function of an attribute (for example, whether an attribute is a key attribute).

Defining attributes

Attributes are either assigned directly to an entity type or transferred from the ABAP/4 Dictionary.

Attributes that are assigned directly obtain their semantic and technical characteristics from a data element which is assigned to the attribute. Since the Data Modeler and the ABAP/4 Dictionary use the same data elements, data elements defined in the Data Modeler can be used later in the ABAP/4 Dictionary and vice versa.

Attributes are transferred from the ABAP/4 Dictionary by assigning a table or view to an entity type. The fields of the table or view are transferred to the entity type as attributes and receive their semantic and technical characteristics from the data elements of the fields. Attributes transferred from the table or view are flagged in the Data Modeler. Assigning a view or table to an entity type establishes the connection between the Data Modeler and the ABAP/4 Dictionary. In specializations the attributes inherited from the generalization can also be displayed (see [Specializations/generalizations \[Page 22\]](#)).

Dictionary assignment

Assigning a table:

An entity type is mapped to exactly one table in the ABAP/4 Dictionary. The fields of the table are transferred to the entity type as attributes and are included in the attribute display.

Assigning a view:

The entity type is mapped to one or more tables of the ABAP/4 Dictionary using a view. Here is a list of instances where it is advisable to assign a view:

1. The information, which from a modeling perspective is distributed over a number of different entity types, is contained in a single table:

Consequently, the table contains fields which are not attributes of the entity type. **Projection** on the required fields produces a view containing only those fields corresponding to the attributes of the entity type. The table also contains entries for other entity types. These are suppressed by means of **selection**.



You have modeled the entity types *Employee* and *Company car owner*. The Dictionary contains only one table which has all the attributes. To resolve this problem, a view is assigned to the entity types involved by means of selection and projection.

2. The attributes of an entity type are distributed over several tables:

Attributes of entity types

You can use a view to bring together the fields from the tables involved. The result is a virtual table which shows precisely those fields that correspond to the attributes of the entity type.



You have modeled the entity type *Courses at the university* with the attributes *Faculty*, *Course number*, *Course title*, and *Course description*. In the ABAP/4 Dictionary, the attributes are distributed over the tables UKURS and UKRSB (because courses are held in several languages).

3. View assigned to a specialization:

When a specialization of an entity type is created, the attributes of the generalization are passed on to it. This also applies to a view that is assigned to a specialization. When such a view is created, it inherits all tables, fields, and selection conditions of the view or table assigned to the generalization.

It is now possible to add more fields and tables to the view, since a specialization has more characteristics. The selection condition can also be tightened up so that only the entities belonging to the specialization are selected.

If, at a later time, the view assigned to the generalization is changed, the changes can, if required, be passed on to the views of the specializations. This does not happen automatically, since minor changes made to the model could have extensive repercussions for views in operational use.

If requested, attributes of the generalization are passed on to the specialization during view maintenance. If a view is assigned to an entity type, a check is made to see whether a generalization exists for the entity type. If so, the attributes of the generalization, which are defined by means of ABAP/4 Dictionary assignment (of a table or view) are transferred to the view of the specialization.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

Relationships

Relationships

Relationships between two entity types are defined by specifying the two entity types involved, that is, the source entity type (the existentially independent) and the target entity type (the existentially dependent entity type). The relationship is oriented; from the independent entity type to the dependent entity type. Relationships have these characteristics:

- category
- cardinality
- their business significance



The relationship *Offers* exists between the entity types *Faculty* (source entity type) and *Courses* (target entity type).

Category

A relationship category can be hierarchical, aggregating, referential or external.

Hierarchical:

There is a hierarchical relationship category between two entity types when the following apply:

- The target entity type is existentially dependent upon the source entity type, that is, the life span of a characteristic of the target entity type is shorter than or equal to the life span of the characteristic of the source entity type.
- The target entity type is generated from the source entity type, that is, the source entity type influences the characteristic values directly.
- The target entity type represents a semantic refinement, that is, the target entity type is a classification of the source entity type which describes the source entity type more precisely.

The key of the source entity type becomes part of the key of the target entity type. The relationship between two entities may not be changed.



The relationship *Offers* with the cardinality 1:CN exists between the entity types *Faculty* (source entity type) and *Courses* (target entity type).

The source entity type *Faculty* has the attributes *Faculty number* (key attribute) and *Faculty name*.

The target entity type *Courses* has the attributes *Faculty number* (key attribute), *Course number* (key attribute), *Number of the course instructor* and *Course title*.



Since, with this category of relationship, the source entity identifies the associated target entity by means of a part of the key attributes, a change to a relationship would mean that the target entity would be deleted and a new target entity would be created.

Aggregating:

There is an aggregating relationship category between two entity types when the following apply:

- The target entity type is existentially dependent on the source entity type, that is, the life span of a characteristic of the target entity type is shorter than or equal to the life span of the characteristic of the source entity type.
- The target entity type is generated from the source entity type, that is, the source entity type influences the characteristic values directly.
- At least one other source entity type, different from the first one, is involved in the formation of the target entity type.

The keys of the source entity types become part of the canonic key of the target entity type.



The aggregating and the hierarchical relationship categories differ from one another only in the third characteristic.

Referential:

There is a referential relationship category when the following apply:

- The target entity type is existentially dependent on the source entity type.
- The source entity type determines the context of the target entity type, that is, an attribute group of the start entity type is present in the target entity type, but this does not generate the target entity type.

The key attributes of the source entity type are included in the target entity type as non-key-attributes. A relationship between two entities can be changed.



The relationship *Teaches* with cardinality 1:CN exists between the entity types *Professor* (source entity type) and *Courses* (target entity type).

The source entity type *Professor* has the attributes *Number* (key attribute), *Name*, *Address*, and *Remuneration class*.

The target entity type *Courses* has the attributes *Faculty number* (key attribute), *Course number* (key attribute), *Number of the course instructor* and *Course name*.

A distinction is made between strong and weak existential dependence. Strong existential dependence requires there to be an assignment to exactly one characteristic of the source entity type for each characteristic of the target entity type. If this condition applies only to one (time-dependent) subset of the target entity type, the existential dependence is said to be weak.



Weak existential dependence can occur in aggregating and referential relationship categories but not in hierarchical relationship categories.

This results in the following situation:

Classification of relationship categories

Relationships

	strong existential dependence	weak existential dependence
Generates	hierarchical aggregating	conditional-aggregating
Context	referential	conditional-referential temporary-referential

External:

A relationship is in the 'external' category if it is between an entity type within a data model and an entity type outside the data model.

Cardinality

The cardinality (n : m) describes the relationship with respect to the number of dependent entities (entities of the target entity type of the relationship) and referenced entities (entities of the source entity type) allowed for the relationship.

The possible values for the left and right sides of the cardinality have the following significance:

n = 1

Each dependent entity has one referenced entity.

n = C

Dependent entities can exist that do not refer to an entity of the source entity type.

m = 1

Each entity of the source entity type has one dependent entity.

m = C

Each entity of the source entity type has a maximum of one dependent entity.

m = N

Each entity of the source entity type has at least one dependent entity.

m = CN

Each entity of the source entity type can have any number of dependent entities.



The Structured Entity Relationship Model (SERM) does not allow m:n relationships (i.e. many-to-many relationships). Therefore, entity types in SERM can be arranged from left to right, from independent to dependent entity type.



The cardinality C:x (x = 1, C, CN, N) should be used above all for relationships of the category 'referential'. However, it can also be used for relationships of the category 'aggregating'.

Hierarchical relationships do **not** allow this cardinality, since all dependent entities must refer to an entity of the source entity type.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

Specializations/generalizations

Specializations/generalizations

The concept of specializations and generalizations allows the formation of entity type subtypes. There are a number of ways in which an entity type can be specialized. The criterion used for the specialization is defined by the specialization category. Specializations are linked by the specialization category to the source entity type (= generalization).



Generalization, specialization category, specialization

Term	Example
Generalization	People at the university
Specialization category	Activity
Specializations	Students, professors, administrative staff, research assistants

The generalization contains the attributes that are common to all entities of an entity type. These attributes are inherited by the specializations (see also [Attributes of entity types \[Page 16\]](#)) and may be supplemented by other specific attributes.



In the university example the generalization *People at the university* has the attributes *Number*, *Name*, and *Address of a university member*, which it passes on to its specializations (*Students*, *Professors*, and so on). The specialization *Students* also has the following attributes: *Matriculation number*, *Assigned professor*, and *Course start*.

The specialization category can be characterized by the attributes:

- **complete:**
Each entity of the generalization occurs in at least one specialization of the category.
- **disjoint:**
Each entity of the generalization occurs in a maximum of one specialization of the category.



The specialization category *Activity* is complete in the university example, since each university member belongs to at least one specialization. This specialization category is **not** disjoint, however, since it is conceivable that a person might be at the same time both a student and a research assistant with the result that they would occur in both specializations.



A specialization category need be neither complete nor disjoint.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

Structural concept: data model and data model hierarchy

Structural concept: data model and data model hierarchy

The information is structured using data models and data model hierarchies (nesting of data models).

Data model

A data model is made up of entity types, relationship categories and specialization categories. Semantically related objects are combined to form their own data model. Complex models are split into smaller submodels which can themselves also be structured.

There are some special types of data model:

- Application data models
- Business objects
- Application data models

A model is termed an application data model if it forms an independent unit from a commercial viewpoint. Examples of application data models are SAP's FI and MM models.

- Business objects

A business object is a set of entity types sharing a common external interface. A business object normally consists of a source entity type and all those entity types hierarchically dependent on it. The source entity type normally functions as representative of the business object to which it belongs.



The entity types *Warehouse* and *Storage bin* could be regarded as a business object. In this case, the entity type *Storage bin* is hierarchically dependent on the entity type *Warehouse*, which is the source entity type.

A data model can have the following references:

- **Representative**

The representative of a data model is the entity type that represents the entity types belonging to the data model when the model is compressed.

- **Referenced model**

The referenced model is the data model on which the current data model is based. A data model is normally derived from the referenced data model by means of projection.

A future version of the Data Modeler will allow you to compare the two models and to display the differences between them.

Data model hierarchy

The underlying structure of a data model is termed the data model hierarchy. It takes the form of an oriented acyclic graph. Data models and entity types can be used in several data models.

For example, the entity type *Plant* is used both in the *FI* data model and the *MM* data model.

Structural concept: data model and data model hierarchy

The data model hierarchy provides you with a fast method of obtaining an overview of the structure of a data model and the entity types and submodels participating in it.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

[Consistency checks \[Page 27\]](#)

Top-down and bottom-up modeling

Top-down and bottom-up modeling

The Data Modeler supports both top-down and bottom-up modeling.

Top-down

To begin with, a complete model is created for the area to be modeled. As modeling proceeds, this model is further refined through the addition of entity types and submodels, which can, in turn, be further refined.

Attributes are entered for the various entity types. If the technical and semantic characteristics of the attributes have been fully clarified, data elements (and domains) representing these characteristics can be created or existing ones can be accessed from within the data model.

The link between the Data Modeler and the ABAP/4 Dictionary is established by assigning entity types to tables or views. You can create new tables or views from within the entity type. The attributes of the entity type can be included in the table as fields and the data elements that are created can be used elsewhere.

Once assignment to a table or view has taken place, the fields of the table or view are displayed as attributes of the entity type and are laid over the attributes (of the same name) previously created for the entity type. Changes to the assigned Dictionary objects (that is, creation/deletion/modification of fields) thus become immediately effective and visible in the Data Modeler too. Since both data model and Dictionary access the same information, their consistency is ensured.

Bottom-up

The Data Modeler can also be used to remodel an existing application.

Entity types are created for the existing tables and views to which the tables and views are then assigned. By these means, the entity types obtain their attributes.

Logically connected entity types are combined to form data models. These can, in turn, be combined with other entity types and data models to form generic data models.

You can, of course, use a mixture of these two strategies for modeling and implementation of an application.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Consistency checks \[Page 27\]](#)

Consistency checks

The basic idea behind the consistency checks is to maintain the flexibility of the Data Modeler by not requiring all models, entity types, and so on to be consistent in each work step.

In the Dictionary, on the other hand, an attempt is made to ensure complete consistency each time an object is created or changed. This is important because the Dictionary objects form the basis for the programs that are executed and for database accesses. However, from the point of view of the modeler, it would be annoying to have to ensure that the complete model has a consistent status at all times. For this reason, only a few checks are made when a model is being created.

The modeler should first create the objects and then check for inconsistencies in a separate check procedure. This is the purpose of the consistency checks. Once the checks have been made, the errors can be corrected.

It is also possible that referenced objects are forgotten when data models, entity types, and so on are transported. Inconsistencies may arise in the target system, even though the models are consistent in the source system. Here too, the inconsistencies can be located using the consistency checks and an appropriate supplementary transport can be organized.

The following important checks currently exist:

Checking data models for completeness

A check is made to find relationships and specializations whose source entity types do **not** belong to the data model. Inconsistencies of this type indicate that not all assignments have been made. This check is not normally appropriate for application data models.



Let data model DM1 comprise the entity types E1, E2, and E3. If, for example, there is a relationship from an entity type E4 to E2 or a generalization E5 for the specialization E3, then the check criterion 'completeness' has been violated twice (inconsistent relationship: E4 to E2, specialization/generalization to be checked: E5 to E3).

Checking data models for the existence of predecessors

A check is made to find the relationships and specializations whose source entity types do **not** exist. The most likely cause of such inconsistencies is a transport error.



Let the data model DM1 comprise the entity types E1, E2, and E3. If, for example, entity type E2 does not exist, the check criterion has been violated.

Checking data models for connectivity

A check is made to ensure that the entity types of a data model are connective. That is, a check is made to see whether there is a path connecting each entity type of the data model to each other entity type of the model. These paths are either relationships or specializations.

If the data model is **not** connective, the set of entity types making up the data model disintegrates into several unconnected subsets. Each of these subsets, however, is in itself connective. Inconsistencies of this type indicate that relationships are missing.

Consistency checks



Let data model DM1 contain the entity types E1, E2, E3, E4, and E5. If a relationship exists between E1 and E2 and between E3 and E4, the data model is not connective. It is divided into three subsets (E1, E2), (E3, E4) and (E5).

Checking hierarchies for the existence of hierarchy objects

A check is made here to find data models and entity types that are included in the hierarchy, but that do **not** exist. Inconsistencies of this type are generally caused by a transport error.



Let data model DM1 include entity types E1, E2 and E3 and submodels DM2 and DM3. If data model DM3 and entity type E2 do not exist, the check criterion is violated twice.

Other subjects:

[Data Modeler: overview \[Page 12\]](#)

[Entity types \[Page 14\]](#)

[Attributes of entity types \[Page 16\]](#)

[Relationships \[Page 18\]](#)

[Specializations/generalizations \[Page 22\]](#)

[Structural concept: data model and data model hierarchy \[Page 24\]](#)

[Top-down and bottom-up modeling \[Page 26\]](#)

Integration of the Data Modeler with the Business Navigator and the Business Object Repository

You can navigate in both the Business Navigator and the Business Object Repository from the Data Modeler:

How to...

[navigate to the Business Navigator \[Page 145\]](#)

[navigate in the Business Object Repository \[Page 146\]](#)

Overview of the data model

Overview of the data model

In the initial screen of the Data Modeler you can call the SAP Architecture model, the SAP Application model or other data models.

SAP Architecture model

The SAP Architecture model is the overall model of the R/3 System. It contains all information objects and their relationships with each other.

SAP Application data model

The SAP Application data models are views of the Architecture model. They contain only information objects and relationships which are significant for each application.

Other models

Other models include both SAP models which are used for courses or documentation, and the customer-specific models.

Accessing the Data Modeler

There are a number of different ways to call the Data Modeler modeling tool:

[Access from the menu \[Page 32\]](#)

[Access from the Object Browser of the ABAP/4 Development Workbench \[Page 33\]](#)

[Access from the ABAP/4 Repository Information System \[Page 34\]](#)

You can access the hierarchy of all application models from the Data Modeler initial screen:

[Accessing the hierarchy of SAP models \[Page 35\]](#)

Access from the menu

Access from the menu

This is the most direct and quickest method of accessing the Data Modeler.

How to...

[access the Data Modeler via the menu \[Page 147\]](#)

Other subjects:

[Access from the Object Browser of the ABAP/4 Development Workbench \[Page 33\]](#)

[Access from the ABAP/4 Repository Information System \[Page 34\]](#)

[Accessing the hierarchy of SAP models \[Page 35\]](#)

Access from the Object Browser of the ABAP/4 Development Workbench

You can also access the Data Modeler via the Object Browser.

How to...

[access the Data Modeler via the Object Browser \[Page 148\]](#)

Other subjects:

[Access from the menu \[Page 32\]](#)

[Access from the ABAP/4 Repository Information System \[Page 34\]](#)

[Accessing the hierarchy of SAP models \[Page 35\]](#)

Access from the ABAP/4 Repository Information System

You can also access the Data Modeler via the ABAP/4 Repository Information System.

How to...

[access the Data Modeler via the ABAP/4 Repository Information System \[Page 150\]](#)

Other subjects:

[Access from the menu \[Page 32\]](#)

[Access from the Object Browser of the ABAP/4 Development Workbench \[Page 33\]](#)

[Accessing the hierarchy of SAP models \[Page 35\]](#)

Accessing the hierarchy of SAP models

You can access the hierarchy of the SAP models directly from the initial screen of the Data Modeler. As this hierarchy contains all application models supplied by SAP, you can quickly gain an overview of the models and navigate to any specific model you are interested in.

How to...

[access the hierarchy of the SAP Model \[Page 151\]](#)

Other subjects:

[Access from the menu \[Page 32\]](#)

[Access from the Object Browser of the ABAP/4 Development Workbench \[Page 33\]](#)

[Access from the ABAP/4 Repository Information System \[Page 34\]](#)

ABAP/4 Repository Information System

ABAP/4 Repository Information System

The description of the ABAP/4 Repository Information System (in short: Repository Information System) given here is restricted to aspects of relevance for work with the Data Modeler. Using the ABAP/4 Repository Information System, you can search for all modeling objects of the Data Modeler that you wish to display or edit.

The following subjects are discussed:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: overview

The ABAP/4 Repository Information System provides you with the two basic functions *Find* and *Where-used list*.

The *Find* function allows you to find objects of a specific object class, that correspond to certain selection criteria. You can, for example, search for a list of all entity types belonging to a particular development class.

The *Where-used list* function allows you to determine the other objects in which a particular object is used. For example, you could search for all data models in which a specific entity type occurs.

Other subjects:

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: access

Repository Information System: access

From the initial screen of the Data Modeler, there are two ways of calling the ABAP/4 Repository Information System.

How to...

[call the Repository Information System from the Data Modeler \[Page 152\]](#)

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: searching for objects

You can search for objects in the Data Modeler with the Repository Information System.

How to...

[search for objects with the Repository Information System \[Page 153\]](#)

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: settings

Repository Information System: settings

The selection options to be defaulted when you are searching for objects and the maximum number of objects matching the search criteria to be displayed are laid down in your user settings.

How to...

[define the user settings in the Repository Information System \[Page 154\]](#)

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: selection options

By specifying selection options for a field, you can determine how the entry you have made is analyzed during the search process.

How to...

[set the selection options in the Repository Information System \[Page 155\]](#)

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: examples for selection options

When selecting data models, **U** is entered in the field *Data model*. Then the arrow pushbutton is used and the *Options* pushbutton chosen.

After selecting *Program* → *Execute*, the output list that is displayed varies according to the selection options set for the field *Data model*.

- *Single value*: No table is selected, since there is no table of the name U.
- *Greater than or equal*: All tables with names starting with U or with a letter occurring after U in the alphabet are selected.
- *Less than or equal*: All tables with names starting with U or with a letter occurring before U in the alphabet are selected.
- *Not equal*: All tables with names that do not begin with U are selected.
- *Less than*: All tables with names starting with a letter that occurs before U in the alphabet are selected.
- *Greater than*: All tables with names starting with a letter occurring after U in the alphabet are selected.

When selecting data models, **U*** is entered in the field *Data model*. The selection options *Pattern* and *Exclude pattern* result in the following selections being made:

- *Pattern*: All tables with names starting with U are selected.
- *Exclude pattern*: All tables with names that do not begin with U are selected.

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: where-used list \[Page 43\]](#)

Repository Information System: where-used list

This function allows you to determine the other objects in which a particular object is used.

How to...

[use the Repository Information System to find out where objects are used \[Page 156\]](#)

Other subjects:

[Repository Information System: overview \[Page 37\]](#)

[Repository Information System: access \[Page 38\]](#)

[Repository Information System: searching for objects \[Page 39\]](#)

[Repository Information System: settings \[Page 40\]](#)

[Repository Information System: selection options \[Page 41\]](#)

[Repository Information System: examples for selection options \[Page 42\]](#)

Data Modeler: handling

General information on using the Data Modeler which is applicable for all modeling objects is given here:

[Version management \[Page 45\]](#)

[Workbench Organizer \[Page 46\]](#)

[Request management \[Page 48\]](#)

[User settings \[Page 49\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

[Printing objects \[Page 51\]](#)

Version management

The Data Modeler is linked to version management, that is, versions are stored for each modeling object. An object can have a number of different versions. These are distinguished by an object status entry and a version number.

A detailed description of version management can be found in the documentation for the *ABAP/4 Dictionary*.



Please note that definitions **cannot** have more than one version and that previous versions cannot, therefore, be retrieved.

Before reloading a stored version, check whether the modeling object being retrieved fits the context of the existing relationships, specializations, and table/view assignments.

Other subjects:

[Workbench Organizer \[Page 46\]](#)

[Request management \[Page 48\]](#)

[User settings \[Page 49\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

[Printing objects \[Page 51\]](#)

Workbench Organizer

Workbench Organizer

The Data Modeler is linked to the Workbench Organizer. This means that changes to modeling objects are logged and can be transported to other systems. The Workbench Organizer ensures that there is only one original version of an object. Only this original version of an object can be changed (in the system where it is located).

Interfaces to the Workbench Organizer when creating or changing objects

When you create a new object, a dialog box appears in which you are requested to assign an object to a development class. Select **Save**. A further dialog box is displayed. You can now select one of your existing requests or create a new request.

A dialog box also appears if you change an existing object and wish to save it. With this dialog box, you can select your own requests or create a new request.

If the new object is a test object, select *Local object*. Such local objects are not assigned to any request and **cannot** be transported.

Program IDs and object types in the Data Modeler

The program ID and the object type define how an object is locked or transported. The table below gives an overview of the program IDs and object types of relevance for the Data Modeler.

Program IDs

ID	Object type	Meaning
R3TR	UENO	Entity type
R3TR	UDMO	Data model

An **entity type** is always locked and transported together with the following subobjects:

- documentation (short text, definition, comment, example, aliases)
- ingoing relationships
- specializations
- generalizations
- attributes
- table or view assignment

A **data model** is always locked and transported together with the following subobjects:

- documentation (short text, definition)
- direct successor in the hierarchy
- graphics positions of the direct successors

You will find a detailed description of the Workbench Organizer in the documentation *Workbench Organizer*.

Other subjects:

[Version management \[Page 45\]](#)

[Request management \[Page 48\]](#)

[User settings \[Page 49\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

[Printing objects \[Page 51\]](#)

Request management

Request management

The request management function allows you to define a Workbench Organizer request and a development class as default values for your work. You can also change any existing entries.

You can access request management from the Data Modeler with *Utilities* → *Request management*.

Here you can, for example, set and cancel standard requests, add and delete employees and select the requests of other users.

For further information about the Workbench Organizer, see the *BC Workbench Organizer* documentation.

Other subjects:

[Version management \[Page 45\]](#)

[Workbench Organizer \[Page 46\]](#)

[User settings \[Page 49\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

[Printing objects \[Page 51\]](#)

User settings

You can set the following user defaults in the Data Modeler:

- *Display customizing info*
If this option is selected, the corresponding field and its contents are displayed on screen and can be changed.
- *Rel. text in graphic*
If this option is selected, relationship texts are included in the graphics (where these exist).

How to...

[define the user settings for the Data Modeler \[Page 157\]](#)

Other subjects:

[Version management \[Page 45\]](#)

[Workbench Organizer \[Page 46\]](#)

[Request management \[Page 48\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

[Printing objects \[Page 51\]](#)

Tool (pre 3.0)

Tool (pre 3.0)

From the Data Modeler you can access the *Information Model* (the tool available prior to Release 3.0). With this tool you can display the data models supplied by SAP in Releases prior to 3.0 (Finance and Materials Management data models, for example).

To access the Information Model, select *Goto* → *Tool (pre 3.0)* in the *Data Modeler: Initial Screen*. You can access the SAP data models by selecting individual modeling objects, the analysis function or the graphics.

Other subjects:

[Version management \[Page 45\]](#)

[Workbench Organizer \[Page 46\]](#)

[Request management \[Page 48\]](#)

[User settings \[Page 49\]](#)

[Printing objects \[Page 51\]](#)

Printing objects

The Data Modeler allows you to print all objects (data model hierarchies, data models, entity types, and their subobjects). For data models and entity types you also have an option allowing you to specify exactly which of your subobjects you would like to be printed.

How to...

[print objects \[Page 158\]](#)

Other subjects:

[Version management \[Page 45\]](#)

[Workbench Organizer \[Page 46\]](#)

[Request management \[Page 48\]](#)

[User settings \[Page 49\]](#)

[Tool \(pre 3.0\) \[Page 50\]](#)

Data Modeler: maintaining entity types

This section contains information on displaying/printing, creating, changing and deleting entity types and their subobjects:

Definition of an entity type

[Entity type: definition \[Page 53\]](#)



You cannot create subobjects of an entity type (attributes or relationships) until you have created the entity type definition.

Entity type subobjects:

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

Data, hypertext, variants of entity type definition:

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

Switching to a different entity type, use of an entity type in models (inverse hierarchy):

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: definition

These subjects are dealt with here:

- [Displaying a definition](#)
- [Creating a definition](#)
- [Changing a definition](#)
- [Deleting a definition](#)
- [Branching from the definition to the graphics function](#)

How to...

[display the definition of an entity type \[Page 159\]](#)

[create the definition of an entity type \[Page 160\]](#)

[change the definition of an entity type \[Page 161\]](#)

[delete the definition of an entity type \[Page 162\]](#)

[branch from the definition of an entity type to the graphics function \[Page 163\]](#)

Other subjects:

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Time-dependent (entity type)

Time-dependent (entity type)

Time-dependent signifies that an entity type or its entities may or may not exist depending on the time or time period.



The entity type *Plant material-Consumption* is an example of a time-dependent entity type. *Plant material-Consumption* contains the amount of plant material consumed within a specified time period. Since the consumption data for a particular period of time is not known until this time period has expired, no entries can exist for the current time period.

Customizing object

An indicator shows whether a modeling object (entity type or data model) can be used in customizing.

You can specify here how a modeling object can be used:

- cannot be used in customizing
- can only be used in customizing
- for general use

Press **F4** to obtain a list of possible entries.

Assigned table/view

Assigned table/view

Here you find the name of the table or view assigned to the entity type.

How to...

[assign a table or a view to an entity type \[Page 164\]](#)

Entity type: Attributes

This section contains information on the following:

- General information on attributes
- Displaying attributes
- Creating attributes
- Maintaining attributes
- Deleting attributes
- Create table and transfer attributes

General information on attributes

Attributes that describe the entities of an entity type are normally assigned to an entity type. In other words, the attributes of an entity type determine the properties it can have.

Attributes are either assigned directly to an entity type, taken over from the ABAP/4 Dictionary or inherited from a generalization.

Attributes that were transferred to the entity type when a table or view was assigned are indicated by an entry in the column *Dic*. Attributes that the entity type inherited from the generalization are indicated by an entry in the column *Inh*.

See also [Entity type attributes \[Page 16\]](#).

How to...

[Display the attributes of an entity type \[Page 165\]](#)

[Create the attributes of an entity type \[Page 166\]](#)

[Change the attributes of an entity type \[Page 167\]](#)

[Delete the attributes of an entity type \[Page 168\]](#)

[Create tables and copy attributes \[Page 169\]](#)

Other subjects

[Entity type: Definition \[Page 53\]](#)

[Entity type: Relationship \[Page 59\]](#)

[Entity type Generalization \[Page 66\]](#)

[Entity type: Specialization/Specialization type \[Page 67\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: Comment \[Page 71\]](#)

[Entity type: Example \[Page 72\]](#)

[Entity type: Alias names \[Page 73\]](#)

[Entity type: Data Browser \[Page 74\]](#)

[Entity type: Hypertext \[Page 75\]](#)

Entity type: Attributes

[Entity type: Variants \[Page 76\]](#)

[Entity type: Change entity type \[Page 78\]](#)

[Entity type: Use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: Change display \[Page 80\]](#)

[Entity type: Print \[Page 81\]](#)

Entity type: attributes

These subjects are dealt with here:

- General information on attributes
- Displaying attributes
- Creating attributes
- Maintaining attributes
- Deleting attributes
- Create table and transfer attributes

General information on attributes

Attributes that describe the entities of an entity type are normally assigned to an entity type. In other words, the attributes of an entity type determine the properties it can have.

Attributes are either assigned directly to an entity type, taken over from the ABAP/4 Dictionary or inherited from a generalization.

Attributes that were transferred to the entity type when a table or view was assigned are indicated by an entry in the column *Dic*. Attributes that the entity type inherited from the generalization are indicated by an entry in the column *Inh*.

See also Attributes of entity types.

How to...

display the attributes of an entity type

create the attributes of an entity type

change the attributes of an entity type

delete the attributes of an entity type

create tables and transfer attributes

Entity type: attributes**Other subjects**

[Entity type: definition \[Page 53\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Relationship role

When more than one relationship exists between two entity types, each separate relationship must be identified through the assignment of a unique relationship role number when it is created.

This relationship role number is a single-digit number.

You should assign the first relationship between two entity types the number 1, the second relationship between these two entity types the number 2 and so on.

Press *F4* on the field *Relationship role* to see the existing role numbers.

Relationship: category

Relationship: category

The relationship category describes the relationship from the viewpoint of the target entity type. The category can be hierarchical, aggregating, referential or external.

Hierarchical:

The target entity type is identified by the source entity type and is also dependent on the source entity type for its existence. The key of the source entity type becomes part of the key of the target entity type. You may not change the relationship between two entities.



The relationship *Offers* with the cardinality 1:CN exists between the entity types *Faculty* (source entity type) and *Courses* (target entity type).

The source entity type *Faculty* has the attributes *Faculty number* (key attribute) and *Faculty name*.

The target entity type *Courses* has the attributes *Faculty number* (key attribute), *Course number* (key attribute), *Number of the course instructor* and *Course title*.



Since, with this category of relationship, the source entity identifies the associated target entity by means of a part of the key attributes, a change to a relationship would mean that the target entity would be deleted and a new target entity would be created.

Aggregating:

The target entity type is identified by more than one source entity type and is dependent for its existence on the source entity type. The keys of the source entity types become part of the canonic key of the target entity type.

There continue to be special cases where the target entity type is not identified by one or more of the source entity types involved.

Referential:

The target entity type simply references the source entity type, that is, it is not identified by the source entity type. The key attributes of the source entity type are transferred to the target entity type as non-key attributes. The relationship between two entities may not be changed.



The relationship *Teaches* with cardinality 1:CN exists between the entity types *Professor* (source entity type) and *Courses* (target entity type).

The source entity type *Professor* has the attributes *Number* (key attribute), *Name*, *Address*, and *Remuneration class*.

The target entity type *Courses* has the attributes *Faculty number* (key attribute), *Course number* (key attribute), *Number of the course instructor* and *Course name*.

External:

A relationship between an entity type within a data model and an entity type outside the data model is described as an 'external' relationship.

Relationship: cardinality**Relationship: cardinality**

The cardinality ($n : m$) describes the relationship with respect to the number of dependent entities (entities of the target entity type) and referenced entities (entities of the source entity type) in the relationship.

The possible values for the left and right sides of the cardinality have the following significance:

$n = 1$

Each dependent entity has one referenced entity.

$n = C$

Dependent entities can exist that do not refer to an entity of the source entity type.

Referential relationships with the additional characteristic that a relationship can be defined or deleted for an entity irrespective of when the entity was created are referred to as **temporarily referential**.

$m = 1$

Each entity of the source entity type has one dependent entity.

$m = C$

Each entity of the source entity type has a maximum of one dependent entity.

$m = N$

Each entity of the source entity type has at least one dependent entity.

$m = CN$

Each entity of the source entity type can have any number of dependent entities.



The Structured Entity Relationship Model (SERM) does not allow $m:n$ relationships (i.e. many-to-many relationships). Therefore, entity types in SERM can be arranged from left to right, from independent to dependent entity type.



The cardinality $C:x$ ($x = 1, C, CN, N$) should be used only for relationships of the category 'referential'. (However, it can also be used for relationships of the category 'aggregating'.)

Hierarchical relationships do not allow this cardinality, since all dependent entities must refer to an entity of the source entity type.

Relationship: temporary reference

This additional attribute of a relationship should only be employed for relationships of the category **referential** where the left side of the cardinality has the value C.

A temporary referential relationship allows you to create or delete a relationship to an entity **irrespective** of when the entity was created.

In all other cases, you have to decide whether to refer to another entity by means of a relationship at the time of creating an entity.

Entity type: generalizations

Entity type: generalizations

These subjects are dealt with here:

- General information on generalizations
- Displaying generalizations
- Creating generalizations

General information on generalizations

The aim of the concept of specializations and generalizations is to allow the subdivision of entity types into subtypes. There are a number of different categories of specialization. The specialization category determines the criterion for the specialization. Specializations are linked to the source entity type (=generalization) via the specialization category.

See also [Specializations/generalizations \[Page 22\]](#).

How to...

[display generalizations \[Page 174\]](#)

[create generalizations \[Page 175\]](#)

[delete generalizations \[Page 176\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: specializations/specialization categories

This section contains information on the following:

- General information on attributes
- Displaying specializations/specialization categories
- Creating specializations/specialization categories
- Changing specialization categories
- Deleting specializations/specialization categories

General information on attributes

The aim of the concept of specializations and generalizations is to allow the subdivision of entity types into subtypes. There are a number of different categories of specialization. The specialization category determines the criterion for the specialization. Specializations are linked to the source entity type (=generalization) via the specialization category.

See also [Specializations/Generalizations \[Page 22\]](#).

How to...

[Display specializations/specialization categories \[Page 177\]](#)

[Create specializations/specialization categories \[Page 178\]](#)

[Change a specialization category \[Page 179\]](#)

[Delete a specialization category/specialization \[Page 180\]](#)

Other subjects

[Entity type: Definition \[Page 53\]](#)

[Entity type: Attributes \[Page 57\]](#)

[Entity type: Relationship \[Page 59\]](#)

[Entity type: Generalizations \[Page 66\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: Comment \[Page 71\]](#)

[Entity type: Example \[Page 72\]](#)

[Entity type: Alias names \[Page 73\]](#)

[Entity type: Data Browser \[Page 74\]](#)

[Entity type: Hypertext \[Page 75\]](#)

[Entity type: Variants \[Page 76\]](#)

[Entity type: Change entity type \[Page 78\]](#)

[Entity type: Use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy : Change display \[Page 80\]](#)

[Entity type: Print \[Page 81\]](#)

Entity type: specializations/specialization categories

Specialization category: attributes

A specialization category can be characterized by the following attributes:

- **complete:**

Each entity of the generalization occurs at least once in a specialization of the category.

- **disjoint:**

Each entity of the generalization occurs in a maximum of one specialization of the category.



The specialization category *Activity* is complete in the university example, since each university member belongs to at least one specialization. This specialization category is **not** disjoint, however, since it is conceivable that a person might be at the same time both a student and a research assistant with the result that they would occur in both specializations.



A specialization category need be neither complete nor disjoint.

Entity type: Dictionary assignment

Entity type: Dictionary assignment

You can assign a table or a view to an entity type with the *Dictionary assignment* function. You can also note tables for future use.

How to...

[display the Dictionary assignment of an entity type \[Page 181\]](#)

[create or change the Dictionary assignment of an entity type \[Page 182\]](#)

[delete the Dictionary assignment of an entity type \[Page 184\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: comment

You can enter additional information or explanations for an entity type in a comment.

How to...

[display, create or change a comment for an entity type \[Page 185\]](#)

[delete a comment for an entity type \[Page 186\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: example

Entity type: example

You can enter an example for an entity type.

How to...

[display, create or change an example for an entity type \[Page 187\]](#)

[delete an example for an entity type \[Page 188\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: aliases

You can define aliases for an entity type. An alias is an alternative term used for an entity type in a different technical field.

How to...

[delete, create or change an alias for an entity type \[Page 189\]](#)

[delete an alias for an entity \[Page 190\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: Data Browser

Entity type: Data Browser

The Data Browser allows you to select and display entities (data) of an entity type. However, this is only possible if a table or view (database or projection view) is assigned to the entity type.

How to...

[display the contents of the table/view which is assigned to an entity type \[Page 191\]](#)

Other subjects

[Entity type: Definition \[Page 53\]](#)

[Entity type: Attributes \[Page 57\]](#)

[Entity type: Relationship \[Page 59\]](#)

[Entity type: Generalizations \[Page 66\]](#)

[Entity type: Specialization/Specialization type \[Page 67\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: Comment \[Page 71\]](#)

[Entity type: Example \[Page 72\]](#)

[Entity type: Alias names \[Page 73\]](#)

[Entity type: Hypertext \[Page 75\]](#)

[Entity type: Variants \[Page 76\]](#)

[Entity type: Change entity type \[Page 78\]](#)

[Entity type: Use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy : Change display \[Page 80\]](#)

[Entity type: Print \[Page 81\]](#)

Entity type: hypertext

The hypertext for an entity type contains the definition of the entity type and links to any other texts (such as comments) existing for this entity type.

How to...

[view the hypertext of an entity type \[Page 192\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: variants

Entity type: variants

These subjects are dealt with here:

- General information on variants
- Creating variants
- Displaying/maintaining variants
- Deleting variants

General information on variants

A variant refers to the definition of an entity type, that is, you can assign several definitions to an entity type using variants.

It is possible, therefore, for an entity type to have a number of different definitions in the modeling process, particularly if the entity type is being used by more than one developer.

The first definition of an entity type, which is also its first variant, is assigned the ID number 00. The ID numbers for the other variants are incremented in each case by 1.



Once the modeling process is complete, all variants must be deleted. Only variant 00, i.e. the definition of the entity type is transported.

If you wish to display or change an entity type for which variants have been created, enter the name of the entity type and activate either *Display* or *Change* to display all the variants together with their corresponding ID numbers in a dialog box. Choose the required variant in this dialog box.

How to...

[create a variant for an entity type \[Page 193\]](#)

[display or change a variant for an entity type \[Page 194\]](#)

[delete a variant for an entity type \[Page 195\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: switching to a different entity type

Entity type: switching to a different entity type

From the definition screen of one entity type you can switch to the definition screen of another entity type.

How to...

[switch to another entity type \[Page 196\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: use in data models (inverse hierarchy)

The inverse hierarchy shows the data models to which a selected entity type is assigned.

How to...

[determine the assignment of an entity type to data models \[Page 197\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Inverse hierarchy: changing the display

Inverse hierarchy: changing the display

You can change the display of the inverse hierarchy by:

- Expanding a sub-tree
- Compressing a sub-tree
- Setting the focus

How to...

[change the display of the inverse hierarchy \[Page 198\]](#)

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: specializations/specialization categories \[Page 81\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

Entity type: specializations/specialization categories

These subjects are dealt with here:

- General information on specializations/specialization categories
- Displaying specializations/specialization categories
- Creating specializations/specialization categories
- Changing specialization categories
- Deleting specializations/specialization categories

General information on specializations/specialization categories

The aim of the concept of specializations and generalizations is to allow the subdivision of entity types into subtypes. There are a number of different categories of specialization. The specialization category determines the criterion for the specialization. Specializations are linked to the source entity type (=generalization) via the specialization category.

See also Specializations/generalizations.

How to...

display specializations/specialization categories

create specializations/specialization categories

change the specialization category_

delete specialization category/specializations

Other subjects:

[Entity type: definition \[Page 53\]](#)

[Entity type: attributes \[Page 59\]](#)

[Entity type: generalizations \[Page 66\]](#)

[Entity type: Dictionary assignment \[Page 70\]](#)

[Entity type: comment \[Page 71\]](#)

[Entity type: example \[Page 72\]](#)

[Entity type: aliases \[Page 73\]](#)

[Entity type: hypertext \[Page 75\]](#)

[Entity type: variants \[Page 76\]](#)

[Entity type: switching to a different entity type \[Page 78\]](#)

[Entity type: use in data models \(inverse hierarchy\) \[Page 79\]](#)

[Inverse hierarchy: changing the display \[Page 80\]](#)

Entity type: specializations/specialization categories

Data Modeler: maintaining data models

The following subjects are dealt with here:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Displaying/printing data models

How to...

[display and print data models \[Page 200\]](#)

For more information, refer to [Printing objects \[Page 51\]](#)

Other subjects:

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Creating data models

How to

[create a data model \[Page 201\]](#)

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Data Model: Type

Data Model: Type

Application model A model is termed an application model if it forms an independent unit from a commercial viewpoint, for example, the *FI* and *MM* application data models.

Business object

A business object is a set of entity types sharing a common external interface. A business object normally consists of a source entity type and those entity types hierarchically dependent on it. The source entity type normally functions as representative of the business object to which it belongs.

Customizing object

An indicator shows whether a modeling object (entity type or data model) can be used in customizing.

You can specify the attribute you wish to assign a modeling object here:

- cannot be used in customizing
- can only be used in customizing
- for general use

To display a list of possible entries, press **F4**.

Data model: references

Represent.

The representative of a data model is the entity type that represents the entity types belonging to the data model when the model is compressed.

Ref. model

The referenced model is the data model on which the current data model is based. A data model is normally derived from the referenced model by means of projection.

Changing a data model

Changing a data model

If you do not know the precise name of the data model to be changed, start by looking for it in the ABAP/4 Repository Information System. Select the modeling object *Data model* in the *Data Modeler: Initial Screen* and click on *Find*. For more information on this, refer to the [ABAP/4 Repository Information System \[Page 36\]](#).

How to...

[change a data model \[Page 202\]](#)

The hierarchy is described in [Data Modeler: hierarchy \[Page 98\]](#).

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Deleting a data model

If you do not know the precise name of the data model to be deleted, start by looking for it in the ABAP/4 Repository Information System. Select the modeling object *Data model* in the *Data Modeler: Initial Screen* and click on *Find*. For more information on this, refer to the [ABAP/4 Repository Information System \[Page 36\]](#).

How to...

[delete a data model \[Page 203\]](#)

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Switching to a different data model

Switching to a different data model

You can switch to another data model from the definition screen of a data model.

How to...

[switch to another data model \[Page 205\]](#)

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Data model: making consistency checks

Data models can be checked for consistency. The following checks are available for this:

- [Check: completeness \[Page 92\]](#)
- [Check: existence of predecessors \[Page 93\]](#)
- [Check: connectivity \[Page 94\]](#)
- [Check: existence of hierarchy \[Page 95\]](#)

How to...

[carry out consistency checks \[Page 206\]](#)

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Consistency check: completeness

Consistency check: completeness

A data model is complete if it contains neither inconsistent relationships nor inconsistent specializations.

Relationships and specializations are inconsistent if their source entity types do not belong to the data model.

Consistency check: existence of predecessors

The predecessors of a data model exist if the data model has neither inconsistent relationships nor inconsistent specializations.

Relationships and specializations are considered as inconsistent in this check if their source entity types do not exist.

Consistency check: connectivity

Consistency check: connectivity

A data model is connective if all the entity types of the data model are connected.

All the entity types of a data model are connective if there is a path connecting each entity type of the data model to each other entity type of the model. These paths are either relationships or specializations.

A data model is not connective if the set of entity types making up the data model disintegrates into several unconnected subsets. Each of these subsets, however, is in itself connective.

If the number of subsets in the check log is greater than 1, the data model is not connective.

Consistency check: existence of hierarchy objects

All the objects of a hierarchy exist if there are neither inconsistent data models nor inconsistent entity types among the objects of this hierarchy.

Data models and entity types are inconsistent in this check if they are referenced in the hierarchy but do not exist.

Data model: use in other data models (inverse hierarchy)

Data model: use in other data models (inverse hierarchy)

The inverse hierarchy shows the data models to which a selected data model is assigned.

How to...

[find out in which other data models a data model is used \[Page 207\]](#)

You can change the display of the inverse hierarchy. For information on how to do this, please refer to [Inverse hierarchy: changing the display \[Page 97\]](#).



You can also call a general where-used list for modeling objects with *Utilities* → *Where-used list*. For information on this, please refer to the [ABAP/4 Repository Information System \[Page 36\]](#).

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Inverse hierarchy: changing the display \[Page 97\]](#)

Inverse hierarchy: changing the display

You can change the display of the inverse hierarchy by:

- Expanding a sub-tree
- Compressing a sub-tree
- Setting the focus

How to...

[change the display of the inverse hierarchy \[Page 208\]](#)

Other subjects:

[Displaying/printing data models \[Page 84\]](#)

[Creating data models \[Page 85\]](#)

[Changing a data model \[Page 88\]](#)

[Deleting a data model \[Page 89\]](#)

[Switching to a different data model \[Page 90\]](#)

[Data model: making consistency checks \[Page 91\]](#)

[Data model: use in other data models \(inverse hierarchy\) \[Page 96\]](#)

Data Modeler: hierarchy

The *data model hierarchy* function allows you to display and make changes to data model hierarchies.

The data model from which you access the hierarchy is displayed as the root. The objects on the level below assigned to the data model are also listed.

The following subjects are dealt with here:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Hierarchy: access

You can access the hierarchy either from the Data Modeler initial screen or from the data model definition:

How to...

[access the data model \[Page 209\]](#)

Other subjects:

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Hierarchy: printing

Hierarchy: printing

How to...

[print the data model hierarchy \[Page 211\]](#)

For more information, refer to [Printing objects \[Page 51\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Hierarchy: changing the display

These options are available for changing the display of the hierarchy in display and maintain modes:

- Hierarchy: refreshing the display
- Hierarchy: expanding a sub-tree
- Hierarchy: compressing a sub-tree
- Hierarchy: setting the focus

How to...

[change the display of the data model hierarchy \[Page 212\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Changing a hierarchy

Changing a hierarchy

To edit a data model hierarchy, you must be in the screen *Change Data Model: Hierarchy*. If you are in display mode (*Display Data Model: Hierarchy*), you can access change mode by activating *Display <-> Change*.

The following options are available for changing the hierarchy display:

[Changing a hierarchy: inserting objects \[Page 103\]](#)

[Changing a hierarchy: deleting assignments \[Page 104\]](#)

[Changing a hierarchy: reassigning modeling objects \[Page 105\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Changing a hierarchy: inserting objects

You can insert objects in the data model hierarchy.

How to...

[insert objects in the data model hierarchy \[Page 213\]](#)

Changing a hierarchy: deleting assignments

Changing a hierarchy: deleting assignments

You delete the assignments of objects in the data model hierarchy.

How to...

[delete assignments in the data model hierarchy \[Page 215\]](#)

Changing a hierarchy: reassigning modeling objects

You can reassign objects in the data model hierarchy.

How to...

[reassign objects in the data model hierarchy \[Page 216\]](#)

Hierarchy: switching data models

Hierarchy: switching data models

You can switch to the hierarchy of another data model.

How to...

[switch to the hierarchy of another data model \[Page 217\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Hierarchy: making consistency checks

You can make consistency checks from the data model hierarchy.

How to...

[carry out consistency checks from the data model hierarchy \[Page 218\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Consistency checks: completeness**Consistency checks: completeness**

A data model is complete if it contains neither inconsistent relationships nor inconsistent specializations.

Relationships and specializations are inconsistent if their source entity types do not belong to the data model.

Consistency check: connectivity

A data model is connective if all the entity types of the data model are connected.

All the entity types of a data model are connective if there is a path connecting each entity type of the data model to each other entity type of the data model. These paths are either relationships or specializations.

A data model is not connective if the set of entity types making up the data model disintegrates into several unconnected subsets. Each of these subsets, however, is in itself connective.

If the number of subsets in the check log is greater than 1, the data model is not connective.

Consistency check: existence of predecessors**Consistency check: existence of predecessors**

The predecessors of a data model exist if there are neither inconsistent relationships nor inconsistent specializations.

Relationships and specializations are considered as inconsistent in this check if their source entity types do not exist.

Consistency check: existence of hierarchy objects

All the objects of a hierarchy exist if there are neither inconsistent data models nor inconsistent entity types among the objects of this hierarchy.

Data models and entity types are inconsistent in this check if they are referenced in the hierarchy but do not exist.

Hierarchy: accessing the graphics

Hierarchy: accessing the graphics

You can access the graphical display of the data model hierarchy from within with the data model hierarchy.

How to...

[switch to the graphic function from the data model hierarchy \[Page 219\]](#)

For information on how to display and edit the graphics, please refer to [Data Modeler: graphics \[Page 114\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: graphics positions \[Page 113\]](#)

Hierarchy: graphics positions

You can access the graphics positions of modeling objects, that is, data models and entity types, from within with the data model hierarchy.

However, you can only display or change the graphics positions of the **successors** of the data model from which you accessed the graphics positions. The graphics positions of the successors are relative to those of the source object.

How to...

[switch to the graphics positions of modeling objects from the data model hierarchy \[Page 220\]](#)

Other subjects:

[Hierarchy: access \[Page 99\]](#)

[Hierarchy: printing \[Page 100\]](#)

[Hierarchy: changing the display \[Page 101\]](#)

[Changing a hierarchy \[Page 102\]](#)

[Hierarchy: switching data models \[Page 106\]](#)

[Hierarchy: making consistency checks \[Page 107\]](#)

[Hierarchy: accessing the graphics \[Page 112\]](#)

Data Modeler: graphics

The Data Modeler's graphics function allows you to display the sections of the real world that have been modeled in the form of extended Structured Entity Relationship Models (SAP SERM).

The following subjects are dealt with here:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: display method (SAP-SERM)

The graphics function provided by the Data Modeler allows you to display the area of the Real World that has been modeled as an extended Structured Entity Relationship Model (SAP-SERM).

The methods of the Structured Entity Relationship Model (SERM) were extended by SAP to include additional semantic relationship categories and a data model hierarchy. This extended model is called SAP-SERM. The SAP-SERM graphics allow you to display even the most complex models in a form that is both clear and easy to understand.

The attribute 'structured' means that the arrangement of the entity types in the graphics is determined by their dependency factor. If two entity types are linked by means of a relationship or specialization, the source entity type (referenced entity type) will always be located to the left of the target entity type (dependent entity type). This structured arrangement of entity types proceeding from left to right facilitates navigation in complex data models. When a graphic is called, the layout mechanism incorporated into the graphics function automatically ensures that the entity types are positioned appropriately for their dependency factor.

The data model hierarchy also contributes towards improving the clarity of the graphical representation. Each data model can contain entity types and other data models. Data models can be assigned different colors. The data model hierarchy thus appears in the graphic as a sequence of nested areas of varying colors. Finally, each data model can be allocated a position relative to the superordinate data model. This ensures that the layout within a data model (arrangement of submodels and entity types) always remains the same irrespective of the environment in which this data model appears in the graphic. However, it is important to note the following. Unlike entity types, data models are not positioned automatically by the graphic. For the time being, the positions of the data models have to be entered by hand (see also [Hierarchy: graphics positions \[Page 113\]](#)).

For more information on the layout, please refer to the section *Layout*.

Graphical display of the various modeling objects

Entity types

In the graphics function entity types are shown as rectangles, which are subdivided into fields. The upper left-hand field contains the entity type ID. The lower, four-line field contains the short description of the entity type.

In the upper right-hand corner there are two smaller fields. The left-hand field contains the customizing code, while the right-hand field specifies the type of Dictionary assignment defined for the entity type.

Possible values for the field for the customizing code are:

Customizing code

Field contents	Meaning
Blank	Entity type is not used in customizing
C	Entity type is used only in customizing
A	Entity type is used generally

Graphics: display method (SAP-SERM)

Possible values for the type of Dictionary assignment are:

Dictionary assignment type

Field contents	Meaning
Blank	No table/view assigned
T	Table assigned
V	View assigned

If an entity type is time-dependent, this is shown graphically by an oval in the lower left-hand corner of the entity type.

Relationships

Relationships are shown in the graphics as narrow black lines. A label above the line indicates the relationship category involved.

Letter	Category
H	hierarchical
A	aggregating
R	referential
X	external

The relationship text (for example, professor **supervises** student) is located below the line. You can define whether or not you wish this text to be displayed in the user settings (for more information on this, please refer to [User settings \[Page 49\]](#)).

The right-hand side of the cardinality is shown by an arrow symbol at the junction with the target entity type. These symbols are used:

Arrow symbols

Symbol	Cardinality of the dependent entity type
1 point	1
Vertical line plus 1 point	c
2 points	n
Vertical line plus 2 points	cn

The category of relationship involved is also shown by the direction from which the relationship enters the entity type. Hierarchical and aggregating relationships enter from the left, referential relationships from above or below.

Specialization categories, specializations

A specialization category of an entity type is represented by a blue triangle, which is linked to the entity type by a broad blue line. The triangle functions as a "junction" for the specializations. These are also shown as broad blue lines, proceeding from the specialization category to the corresponding entity types.

Data models

Data models are shown as rectangular colored areas. The short text for the data model appears in the upper left-hand corner. All entity types and submodels of the data model are contained in the colored area. Through this method of representation employing nested colored areas, the clarity of the data model is enhanced.

Layout

Data models (frames)

The manual position maintenance function (see [Hierarchy: graphics positions \[Page 113\]](#)) allows you to assign a line and column position to each data model in a virtual raster. The values for the line and column positions are specified relative to the superordinate data model (differences in line and column position). On the basis of these values, the graphics function automatically calculates the absolute positions in the raster.

If the data models are not positioned manually, this will be done by the automatic layout function for entity types and relationships (see below). This may result in overlapping with parts of data models being concealed.

Entity types, specialization categories (nodes)

The manual position maintenance function (see [Hierarchy: graphics positions \[Page 113\]](#)) allows you to assign a line and column position in a virtual raster to every entity type in a data model. (This raster is a refinement of the data model raster described above). The values for the line and column positions are specified relative to the superordinate data model. On the basis of these values, the graphics function then automatically calculates the absolute positions in the raster.

As well as the manual position maintenance function, there is also a graphical position maintenance function for the entity types of a data model (see [Graphics: positioning mode \[Page 127\]](#)).

It is not necessary to enter any positions for the entity types of a data model. The entity types are arranged in the graphic by the automatic layout mechanism. The following criteria come into play here:

- data model to which an entity type belongs, position of the data model
- dependency factor (arrangement from left to right)
- optimization of link flow (minimum possible number of links crossing)

The positions of specialization categories cannot be maintained manually. These are calculated by the layout mechanism.

Graphics: display method (SAP-SERM)**Relationships, specializations (links)**

The course taken by the relationships and specializations is optimized automatically by the graphics program. The various relationship categories have different priorities. Specializations have the highest priority, that is, the corresponding entity types are located close to the generalizing entity type. Hierarchical and aggregating relationships have medium priority, referential relationships low priority.

Other subjects:

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: accessing

There are several different ways of calling the graphics:

- From within the Data Modeler
- From the Object Browser of the ABAP/4 Development Workbench
- From the ABAP/4 Repository Information System

How to...

[call the graphics function \[Page 221\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: display mode/maintain mode

Graphics: display mode/maintain mode

Display mode

In display mode the graphics window is divided into two sections. On the left is the display area which shows a section of the complete graphic. You can use the scroll bars bordering the graphics window to move around in the graphic, changing the section being displayed (see [Graphics: changing the section displayed \[Page 122\]](#).) In the navigation area, a green frame indicates the section of the complete graphic currently visible in the display area. The section can be changed by dragging the green frame or by enlarging or reducing it in size.

You can switch between display and maintain mode with *Display*<->*Change*.

Maintain mode

In maintain mode the graphics window is divided into three separate areas. On the left is the display area, on the upper right-hand side the navigation area and on the lower right-hand side the insertion area.

In the navigation area a green frame indicates the section of the graphics currently visible in the display area. The section can be changed by dragging the green frame or by enlarging or reducing it in size.

The insertion area contains a model symbol for an entity type and a specialization category (blue triangle). You will need these when you create objects of these types (see [Graphics: creating objects \[Page 125\]](#)).

In change mode some functions are available under *Edit* that are not available in display mode: *Change*, *Display*, *Connect*, *Position*.

Positioning mode

By selecting *Edit* → *Position cursor* → *Positioning on/off* you can switch between the maintain mode and the positioning mode. Positioning mode is also activated when you insert an entity type into a data model.

For more information on the positioning mode, refer to [Graphics: positioning mode \[Page 127\]](#).

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: changing the section displayed

Graphics: changing the section displayed

This section contains information on the following functions:

- Zoom function
- Selecting a section
- Centering on selecting objects
- Find

How to...

[change the section of the screen displayed in the graphics function \[Page 223\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: Select

Determining which objects can be selected

You can specify which graphics objects can be selected with *Settings* → *Select (objects)*.

The default setting is for all object types to be selectable:

- nodes (entity types, specialization categories)
- links (relationships, specializations)
- frames (data models)

How to...

[Select objects in the graphics \[Page 224\]](#)

Other subjects

[Graphics: Display method \(SAP-SERM\) \[Page 115\]](#)

[Call graphics \[Page 119\]](#)

[Graphics: Display/Change mode \[Page 120\]](#)

[Graphics: Change section displayed \[Page 122\]](#)

[Graphics: Display/Change objects \[Page 124\]](#)

[Graphics: Create objects \[Page 125\]](#)

[Graphics: Delete objects \[Page 126\]](#)

[Graphics: Positioning mode \[Page 127\]](#)

[Grafik: Get environment, Hide, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: Other functions \(process\) \[Page 129\]](#)

[Graphics: Change settings \[Page 130\]](#)

[Graphics: Print \[Page 132\]](#)

Graphics: displaying/changing objects

Graphics: displaying/changing objects

This section contains information on the following:

- displaying definitions of individual objects
- displaying/changing definitions/object components
- editing several object definitions in sequence
- entity type: displaying hypertext
- entity type: Data Browser (data display)

How to...

[display/change objects in the graphics function \[Page 225\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: creating objects

In the graphics maintain mode you can create the following objects:

- entity types
- relationships
- specialization categories
- specializations

How to...

[create objects in the graphics function \[Page 227\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: deleting objects

Graphics: deleting objects

The graphics allow you to delete the following objects:

- entity types
- relationships
- specialization categories
- specializations



Objects are physically deleted from the system, not just deleted from the graphic.

How to...

[delete objects in the graphics function \[Page 229\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: positioning mode

The graphics allow you to change the positions of entity types in a data model and store them relative to the data model.

How to...

[change the positions in the graphics \[Page 230\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: Get environment, Suppress, Reload (Utilities)

Get environment

This function allows you to load the predecessors (referenced entity types linked via relationships or generalizations) and/or successors (dependent entity types linked via relationships or specializations) into the graphic.

Suppress

To remove entity types from the graphic display you can use the *Suppress* function. (The entity types are not physically deleted.)

Reload

The function allows you to branch to the ABAP/4 Repository Information System to search for other data models or entity types, which you can then transfer to the graphic. When you do this, however, the previous contents of the graphic disappear from display.

How to...

[use the graphics utilities Get environment, Suppress and Reload \[Page 231\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: other functions (Edit)

This section contains information on the following functions:

- Compressing/expanding data models
- Adjustment
- Visibility
- Restricting the area that is visible

Visibility

You can use this function to make previously selected nodes (entity types, specialization categories) and links (relationships, specializations) invisible.

Restricting the visible area

This function allows you to restrict the visible area to the selected entity types. Direct predecessors and successors are shown as black dots and are labeled with the relevant entity type ID.

How to...

[use the graphics utilities Compress/Expand data model, Adjust, Visibility and Restrict area \[Page 232\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: changing the settings \[Page 130\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: changing the settings

Graphics: changing the settings

A number of graphics functions that do not affect the contents of the database are collected together under the menu option *Settings*. These settings apply only to the current graphic and cannot be stored.

You will find information here on the following menu options:

- Header
- Select (type)
- Select (objects)
- Adjust automatically
- Change partitioning
- Change sequence
- Movability on/off

Select (objects)

You can specify here which objects in the graphic are to be selectable (nodes, links, frames).

Header

You can make these settings under this menu option:

- Activate a header for graphic and print output
- Activate a header for print output only
- Deactivate a header for graphic and print output
- Change a header

Adjust automatically

This menu option allows you to activate or deactivate automatic adjustment.

If automatic adjustment is activated, the layout mechanism is activated each time a change is made to the graphic. With each change that is made, the layout with respect to the positioning of the nodes and the paths taken by the links is optimized.

Change partitioning

You can specify here how the inner frames are to be partitioned in maintain mode.

Change sequence

You can specify here the sequence of the areas (display, navigation, and insertion area) to be used in maintain mode. Proceed as follows:

Movability on/off

This function allows you to activate or deactivate the movability of all entity types. You can switch on this function, for example, if you wish to change the layout of the graphic manually for printing purposes.

Color assignment...

Under this menu option you can assign colors to the various areas of the graphics window.

How to...

[change the settings in the graphics function \[Page 233\]](#)



You cannot save the positions here. (For information on positioning entity types in data models, see [Graphics: positioning mode \[Page 127\]](#).)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: printing \[Page 132\]](#)

Graphics: printing

Graphics: printing

It is possible to print the contents of the graphic.

How to...

[print the graphic \[Page 234\]](#)

Other subjects:

[Graphics: display method \(SAP-SERM\) \[Page 115\]](#)

[Graphics: accessing \[Page 119\]](#)

[Graphics: display mode/maintain mode \[Page 120\]](#)

[Graphics: changing the section displayed \[Page 122\]](#)

[Graphics: selecting \[Page 123\]](#)

[Graphics: displaying/changing objects \[Page 124\]](#)

[Graphics: creating objects \[Page 125\]](#)

[Graphics: deleting objects \[Page 126\]](#)

[Graphics: positioning mode \[Page 127\]](#)

[Graphics: Get environment, Suppress, Reload \(Utilities\) \[Page 128\]](#)

[Graphics: other functions \(Edit\) \[Page 129\]](#)

[Graphics: changing the settings \[Page 130\]](#)

Using the Data Modeler

[Accessing the Data Modeler \[Page 134\]](#)

[Entity types \[Page 136\]](#)

[Attributes \[Page 137\]](#)

[Relationships \[Page 138\]](#)

[Generalization/specialization \[Page 139\]](#)

[Dictionary assignment \[Page 140\]](#)

[Data models \[Page 141\]](#)

[Data model hierarchy \[Page 142\]](#)

[Graphics \[Page 143\]](#)

[Repository Information System \[Page 135\]](#)

[Other themes \[Page 144\]](#)

Accessing the Data Modeler

[How to access the Data Modeler via the menu \[Page 147\]](#)

[How to access the Data Modeler via the Object Browser \[Page 148\]](#)

[How to access the Data Modeler via the ABAP/4 Repository Information System \[Page 150\]](#)

[How to access the hierarchy of the SAP Model \[Page 151\]](#)

Repository Information System

[How to call the Repository Information System from the Data Modeler \[Page 152\]](#)

[How to search for objects with the Repository Information System \[Page 153\]](#)

[How to define the user settings in the Repository Information System \[Page 154\]](#)

[How to set the selection options in the Repository Information System \[Page 155\]](#)

[How to use the Repository Information System to find out where objects are used \[Page 156\]](#)

Entity types

Entity types

[How to display the definition of an entity type \[Page 159\]](#)

[How to create the definition of an entity type \[Page 160\]](#)

[How to change the definition of an entity type \[Page 161\]](#)

[How to delete the definition of an entity type \[Page 162\]](#)

[How to branch from the definition of an entity type to the graphics function \[Page 163\]](#)

[How to display, create or change a comment for an entity type \[Page 185\]](#)

[How to delete a comment for an entity type \[Page 186\]](#)

[How to display, create or change an example for an entity type \[Page 187\]](#)

[How to delete an example for an entity type \[Page 188\]](#)

[How to delete, create or change an alias for an entity type \[Page 189\]](#)

[How to delete an alias for an entity \[Page 190\]](#)

[How to view the hypertext of an entity type \[Page 192\]](#)

[How to create a variant for an entity type \[Page 193\]](#)

[How to display or change a variant for an entity type \[Page 194\]](#)

[How to delete a variant for an entity type \[Page 195\]](#)

[How to switch to another entity type \[Page 196\]](#)

[How to determine the assignment of an entity type to data models \[Page 197\]](#)

[How to change the display of the inverse hierarchy \[Page 198\]](#)

[How to print entity types and their subobjects\) \[Page 199\]](#)

[How to print objects \[Page 158\]](#)

Attributes

[How to assign a table or a view to an entity type \[Page 164\]](#)

[How to display the attributes of an entity type \[Page 165\]](#)

[How to create the attributes of an entity type \[Page 166\]](#)

[How to change the attributes of an entity type \[Page 167\]](#)

[How to delete the attributes of an entity type \[Page 168\]](#)

[How to create tables and transfer attributes \[Page 169\]](#)

Relationships

Relationships

[How to display relationships \[Page 170\]](#)

[How to create relationships \[Page 171\]](#)

[How to change relationships \[Page 172\]](#)

[How to delete relationships \[Page 173\]](#)

Generalization/specialization

[How to display generalizations \[Page 174\]](#)

[How to create generalizations \[Page 175\]](#)

[How to delete generalizations \[Page 176\]](#)

[How to display specializations/specialization categories \[Page 177\]](#)

[How to create specializations/specialization categories \[Page 178\]](#)

[How to change the specialization category \[Page 179\]](#)

[How to delete specialization category/specializations \[Page 180\]](#)

Dictionary assignment

Dictionary assignment

[How to display the Dictionary assignment of an entity type \[Page 181\]](#)

[How to create or change the Dictionary assignment of an entity type \[Page 182\]](#)

[How to delete the Dictionary assignment of an entity type \[Page 184\]](#)

[How to display the contents of the table/view which is assigned to an entity type \[Page 191\]](#)

Data models

[How to display and print data models \[Page 200\]](#)

[How to create a data model \[Page 201\]](#)

[How to change a data model \[Page 202\]](#)

[How to delete a data model \[Page 203\]](#)

[How to switch to another data model \[Page 205\]](#)

[How to carry out consistency checks \[Page 206\]](#)

[How to find out in which other data models a data model is used \[Page 207\]](#)

[How to change the display of the inverse hierarchy \[Page 208\]](#)

Data model hierarchy

[How to access the data model \[Page 209\]](#)

[How to print the data model hierarchy \[Page 211\]](#)

[How to change the display of the data model hierarchy \[Page 212\]](#)

[How to insert objects in the data model hierarchy \[Page 213\]](#)

[How to delete assignments in the data model hierarchy \[Page 215\]](#)

[How to reassign objects in the data model hierarchy \[Page 216\]](#)

[How to switch to the hierarchy of another data model \[Page 217\]](#)

[How to carry out consistency checks from the data model hierarchy \[Page 218\]](#)

[How to switch to the graphic function from the data model hierarchy \[Page 219\]](#)

[How to switch to the graphics positions of modeling objects from the data model hierarchy \[Page 220\]](#)

Graphics

[How to call the graphics function \[Page 221\]](#)

[How to change the section of the screen displayed in the graphics function \[Page 223\]](#)

[How to select objects in the graphics function \[Page 224\]](#)

[How to display/change objects in the graphics function \[Page 225\]](#)

[How to create objects in the graphics function \[Page 227\]](#)

[How to delete objects in the graphics function \[Page 229\]](#)

[How to change the positions in the graphics \[Page 230\]](#)

[How to use the graphics utilities Get environment, Suppress and Reload \[Page 231\]](#)

[How to use the graphics utilities Compress/Expand data model, Adjust, Visibility and Restrict area \[Page 232\]](#)

[How to change the settings in the graphics function \[Page 233\]](#)

[How to print the graphic \[Page 234\]](#)

Other themes

Other themes

Business Object Repository

[How to navigate in the Business Object Repository \[Page 146\]](#)

Business Navigator

[How to navigate to the Business Navigator \[Page 145\]](#)

User settings

[How to define the user settings for the Data Modeler \[Page 157\]](#)

navigate to the Business Navigator

1. Switch to the definition screen of an entity type.
2. Select *Goto* → *Functions/processes*
If an assignment exists, a popup appears with the functions for which the entity type is used as an input or output parameter.
3. Select a function.
The detail view of the function is displayed.
4. You can display the process graphically via *Goto* → *Process*.



You can also switch from the graphic display to the Business Navigator by marking the entity type and selecting *Goto* → *Functions/processes*.

navigate in the Business Object Repository

navigate in the Business Object Repository

1. Switch to the definition screen of your data model.
2. Select *Goto* → *Object type*

If an assignment exists, you enter the Business Object Repository. You will find more information about the Business Object Repository in the WF - SAP Business Workflow documentation.



You can also switch from the graphic display to the Business Object Repository. To do this, select a data model and choose *Goto* → *Object type*.

Display mode

If no assignment exists, an error message is displayed.

Maintenance mode

If no assignment exists and the check box *Business object* was clicked, you will create a business object in the following. Otherwise you will generate an object type.

3. In the *Goto* menu, select the option *Object type*.
4. In the subsequent dialog box, confirm the creation of the object.
5. In the next dialog box, enter the definition of the object.



The name and the short text of the object should correspond to the name and short text of the data model.

access the Data Modeler via the menu

Select *Tools* → *ABAP/4 Workbench* in the initial screen of the R/3 System.

In the screen *ABAP/4 Development Workbench* select *Development* → *Data Modeler* to display the screen *Data Modeler: Initial Screen*.

access the Data Modeler via the Object Browser

access the Data Modeler via the Object Browser

Select *Tools* → *ABAP/4 Workbench* in the initial screen of the R/3 System.

In the screen *ABAP/4 Development Workbench* click on the pushbutton *Object Browser*. The screen *Object Browser: Initial Screen* is displayed. You can now choose whether to continue working under *Object list* or *Single object*.

Object list

Under the heading *Object list*, you can branch to object lists by selecting one of the options *Development class*, *Program*, *Function group* or *Local priv. objects*, making an entry in the relevant field, and clicking on *Display*. In the list that is now displayed, you can search for the object you require, select it by clicking on it, and choose a processing option (*Display*, for example). The Data Modeler screen appropriate for your processing type selection is displayed.

If you double-click on an object name, you branch to the relevant definition screen in display mode in the case of entity types, and to the relevant data model hierarchy in the case of data models. You can change the editing mode (*Display* ↔ *Change*) in these screens and use the functions of the Data Modeler.



Select *Development class*, enter **STFO** and click on the *Display* pushbutton. In the screen *Object Browser: Development Class STFO*, position the cursor on the node *Business Engineering* and expand it with *Edit* → *Expand subtree*. All modeling objects (data models and entity types) assigned to this development class are displayed. If you double-click on the name of a data model, the screen *Display Data Model: Hierarchy* of the Data Modeler is displayed. You can display all modeling objects assigned to the data model here.

Single object

With *Single object* you can branch directly to an individual modeling object.

Under *Single object* select the option *Business Engineering* and click on *Edit*. In the screen *Modeling Objects*, you have to select one of the options *Data model* or *Entity type*.

The procedure to be used varies according to whether you know the full name, only part of the name, or whether you do not know the name at all.

Name known

Enter the name in the entry field and choose the editing type you require (for example, *Change*).

The relevant screen of the Data Modeler is displayed. You can change the editing mode (*Disp.* ↔ *Change*) and can use the functions of the Data Modeler.

Part of name known

Enter the part of the name you know in the entry field and use * to represent the part you do not know (for example, **uni***). Click on the arrow to the right of the entry field.

A hit list of all objects corresponding to the pattern you entered appears. To copy the name of the object you require to the entry field, position the cursor on the object and click on *Choose*.

access the Data Modeler via the Object Browser

Then choose the editing type you require (for example, *Display*) in the screen *Modeling Objects*. The relevant screen of the Data Modeler appears. If required, you can now change the editing mode (*Disp.* <-> *Chnge*) and can use the functions of the Data Modeler.

Name unknown

Without making an entry in the input field, click on the arrow to the right of the field. A hit list is displayed.

Position the cursor on the object you require in the hit list and click on *Choose* to copy the name to the entry field of the screen you started from.

Then choose the editing type you require (for example, *Display*) in the screen *Modeling Objects*. The relevant screen of the Data Modeler is displayed. You can change the editing mode (*Disp.* <-> *Chnge*) and can use the functions of the Data Modeler.



Select *Modeling objects* and click on *Edit*. Select *Data model* and enter **Unimodell**. Choose *Display* to go to the screen *Display Data Model: Definition* of the Data Modeler. This screen now contains the definition of the university model.

access the Data Modeler via the ABAP/4 Repository Information System

access the Data Modeler via the ABAP/4 Repository Information System

In the initial screen of the R/3 System, choose *Tools → ABAP/4 Workbench*.

In the screen *ABAP/4 Development Workbench*, choose *Overview → Repository Info Sys*. In the screen *ABAP/4 Repository Information System*, position the cursor on the node *Modeling* and expand the node with *Edit → Expand subtree*. Double-click with the mouse on the object class to be found. The relevant selection screen will be displayed.

The standard selection for the object class is defaulted. To obtain all selection options for this object class, choose *Edit → All selections*.

Enter your selection criteria and select *Program → Execute*. A hit list is generated.

If you double-click on the name of an object in the list, the definition screen for this object in the Data Modeler is displayed; Pressing *F3* returns you to the hit list. Alternatively, you can select the object you require and choose an editing type.

In the definition screen in the Data Modeler, you can change the editing mode (*Disp.<-> Chnge*) and can use the functions of the Data Modeler.



In the screen *ABAP/4 Repository Information System*, expand the *Modeling* node with *Edit → Expand subtree*. Double-clicking on *Data models* displays the screen *ABAP/4 Repository Information System: Data Models*. Enter **STFO** in the field *Development class* and click on *Execute*. A hit list is displayed. Double-clicking on *UNIMODELL* takes you to the relevant definition screen in the Data Modeler.

access the hierarchy of the SAP Model

1. Call the initial screen of the Data Modeler.
2. In the initial screen, select *Modeling object* → *SAP Application mod.* or *SAP Architecture mod.* or the corresponding pushbutton. You branch automatically to the hierarchy of the SAP models.

call the Repository Information System from the Data Modeler

call the Repository Information System from the Data Modeler

1. From the menu

You can call the ABAP/4 Repository Information System from the initial screen of the Data Modeler by selecting *Environment* → *Repository Info Sys*. The screen listing the areas of the ABAP/4 Repository Information System is displayed.

Most of the areas are preceded by a symbol. This indicates that subareas exist. To display these subareas, select the line you are interested in followed by *Edit* → *Expand subtree*. The subareas hidden beneath the nodes are displayed.

To allow searching for modeling objects, position the cursor on *Modeling* and select *Edit* → *Expand subtree*. Under the heading *Data modeling*, you will find all the points relating to the Data Modeler, namely *Data models*, *Entity types*, and *Entity type attributes*. When you double-click on one of these points, the corresponding selection screen appears.

2. From a modeling object

Leaving the entry field *Modeling object* empty, select the object class you require under *Selection* followed by the menu option *Find*. The corresponding selection screen (for entity types or data models) appears.



You can call the ABAP/4 Repository Information System from other points in the Data Modeler. You will be informed of this at the appropriate points of this documentation.

search for objects with the Repository Information System

Call the relevant selection screen in the ABAP/4 Repository Information System.

The standard selections for the object class in question are displayed. The maximum number of hits to be selected is also shown. These values are preset in your user settings.

If you wish to display all the objects from a particular object class, enter * in the first search field on the first line. Then select the menu options *Program* → *Execute*. A list corresponding to your selection will be output.

If you wish to search via fields that are not included in the standard selections, you can display all available selections for a particular object class. To do so, choose *All selections*. The additional selection options are now displayed underneath the standard selections.

To search for objects with particular attributes, you have to make the relevant entries in the search fields provided. The way in which these entries are analyzed is determined via the arrow pushbutton next to the input field.

You can search for single values or for ranges of values. If you want to search for an object with a specific attribute, it is sufficient to enter this attribute.



If you wish to search for all data models with names starting with U, it is sufficient to enter **U*** in the field *Data model*. The option *Pattn* is then selected automatically.

If you wish to search for all data models whose names start with a letter before U in the alphabet and after X, on the line *Data model* you must enter **U** in the first field and press the arrow pushbutton. A dialog box appears. In the second field, enter **X**. Position the cursor on each field and click on *Options*. Select the relevant search criteria and start the search.

define the user settings in the Repository Information System

define the user settings in the Repository Information System

Defining the user settings

To define your user settings, you have to call the initial screen of the ABAP/4 Repository Information System.

To do so, call the ABAP/4 Development Workbench from the R/3 initial screen with *Tools* → *ABAP/4 Workbench*.

In the ABAP/4 Development Workbench select *Overview* → *Repository Info Sys*.

Setting a start variant

The selection options available to you are determined by the choice of variant. A standard variant is normally set. You can change the variant by selecting *Settings* → *User parameters* in the initial screen of the ABAP/4 Repository Information System.

A dialog box containing a number of different variants appears. Select the variant you require and click on the *Save* pushbutton. The selected variant is now adopted as your start variant.

Specifying the maximum number of hits

The maximum number of hits determines the maximum number of objects matching the search criteria that is to be selected. For example, if 100 is entered as the maximum number of hits and you are searching for all tables with names beginning with U, only the first 100 tables matching this selection criterion will be selected. You can change the standard entry for the maximum number of hits.

To do so, select *Settings* → *User parameters* in the initial screen of the ABAP/4 Repository Information System. A dialog box appears in which you can enter the value you require. Click on *Save*. Your selection for the maximum number of hits is now adopted as your standard setting.

set the selection options in the Repository Information System

Press the arrow pushbutton beside the relevant field. A dialog box appears. Click on the *Options* pushbutton to maintain the selection options:

- *Single value*: All entries matching the entry value are selected.
- *Greater than or equal*: All entries greater than or equal to the entry value are selected.
- *Less than or equal*: All entries with values less than or equal to the entry value are selected.
- *Pattern*: All entries matching the pattern are selected. This option is displayed only when you have made a generic entry in the field, such as U* in the field *Data model*.
- *Exclude pattern*: All entries that do not match the pattern are selected. This option is displayed only if you have made a generic entry in the field, such as U* in the field *Data model*.
- *Not equal*: All entries not equal to the entry are selected.
- *Less than*: All entries with values that are less than the entry value are selected.
- *Greater than*: All entries with values greater than the entry value are selected.

use the Repository Information System to find out where objects are used

use the Repository Information System to find out where objects are used

Access the ABAP/4 Repository Information System. To do so, select *Tools* → *ABAP/4 Workbench* in the initial screen of the R/3 System. In the ABAP/4 Development Workbench, select *Overview* → *Repository Info Sys*.

To find out where modeling objects are used, position the cursor on *Modeling* and select the menu options *Edit* → *Expand subtree*. Under the heading *Data modeling*, you will find all the points relating to the Data Modeler, namely *Data models*, *Entity types*, and *Entity type attributes*. Click on *Data models* or *Entity types* and select *Repository Infosys*. → *Where-used list*.

A dialog box appears in which you have to enter the name of the object for which you wish to see a where-used list. In the case of entity types, you also have to specify which type of where-used list you require (i.e. for use in data models or in tables). Once you have done so, click on *Cont*. A hit list is displayed.

You can execute the following functions from within this list.

- You can access the display screen for an object by selecting the object you are interested in choosing *Display*.
- You can access the maintenance screen for an object from within the list by selecting the object in the first column of the display and selecting *Change*.
- You can display the occurrences of an object from the list in other objects by selecting the object and choosing *Utilities* → *Where-used list*.

define the user settings for the Data Modeler

To maintain the user settings, proceed as follows:

1. Select *Utilities* → *User settings* in any screen of the Data Modeler.



You **cannot** access the user settings from the graphics window.

2. Make your changes in the *User Settings* dialog box.
3. Select *Save* to save your changes.

print objects

print objects

Proceed as follows:

1. Call the definition screen for the object in question.
2. Select menu options *<Object type> → Print*.
3. The remaining procedure varies according to whether you started from the definition screen of a data model or entity type or of a subobject (such as attributes or relationships) or data model hierarchy:

Data model or entity type:

A dialog box appears in which you can specify which subobjects you wish to be printed along with the object.

For example, by selecting one of the options in this dialog box, you can choose whether you wish all or only selected components and/or assignments to be printed along with your entity types.

Confirm this first dialog box by selecting *Print*.

A second dialog box appears in which you can enter the name of the output device, the spool request, your output options, and whether you require a cover sheet.

If you select *Print preview* you can see what your printout will look like.

Confirm this second dialog window with *Print*.

Subobject or data model hierarchy

If you started from a data model hierarchy or from the definition screen of a subobject, a dialog box is now displayed in which you can enter the name of the output device, the spool request, your output options, and whether you require a cover sheet.

If you select *Print preview* you can see what your printout will look like.

Confirm this second dialog box with *Print*.

display the definition of an entity type

The procedure to be used varies according to whether you know the complete name of the entity type, part of the name or whether you do not know the name at all.

Name known

1. In the *Data Modeler: Initial Screen* enter the name of the entity type in the field *Modeling object*.
2. Select *Entity type* under *Selection*.
3. Choose *Display*. The screen *Display Entity Type: Definition* appears.

Part of name known

1. In the *Data Modeler: Initial Screen* select *Entity type* under *Selection*.
2. Enter the part of the name you know into the entry field *Modeling object*. Use * to represent the part you do not know (for example, **uni***).
3. Click on the arrow to the right of the entry field.
4. A hit list is displayed. Select the entity type you require by clicking on it and activate *Choose*. The name is now entered in the field on the initial screen.
5. Choose *Display*. The screen *Display Entity Type: Definition* appears.

Name unknown

1. Leave the input field *Modeling object* empty in the *Data Modeler: Initial Screen*.
2. Under *Selection* select *Entity type*.
3. Choose *Find*. The corresponding standard selection screen of the ABAP/4 Repository Information System is displayed.

Enter your selection criteria and choose *Program* → *Execute*.

A hit list is displayed. Select the entity type you require from this list and choose *Display*. The screen *Display Entity Type: Definition* appears.

create the definition of an entity type

create the definition of an entity type

1. In the *Data Modeler: Initial Screen* enter a (unique) name for the new entity type in the entry field *Modeling object*.

Naming conventions:

Entity type names can consist of letters, figures, and underscores in any given sequence. The first eight characters of an entity type name should not be the same as those of another entity type, since this would mean that the default view name *Z_<entity type name (eight characters)>* could not be used for Dictionary assignment,(within SAP, the default view name is *U_<entity type name (eight characters)>*).

2. Select *Entity type* under *Selection*.
3. Click on *Create*.
4. Specify the attributes for the entity type in the screen *Create Entity Type: Definition*.
 - Enter the short text. This is a required entry.



If you are not sure whether the entity which you want to create already exists in the system, before saving you can search for entity types with similar short texts using *Edit → Find similar ent..*

- Under *Category* specify which *Time unit* applies (pressing F4 displays a list of possible time units).
- Select the appropriate code under *Customizing*.
- Enter the definition of the entity type.

If you press ENTER when creating the definition, a link (*&IM<entity type ID>*) is transferred automatically to the entity type short text.

In display mode this link is replaced automatically by the short text.

In maintain mode the default settings for the links are displayed. With *Edit*

Short text on<->off, you can display or suppress display of the short texts.

After saving your entries, press *Editor* and the text editor is displayed. You can enter more detailed definitions here.

You cannot make any entries under *Assigned table/views*.

5. Save the entity type and make the appropriate entries in the dialog box *Maintain Object Catalog Entry* (under *Development class* or as *Local object*) and in the dialog box of the *Workbench Organizer*.

change the definition of an entity type

1. In the *Data Modeler: Initial Screen* enter the name of the entity type in the entry field *Modeling object*.
2. Select *Entity type* under *Selection*.
3. Click on *Change*.
4. Change the attributes as required in the screen *Maintain Entity Type: Definition*:
 - Short text
 - Time dependent and *Time unit*
 - Customizing
 - Definition

You cannot make any changes under *Assigned table/views*.
5. Save the entity type and make the appropriate entries in the dialog box of the Workbench Organizer.

delete the definition of an entity type

delete the definition of an entity type

1. In the *Data Modeler: Initial Screen* enter the name of the entity type in the entry field *Modeling object*.
2. Select *Entity type* under *Selection*.
3. Click on *Change*.
4. Choose *Entity type* → *Delete* in the screen *Maintain Entity Type: Definition*. A pop-up window appears.
5. In the *Delete entity type* pop-up, confirm that you wish to delete the entity type by selecting *Yes*. The remaining procedure varies according to whether or not references and/or dependent objects exist:

No references or dependent objects exist

If there are no references (occurrences in data models, relationships, and so on) or dependent objects (such as directly assigned attributes) for this entity type, it will now be deleted. The system returns you to the initial screen and a message stating that the entity type has been deleted is output in the status line.

References exist

If references still exist for this entity type, a list of these will be output. You cannot delete an entity type until you have deleted all its references.

The procedure is as follows:

- Double-click on the name of the first reference. The relevant screen in which you can delete the reference appears.
- Delete the reference.
- Click on *Back* to return to the list of references.
- Delete all the references.

Once all the references have been deleted, the entity type will itself be deleted, provided no dependent objects exist.

If dependent objects still exist, a list of these will be output.

Dependent objects exist

If the entity type to be deleted has dependent objects, a list of these will be output.

If you wish to delete the entity type with its dependent objects, choose *Delete* in this screen. The system returns you to the initial screen and a message stating that the entity type has been deleted is output in the status line.

branch from the definition of an entity type to the graphics function

branch from the definition of an entity type to the graphics function

To call the graphics from within the definition of an entity type, choose *Utilities* → *Graphics*.

The graphics window is generated and the entity type is displayed within the window in the display area.

assign a table or a view to an entity type**assign a table or a view to an entity type**

To assign a table or view to an entity type or to display or change the existing assignment, you must select *Goto* → *Dict. assignment* to access the screen *Table/View Assignment* after saving the entity type definition.

You can specify and check the assignment in this screen. Furthermore, you can access the definition of the table or view in the ABAP/4 Dictionary and, using the Data Browser, branch to any entities that may exist.

display the attributes of an entity type

In the screen *Display Entity Type: Definition* select *Goto* → *Attributes* to access the screen *Display Attributes*.

The attributes of the entity type are displayed here. Both attributes that were assigned directly and those taken over from the Dictionary are displayed. Attributes that were taken over from the ABAP/4 Dictionary to the entity type when a table or view was assigned are indicated by an entry in the column *Dic*.

In order to be able to see any attributes that may have been taken over from the generalization (inherited attributes), you have to select *Display options*. A dialog box appears in which you have to select *Inherited attributes* and click on the pushbutton *Continue*. All inherited attributes are indicated by an entry in the column *Inh*.

create the attributes of an entity type

create the attributes of an entity type

You can create only **directly** assigned attributes in the Data Modeler.

1. Choose *Goto* → *Attributes* in the screen *Change Entity Type: Definition*. The *Create Attributes* screen is displayed.

2. Enter a name for the attribute.

Naming conventions:

Attribute names may not be more than 10 characters in length. Only alphanumeric characters and underscores may be used.

3. Assign the attribute a category (*Cat*).

The attribute category is used to indicate the purpose of an entity type. Press *F4* to obtain a list of possible entries. An attribute can be assigned a maximum of two categories.

4. Assign a data element to the attribute:

Assign existing data element, name known

Enter the name of the data element.

Assign existing data element, name not known

Position the cursor on the empty data element field and click on the arrow that appears to the right of the entry field to access the ABAP/4 Repository Information System. The selection screen for data elements is displayed.

Enter your selection criteria in the selection screen and choose *Program* → *Execute*. Select the data element you require from the hit list and click on *Choose* to copy it to the *Create Attributes* screen.

Assign existing data element, part of name known

Enter the part of the name you know into the input field and use * to represent the part of the name you do not know. Click on the arrow to the right of the entry field. A hit list is displayed.

Using the cursor, select a data element and press `ENTER`.

Create and assign data element

Enter a name for the new data element. Then double-click on the name to access the ABAP/4 Dictionary. You can now create a new data element. You can read how to do this in the documentation for the *ABAP/4 Dictionary*. Once you have saved, checked, and activated your data element, click on *Back* to return to the *Create Attributes* screen.

5. Check the new attributes before saving.

change the attributes of an entity type

In the Data Modeler you can change the directly assigned attributes.

You can only change the category (*Cat*) of attributes that were transferred from the Dictionary or inherited (indicated by an entry in the column *Dic* and/or *Inh*).

1. Choose *Goto* → *Attributes* in the screen *Change Entity Type: Definition*. The *Maintain Attributes* screen appears.
2. Make your changes (see *Entity type: Creating attributes*).
3. Check the changed attributes before saving.
4. Click on *Back* to return to the entity type definition.

delete the attributes of an entity type**delete the attributes of an entity type**

You can delete only directly assigned attributes in the Data Modeler.

1. Choose *Goto* → *Attributes* in the screen *Change Entity Type*. The screen *Maintain Attributes* appears.
2. To delete individual attributes, position the cursor on the relevant line and choose *Edit* → *Delete line*.

To delete all directly assigned attributes, choose *Attributes* → *Delete*.

create tables and transfer attributes

In the Data Modeler you can create tables directly and transfer the attributes from the definition of the entity type:

1. In the screen *Display entity type: Definition* choose the option *Goto → Dictionary assignment*. The screen *Display Table/View Assignments* appears.
2. Use *Display <-> Change* to switch to maintenance mode.
3. In the *Goto* menu choose the option *Table/view*. The screen *ABAP/4 Dictionary: Table/structure: Change fields* is displayed. Fill all the mandatory fields.
4. In the *Extras* menu choose the option *Copy fields...* In the dialog box *Transfer entity fields* enter the ID of the entity type.
5. In the *Edit* menu choose the option *Insert*. The attributes of the entity type are inserted as field attributes.

display relationships

display relationships

Proceed as follows to display existing outgoing/ingoing relationships of an entity type:

1. Choose *Goto* → *Ingoing* (or *Outgoing*) *relationships* in the screen *Display Entity Type: Definition*. The screen *Display Ingoing* (or *Outgoing*) *Relationships* appears. A list of the ingoing (outgoing) relationships is displayed.
2. To access the definition of a relationship, double-click on the name of the source entity type (or target entity type) in the list. The screen *Display Relationship: Definition* appears.

create relationships

1. Choose *Goto* → *Ingoing* (or *Outgoing*) *relationships* in the screen *Change Entity Type: Definition*.

If you are in the screen *Display Entity Type: Definition*, click on *Display<->Change* to switch to maintain mode.

The screen *Maintain Ingoing* (or *Outgoing*) *Relationships* appears. A list of existing ingoing (outgoing) relationships is displayed.

2. Choose *Relationship* → *Create*. The dialog box *Create Relationship: Definition* appears.
3. Enter the name of the source entity type in the field *Entity type from* or of the target entity type in the field *Entity type to* in the dialog box.

If a relationship already exists between the two entity types, you should increment the number of the relationship role. Press **F4** on the field *Relationship role* to see the existing role numbers.

Choose *Continue*.

4. In the screen *Create Relationship: Definition* make the following entries under *Attributes*:

- Category
- Cardinality
- Temporary reference (only if category is referential and value for left side of the cardinality is C)

To display possible entries, position the cursor on the relevant entry field and click on the arrow that appears to the right of the field.

5. Enter the relationship texts and the corresponding definitions. The definition can also be generated automatically from the cardinality. For this purpose, it must be empty. To generate a definition in this way, press **ENTER**.
6. Save the definition of the relationship.

change relationships

change relationships

1. Choose *Goto* → *Ingoing* (or *Outgoing*) *relationships* in the screen *Change Entity Type: Definition*.

If you are in the screen *Display Entity Type: Definition*, click on *Display* ↔ *Change* to switch to maintain mode.

The screen *Maintain Ingoing* (or *Outgoing*) *Relationship* appears. A list of existing ingoing (outgoing) relationships is displayed.
2. Position the cursor on the relationship you wish to change and choose *Edit* → *Choose* or double-click on the relevant entity type. The screen *Maintain Relationship: Definition* appears.

If you change the cardinality, a pop-up window appears in which you are asked if the definition texts are to be regenerated.
3. Save your changes.
4. Click on *Back* to return to the screen *Maintain Ingoing* (or *Outgoing*) *Relationship*. If you click on *Back* again, the screen *Change Entity Type: Definition* appears.

delete relationships

1. Choose *Goto* → *Ingoing* or *Outgoing relatshps* in the screen *Change Entity Type: Definition*.
If you are in the screen *Display Entity Type: Definition*, click on *Display<->Change* to switch to maintain mode.
The screen *Maintain Ingoing (or Outgoing) Relationship* appears. A list of existing ingoing (outgoing) relationships is displayed.
2. Position the cursor on the relationship you wish to delete. If you double-click with the mouse, the screen *Maintain Relationship: Definition* appears.
3. Choose *Relationship* → *Delete* in the screen *Maintain Relationship: Definition*. A confirmation pop-up appears.
4. Confirm that you wish to delete the specified relationship with *Yes*. The system automatically displays the updated screen *Maintain Ingoing (or Outgoing) Relationship* and outputs a message in the status line indicating that the relationship has been deleted.

display generalizations**display generalizations**

1. Choose *Goto* → *Generalizations* in the screen *Display Entity Type: Definition*.
2. The screen *Display Generalizations* shows you the generalizations existing for this entity type.

You can access the definition of an entity type by double-clicking on its name. To access the definition of a generalization, position the cursor on a specialization category and choose *Goto* → *Generalization*.

create generalizations

You should first ensure that the entity type which is to be the generalization, and the corresponding specialization category for the generalization actually exist (see *Entity type: specializations/specialization categories*).

1. Choose *Goto* → *Generalizations* in the screen *Change Entity Type: Definition*.
2. Choose *Generalization* → *Create general...* in the screen *Maintain Generalizations*. A dialog box *Create Generalization* is displayed.
3. Enter the name of the generalization under *Entity type from* and the number of the specialization category under *Specializ. Cat.* Press *Continue*.
4. The *Change Generalization* screen is displayed. Save the generalization here.

delete generalizations**delete generalizations**

1. Choose *Goto → Generalization* in the screen *Change Entity Type: Definition*.
2. In the screen *Change Generalizations*, position the cursor on a specialization category and choose *Goto → Generalization*.
3. In the next screen you can delete the generalization with *Generalization → Delete*.

display specializations/specialization categories

1. Choose *Goto* → *Specializations* in the definition screen of the entity type whose specializations/ specialization categories you wish to see.
2. The screen *Display/Maintain Specializations* appears. The screen contains a list of all existing specialization categories along with the corresponding specializations.
3. To display the definition of a specialization category or entity type, double-click on the definition.

create specializations/specialization categories

create specializations/specialization categories

Step 1:

Before you can create specializations for an entity type, you must create the corresponding specialization category.

1. Choose *Goto* → *Specializations* in the definition screen of the entity type for which you wish to create specializations/specialization categories.
2. Choose *Create specialization* → *Create spec. cat.* in the screen *Change Specializations*. The dialog box *Create Specialization Category* appears.
3. Enter the number of the specialization category in the dialog box and press *Continue*.
4. Make the following entries in the screen *Create Specialization Category*:
 - *Short text* (mandatory)
 - Attribute: complete or disjoint.
 - *Definition*
5. Save your entries and return with *Back*.

Step 2:

You can now enter the specializations.

1. Position the cursor on the specialization category you have created and press *Create special.* The dialog box *Create Specialization* appears.
2. Enter the name of the specialization in the dialog box under *Entity type to* and press *Continue*.
3. Save the created specialization in the new screen and return to the definition screen with *Back*.

change the specialization category

1. Choose *Goto* → *Specializations* in the definition screen of the entity type for which you wish to change one or more specialization categories.
2. You can access the screen *Maintain Specialization Category* by double-clicking on the specialization category in the screen *Maintain Specialization*.
3. Make the changes in this screen and save them (see also *Creating specialization categories/specializations*).

delete specialization category/specializations**delete specialization category/specializations**

1. Choose *Goto* → *Specializations* in the definition screen of the entity type for which you wish to delete a specialization category.
2. You can access the screen *Maintain Specialization Category* by double-clicking on the specialization category to be deleted.
3. Choose *Specializ. category* → *Delete...*

The procedure now varies according to whether specializations still exist for this specialization category.

No specializations exist

The pop-up window *Delete Specialization Category* appears. Confirm your deletion request with *Yes*. The specialization category is deleted and a message informing you of this is output.

Specializations exist

The pop-up window *Delete Specialization Category* appears. Confirm that you wish to delete with *Yes*.

A list of existing specializations appears. Delete these specializations by double-clicking on each individual specialization.

The screen *Maintain Specialization* is displayed. Choose *Specialization* → *Delete*. A pop-up window appears in which you have to confirm your deletion request with *Yes*. The corresponding specialization is deleted.

Delete all the specializations in this way. Once all specializations have been deleted, the specialization category will also be deleted.

display the Dictionary assignment of an entity type

display the Dictionary assignment of an entity type

If there is an entry under *Assigned table/view* in the definition screen for an entity type, you can access the screen *Table/View Assignment* with *Goto* → *Dict. assignment*.

create or change the Dictionary assignment of an entity type

create or change the Dictionary assignment of an entity type

1. Choose *Goto* → *Dict. assignment* in the definition screen of the saved entity type (in maintain mode). The screen *Display (or Change) Table/View Assignment* appears.

The default view name (*Z_<Entity type name, eight characters>*) can be found in the input field for the table or view name if an assignment has not yet been made.
2. You can either assign a table or a view or change the existing assignment:
 - **Table assignment**

Select *Table* and delete the default view name (only if you are setting up a Dictionary assignment for the first time). Enter the corresponding table name:

Assign existing table, name known

Enter the name of the table.

Assign existing table, name not known

Position the cursor on the empty input field and click on the arrow to the right of the input field to access the ABAP/4 Repository Information System. The selection screen for tables appears.

Enter your selection criteria in the selection screen and choose *Execute*. Select the table you require from the hit list and confirm it with *Choose* to transfer it to the screen.

Create and assign new table

Enter the name of the new table. You can access the ABAP/4 Dictionary by double-clicking on the name. Create a new table there. See also the documentation for the *ABAP/4 Dictionary*. After saving, checking and activating the table, return to the Data Modeler by pressing the *Back* pushbutton.
 - **View assignment**

Select *View*. The default view name can be changed for example if you want to assign a view which already exists to the entity type.

Assign existing view, name known

Enter the name of the view.

Assign existing view, name not known

Position the cursor on the empty input field and click on the arrow to the right of the input field in order to access the ABAP/4 Repository Information System. You will access the selection screen for views.

Enter your selection criteria in the selection screen and choose *Execute*. Select the view you require from the hit list and confirm it with *Choose* to transfer it to the screen.

Create and assign new view

Enter the name of the new view. You can access the ABAP/4 Dictionary by double-clicking on the name. Create a new view there. See also the documentation for the

create or change the Dictionary assignment of an entity type

ABAP/4 Dictionary. After saving, checking and activating the view, return to the Data Modeler by pressing the *Back* pushbutton.

3. Check and save the table/view assignment.

delete the Dictionary assignment of an entity type**delete the Dictionary assignment of an entity type**

Please note that this function only deletes the assignment, not the assigned table or view.

1. Choose *Goto* → *Dict. assignment* in the definition screen of the entity type (in maintain mode). The screen *Maintain Tables/View Assignment* is displayed.
2. Choose *Assignment* → *Delete*. A pop-up window is displayed in which you are requested to confirm your deletion request.
3. Confirm your deletion request in the pop-up window by clicking on *Yes* if you are sure you wish to delete the assignment.
4. A second pop-up window informs you that the table or view cannot be deleted in the ABAP/4 Dictionary. Select *Continue*.

display, create or change a comment for an entity type

Access a comment for an entity type from the definition screen with *Goto* → *Comment*.

You can create, change or delete a comment here in maintain mode. If you click on the *Editor* pushbutton or double-click on the lines involved, the text editor is displayed. You can enter a more detailed comment here. Once you have created or changed a comment, save your work.

delete a comment for an entity type

delete a comment for an entity type

To delete a comment, choose *Comment* → *Delete* in the screen *Maintain Entity Type: Comment*. Confirm your deletion request in the pop-up window that appears by selecting *Yes*. You are then automatically returned to the definition screen.

display, create or change an example for an entity type

Access an example from the definition screen of an entity type with *Goto* → *Example*.

You can create, change or delete an example in maintain mode. If you click on the *Editor* pushbutton or double-click on the lines involved, the text editor is displayed. You can enter a more detailed example here. Once you have created or changed an example, save your work.

delete an example for an entity type

delete an example for an entity type

To delete an example, choose *Example* → *Delete* in the screen *Maintain Entity Type: Example*. Confirm your deletion request in the pop-up window that appears by selecting *Yes*. You are then automatically returned to the definition screen.

delete, create or change an alias for an entity type

A list of the aliases for a specific entity type can be accessed from the definition screen of the entity type in question with *Goto* → *Aliases*.

You can create, change or delete aliases here in maintain mode. Additional lines will be displayed if you press the *Insert line* pushbutton. Once you have created or changed the aliases, save your work.

delete an alias for an entity**delete an alias for an entity**

To delete a single alias, position the cursor on the alias in the screen *Maintain Entity Type: Aliases* and choose *Delete line*. The alias is deleted.

To delete all aliases for an entity type, choose *Aliases* → *Delete* in the screen *Maintain Entity Type: Aliases*. Confirm your deletion request in the pop-up window that appears with *Yes*. All the aliases are deleted and you are automatically returned to the definition screen.

display the contents of the table/view which is assigned to an entity type

display the contents of the table/view which is assigned to an entity type

1. Starting from the definition screen of the entity type, choose *Utilities* → *Data Browser*. The selection screen of the Data Browser for the assigned table or view is displayed.
2. Enter the selection criteria and choose *Program* → *Execute*. A list of the entities found is displayed.

view the hypertext of an entity type

view the hypertext of an entity type

From the definition screen of an entity type, you can access the corresponding hypertext with *Utilities* → *Hypertext*.

You can access the other texts by clicking on the defined links. If you click on a highlighted term in the definition, the definition of the entity type with this short text is displayed.

If you click on *Exit w/o saving*, you are returned to the definition screen for the source entity type.

If you accessed other entity type definitions by clicking, activating *Back* repeatedly takes you back one hypertext display at a time until you arrive at the definition screen for the source entity type.

Select *Cancel* or *Exit* to interrupt this sequence and to display the definition screen for the entity type whose hypertext definition you are just leaving.

create a variant for an entity type

1. After you have saved the entity type, select *Edit* → *Variants* → *New variant* to display the screen *Create Entity Type: Definition* again. The title for the definition is now *Definition (variant 01)*. The definition text which had already been saved is no longer visible.
2. Enter the variant for the definition. To access the text editor, click on the *Editor* pushbutton or double-click on the text lines involved. You can enter more detailed definitions here. To exit from the text editor, choose *Back*.
3. Save the variant with *Edit* → *Variants* → *Save variant*.

display or change a variant for an entity type**display or change a variant for an entity type**

1. Select *Edit* → *Variants* → *Get variant...* in the definition screen of the entity type.
2. The dialog box *Entity Type Variants* appears. This dialog box contains a list of all the variants of this entity type. Position the cursor on the variant you require and activate *Choose*.
3. The selected variant is displayed. If you are in maintain mode, you will be able to make changes (otherwise activate *Display<->Change*).
4. Select *Edit* → *Variants* → *Save variant* to save your changes.

delete a variant for an entity type

1. Access the variant you wish to delete as described above under *Display Variants* or *Maintain Variants*.
2. Choose *Edit* → *Variants* → *Delete variant*. A pop-up window appears asking you to confirm your deletion request.
3. Confirm your deletion request in the pop-up window by selecting *Yes*.

switch to another entity type

switch to another entity type

Proceed as follows:

1. Choose *Entity type* → *Other entity type...*
2. A dialog box appears for you to enter the name of the required entity type.

The procedure to be used varies according to whether you know the full name, only part of the name, or whether you do not know the name at all.

Name known

Enter the name in the entry field. Click on the *Continue* pushbutton to display the definition of this entity type.

Part of name known

Enter the part of the name which you know in the entry field and use * to represent the part you do not know (for example, **uni***). Click on the arrow to the right of the entry field.

The ABAP/4 Repository Information System is displayed. You can enter further selection criteria here. Select *Execute*.

A hit list of all the objects corresponding to the pattern you entered is displayed. To transfer the name of the entity type to the dialog box, position the cursor on the required entity type and press *Choose*. Press *Continue* again to access the definition of this entity type.

Name unknown

Without making an entry in the input field, click on the arrow to the right of the field. You branch to the ABAP/4 Repository Information System.

You can search for the entity type using the Information System. Position the cursor in the hit list on the name of entity type you are looking for and press the *Choose* pushbutton to transfer the name to the dialog box. Press the *Continue* pushbutton to access the definition of this entity type.

determine the assignment of an entity type to data models

1. Enter the name of the entity type for which you wish to display the assignment to data models in the *Modeling object* field of the *Data Modeler: Initial Screen*.
2. Select *Entity type* under *Selection*.
3. Press *Display* (or *Change*).
4. Select *Utilities* → *Inverse hierarchy* in the screen *Display (or Change) Entity Type: Definition*. The screen *Inverse Hierarchy: Entity Type <entity_type_name>* is displayed.

The entity type is displayed as a root from which you accessed the inverse hierarchy in the screen *Inverse Hierarchy: Entity Type <entity_type_name>*. The next level up is displayed below, that is, the data model(s) directly superior to the entity type. This means that the “higher” an object is in the hierarchy, the “lower” it will be displayed here.

To display the color key, choose *Utilities* → *Color key...*

You can change the display of the inverse hierarchy.



You can also call a general where-used list for modeling objects with *Utilities* → *Where-used list*.

change the display of the inverse hierarchy

change the display of the inverse hierarchy

Expanding a sub-tree

If you wish to display all the objects from the next level up in the hierarchy of a data model, proceed as follows:

1. Click on the name of the data model.
2. Choose *Edit* → *Expand sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Compressing a sub-tree

If you no longer wish to display the objects from the next level up in a data model, proceed as follows:

1. Click on the name of the data model.
2. Choose *Edit* → *Compress sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Setting the focus

If you wish another data model to become the root of the display, proceed as follows:

1. Click on the name of the data model you require.
2. Choose *Edit* → *Set focus*.

Above the new display, the path from the root of the previous display to the new root is shown. You can turn it into the root of the display by double-clicking on one of the path definitions.

print entity types and their subobjects)

Proceed as follows:

1. Access the definition screen for the object involved (entity type or subobject).
2. Select the menu options <Object type> → *Print*.
3. The remaining procedure varies according to whether you started from the definition screen of an entity type or a subobject (for example, attributes or relationships):

Entity type definition:

A dialog box is displayed in which you can specify which subobjects are to be printed.

By making selections in this dialog box, you can specify whether all or only selected components and/or assignments are to be output.

Select *Print* to confirm this first dialog box.

A second dialog box appears in which you can enter the name of the output device, the spool request, your output options, and whether you require a cover sheet.

If you select *Print preview* you can see what your printout will look like.

Confirm this second dialog box with *Print*.

Subobject definition

If you started from the definition screen of a subobject, a dialog box is now displayed in which you can enter the name of the output device, the spool request, your output options, and whether you require a cover sheet.

If you select *Print preview* you can see what your printout will look like.

Confirm this dialog box with *Print*.

display and print data models

display and print data models

Displaying data models

The procedure to be used varies according to whether you know the full name of the data model, only part of the name of the data model, or whether you do not know the name at all:

Name known

1. Enter the name of the data model in the field *Modeling object* in the *Data Modeler: Initial Screen*.
2. Select *Data model* under *Selection*.
3. Click on *Display*. The screen *Display Data Model: Definition* appears.

Part of name known

1. Select *Data model* under *Selection* in the *Data Modeler: Initial Screen*.
2. Enter the part of the name you know in the field *Modeling object* and use * to represent the part you do not know (for example, **uni***).
3. Click on the arrow to the right of the entry field.
4. A hit list of all objects corresponding to the pattern you entered appears. To copy the name of the data model you require, select the data model by clicking on it and activating the *Choose* pushbutton. The name is now entered in the initial screen.
5. Select *Display*. The screen *Display Data Model: Definition* is displayed.

Name unknown

1. Leave the input field *Modeling object* empty in the *Data Modeler: Initial Screen*.
2. Select *Data model* under *Selection*.
3. Select *Find*. The relevant standard selection screen of the ABAP/4 Repository Information System appears.

Enter your selection criteria and select *Program* → *Execute*.

A hit list is displayed. Select the data model you require and click on *Display*. The screen *Display Data Model: Definition* appears.

Printing data models

To print the data for a data model, select *Data model* → *Print*.

create a data model

Proceed as follows to create a new data model:

1. In the *Data Modeler: Initial Screen* enter a (unique) name for the new data model in the field *Modeling object*.

Naming conventions:

Data model names may only contain alphanumeric characters and underscores.

2. Select *Data model* under *Selection*.
3. Click on *Create*.
4. Specify the attributes for the data model in the screen *Create Data Model: Definition*:

- *Short text* (mandatory)
- *Category*
- *Graphics information*:

You can define here the color you wish your data model to have in the graphics. A list of available colors can be displayed with F4.

- *References*
- *Definition*:

You can access the text editor by clicking on *Editor* or by double-clicking on the definition text. You can enter more detailed definitions here.

5. Save the data model and make the relevant entries in the dialog box *Maintain Object Catalog Entry* (under *Development class* or as *Local object*) and in the dialog box of the *Workbench Organizer*.

If you wish to insert the data model into the hierarchy and/or wish to assign the relevant entity types/data models, you must access the hierarchy.



To assign the new data model to another data model, you have to access the corresponding data model of the higher level. You can access the hierarchy from the definition of the higher-level data model and insert the new data model in this model.

Select *Hierarchy* to access the screen *Change Data Model: Hierarchy*.

change a data model

change a data model

Proceed as follows to change a data model:

1. Enter the name of the data model to be changed in the field *Modeling object* in the *Data Modeler: Initial Screen*.
2. Select *Data model* under *Selection*.
3. Click on *Change*.
4. Change the attributes in the screen *Change Data Model: Definition*:
 - *Short text* (mandatory)
 - *Category*
 - *Graphics information*:

You can define here the color you wish your data model to have in the graphics. A list of available colors can be displayed with F4 .
 - *References*
 - *Definition*:

You can access the text editor by clicking on *Editor* or by double-clicking on the definition text. You can enter more detailed definitions here.
5. Save your data model.

To make changes to the hierarchy, you must access the hierarchy.

Select *Hierarchy* to access the screen *Change Data Model: Hierarchy*.

delete a data model

Proceed as follows to delete a data model:

1. Enter the name of the data model to be deleted in the field *Modeling object* in the *Data Modeler: Initial Screen*.
2. Select *Data model* under *Selection*.
3. Click on *Change*.
4. Select *Data model* → *Delete* in the screen *Change Data Model: Definition*.
5. In the pop-up window that appears, confirm that you wish to delete the data model.
6. A list is displayed of any assignments (that is, objects assigned to the model) and/or references (that is, the data model is itself assigned to one or more data models) that may exist for the data model. The remaining procedure varies according to whether assignments and/or references exist:

Only assignments

If only assignments exist, click on the *Delete* pushbutton in the screen *Change Data Model: Definition* if you want to delete the data model and its assignments. The system returns you to the input screen and issues an appropriate success message.

If only assignments exist, click on the *Delete* pushbutton in the screen *Change Data Model: Definition* if you want to delete the data model and its assignments. The system returns you to the input screen and issues an appropriate success message.

Only references

If only references exist for the data model, these must all be deleted first.

To delete a reference, double-click on the name of the superordinate data model in the list. The screen *Change Data Model: Hierarchy* appears. In this screen you can remove the data model to be deleted from the hierarchy. Position the cursor on the data model to be deleted and select *Modeling object* → *Delete assignment*. Return to the list with *Back*.

Once the last reference has been deleted, the data model itself is deleted. You return to the screen you started from.

Assignments and references

First, only the references which you have to delete are displayed (see **References exist**).

When you return to the list, the assignments are also displayed. These will be deleted when you delete the data model. Click on *Delete* if you wish to delete the data model and its assignments.



When deleting references, you might find that a superordinate data model is locked by another user. If so, you will need to contact this user and link your tasks by means of a common change request. (For more details, please read the *Workbench Organizer* documentation.)

delete a data model

switch to another data model

From the screen *Create/Change Data Model: Definition*, proceed as follows to display or change a different data model:

1. Choose *Data model* → *Other model...*
2. A dialog box appears for you to enter the name of the data model you require.

The procedure to be used varies according to whether you know the full name, only part of the name, or whether you do not know the name at all:

Name known

Enter the name in the entry field. Click on the *Continue* pushbutton to display the definition of the data model.

Part of name known

Enter the part of the name which you know in the entry field and use the * to represent the part you do not know (for example, **uni***). Click on the arrow to the right of the entry field.

The ABAP/4 Repository Information System is displayed. Enter any further selection criteria you wish to specify and click on *Execute*.

A hit list of all the objects corresponding to the pattern you entered is displayed. To transfer the name of the data model to the dialog box, position the cursor on the required data model and press *Choose*. Press *Continue* to access the definition of this entity type.

Name unknown

Ensure that the input field is empty, click on the arrow to the right of the field. You branch to the ABAP/4 Repository Information System. You can search for the data model here.

Position the cursor in the hit list on the name of the data model you are looking for and press the *Choose* pushbutton to transfer the name to the dialog box. Press the *Continue* pushbutton to access the definition of this data model.

carry out consistency checks

carry out consistency checks

To display the screen *Check Consistency of Data Models*, select *Utilities* → *Consistency checks* in the data model definition or the hierarchy.

To carry out consistency checks, proceed as follows:

1. Define the data models to be checked.

You can check a single data model or a set of data models. The name of the data model from which you called the screen *Check Consistency of Data Models* is defaulted as the first value in the selection line *Data model*. Just as in the ABAP/4 Repository Information System, you can select a certain set of data models here.

You also have the option of selecting the models to be checked from the data model hierarchy and proceeding with menu options *Utilities* → *Consistency checks*. In the screen *Check Consistency of Data Models* you can display the list of data models previously selected by clicking on the arrow pushbutton => and make changes if required.

2. Specify whether the set of data models to be checked contains only *Application models*, only *Business objects* or both.
3. Select the checks you require. You can select any combination of the four checks:
 - *All checks*: All four checks are made.
 - Check: completeness
 - Check: existence of predecessors
 - Check: connectivity
 - Check: existence of hierarchy_obj.
4. Select *Program* → *Execute* or *Program* → *Execute + Print* to make the consistency checks and obtain the check log.

If you choose *Execute*, the check log will be output only to the screen. If you select *Execute + Print*, you can specify a printer to which you wish to send a printout of the log.



The check log could be very long, depending on the number of data models checked.

find out in which other data models a data model is used

find out in which other data models a data model is used

Proceed as follows:

1. Select *Data model* under *Object selection*.
2. Press *Display* (or *Change*).
3. Select *Utilities* → *Inverse hierarchy* in the screen *Display (or Change) Data Model: Definition*. The screen *Inverse Hierarchy: Data Model <data_model_name>* appears.

The data model is displayed as a root from which you accessed the inverse hierarchy in the screen *Inverse Hierarchy: Data Model <data_model_name>*. The next level up is displayed below, that is, the data model(s) directly superior to the source data model. This means that the “higher” an object is in the hierarchy, the “lower” it will be displayed here.

To display the color key, choose *Utilities* → *Color key...*

change the display of the inverse hierarchy

change the display of the inverse hierarchy

Expanding a sub-tree

If you wish to display all the objects from the next level up in the hierarchy of a data model, proceed as follows:

1. Click on the name of the data model.
2. Choose *Edit* → *Expand sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Compressing a sub-tree

If you no longer wish to display the objects from the next level up in a data model, proceed as follows:

1. Click on the name of the data model.
2. Choose *Edit* → *Compress sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Setting the focus

If you wish another data model to become the root of the display, proceed as follows:

1. Click on the name of the data model you require.
2. Choose *Edit* → *Set focus*.

Above the new display, the path from the root of the previous display to the new root is shown. You can turn it into the root of the display by double-clicking on one of the path definitions.

access the data model

From the initial screen of the Data Modeler

Hierarchy of SAP models

You can call the hierarchy of the SAP models directly from the initial screen of the Data Modeler. Since this hierarchy contains all application models supplied by SAP, you can quickly gain an overview of all the models or navigate to a specific model.

To access the hierarchy:

1. Call the initial screen of the Data Modeler.
2. Select menu options *Modeling object* → *SAP Application mod.* or *SAP Architecture mod.*, or click on the corresponding pushbutton. The hierarchy of all SAP models is displayed.

Accessing the hierarchy from a specific data model

The procedure to be used varies according to whether you know the full name of the data model from which you wish to access the data model hierarchy, only part of the name, or whether you do not know the name at all.

Complete name known

1. Enter the name of the data model in the field *Modeling object* in the *Data Modeler: Initial Screen*.
2. Select *Modeling object* → *Data mod. hierarchy*. The screen *Display Data Model: Hierarchy* is displayed.

Part of name known

1. Select *Data model hierarchy* under *Selection* in the *Data Modeler: Initial Screen*.
2. Enter the part of the name that you know in the field *Modeling object* and use * to represent the part you do not know (for example, **uni***).
3. Click on the arrow to the right of the entry field.
4. A hit list of all objects corresponding to the pattern you entered appears from which you can select the required data model. Press *Choose*. The name now appears in the initial screen.
5. Select *Modeling object* → *Data model hierarchy*. The screen *Display Data Model: Hierarchy* is displayed.

Name unknown

1. Leave the input field *Modeling object* empty in the *Data Modeler: Initial Screen*.
2. Select *Find*. The relevant standard selection screen of the ABAP/4 Repository Information System is displayed.

Selecting a specific data model

Enter your selection criteria and select *Program* → *Execute*. A hit list is displayed. Select the data model you require from the hit list and click on *Display*. The screen *Display Data Model: Definition* is displayed.

access the data model**List of all existing data models**

Make **no** entry in the line *Data model* in the selection screen.

Under *Settings* set the *Max. no. of hits* to 9999 and select *Program* → *Execute*. A hit list is displayed. Select the data model you require from the list and activate *Display*. The *Display Data Model: Definition* screen appears.

3. Select *Goto* → *Hierarchy*. The *Display Data Model: Hierarchy* screen appears.

From the data model definition

From the definition of a data model, select *Goto* → *Hierarchy* to access the data model hierarchy.

The data model you selected or from whose definition you accessed the hierarchy is now displayed as the root along with its direct successors in the screen *Display Data Model: Hierarchy* or *Change Data Model: Hierarchy*.

You can toggle between display and maintain modes with *Display* <-> *Change*.

To display the color key, select *Utilities* → *Color key*.

print the data model hierarchy

To print a data model hierarchy, select *Modeling object* → *Print*.

change the display of the data model hierarchy

change the display of the data model hierarchy

Hierarchy: refreshing the display

With *Edit* → *Refresh*, you can refresh the display to show any changes made to the hierarchy caused by changes to the graphic by other users or by your own Data Modeler calls. The new display corresponds to the current state in the database.

Hierarchy: expanding a sub-tree

If you wish to see any objects located under a submodel (that is, objects assigned to the data model), proceed as follows:

1. Position the cursor on the name of the data model.
2. Select *Edit* → *Expand sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Hierarchy: compressing a sub-tree

If you do **not** wish to see the objects located under a submodel (that is, objects assigned to the data model), proceed as follows:

1. Position the cursor on the name of the data model.
2. Select *Edit* → *Compress sub-tree*.

Alternatively, you can click on the symbol to the left of the data model name.

Hierarchy: setting the focus

If you wish another data model to become the root of the display, so that you can display or edit the objects located underneath, proceed as follows:

1. Position the cursor on the data model which is to become the new root of the display.
2. Select *Edit* → *Set focus*.

Above the new display, the path from the root of the previous display is shown.

You can turn it into the root of the display by double-clicking on one of the path definitions.

insert objects in the data model hierarchy

To insert existing or new objects **on a lower level** or **on the same level** of an existing data model hierarchy, proceed as follows:

1. Position the cursor on the object to which another object is to be assigned on a lower level or on the same level of the hierarchy.



The cursor position determines where the object is inserted. Objects are not resorted automatically when new objects are inserted.

2. Select *Modeling object* → *Insert* → *On same level...* or *Subordinate...*



No other object can be assigned on the same level as a root and no other objects can be assigned subordinate to an entity type.

An object can be assigned to different data models of the hierarchy, but cannot be assigned to the same data model twice.

3. A dialog box appears in which you have to specify whether the object to be inserted is a data model or an entity type.
4. Enter the name of the object to be inserted in the next dialog box. The procedure to be used varies according to whether the object already exists and whether you know its full name:

Object exists, name known

Enter the name and select *Continue*. The entry for the object is displayed in the hierarchy.

Object exists, part of name known

Enter the part of the name you know in the entry field and use * to represent the part you do not know (for example, **uni***). Click on the arrow to the right of the entry field. A hit list of all objects corresponding to the pattern you entered appears. Select the object you require and choose *Transfer*. The entry for the object is now displayed in the hierarchy.

Object exists, name unknown

Select *Find* in this pop-up window to access the ABAP/4 Repository Information System. You can search here for the object you require.

Select the name of the object you require from the hit list and click on *Continue*. The entry for the object is now displayed in the hierarchy.

Object to be created

If the object to be inserted does not yet exist, a further pop-up window will appear. If you wish to create the object, confirm with *Yes*. The screen for creating objects will be displayed automatically. For information on how to proceed, refer to *Create data model* or *Entity Type: Definition*. Once the object has been created, return to the hierarchy with **F3**. The entry for this object is now displayed.

insert objects in the data model hierarchy

You need not save any changes to the hierarchy because the system automatically saves them.

delete assignments in the data model hierarchy

Proceed as follows to remove one or more objects from the hierarchy:

1. Select the objects to be removed by positioning the cursor on each of the objects and activating *Select/deselect node* (for entity types) or *Sel./desel. sub-tree* (for data models).
2. Select *Modeling object* → *Delete assignment*.

The objects are automatically removed from the hierarchy.



You cannot remove an object from the hierarchy without also removing **all** its subordinate objects.

Delete assignment only removes objects from the data model hierarchy in question.

reassign objects in the data model hierarchy

reassign objects in the data model hierarchy

Proceed as follows to reassign modeling objects:

1. Position the cursor on the modeling object to be reassigned, that is, on the name of the data model (= sub-tree to be reassigned along with all its subordinate objects) or on the entity type you wish to reassign. Activate *Sel./desel. sub-tree* for data models or *Select/deselect node* for entity types.

If you wish to reassign individual entity types and/or submodels, select the objects one after another with *Select/deselect node* or *Sel./desel. sub-tree*.

2. Position the cursor on the object on the same or lower level of the hierarchy to which the objects are now to be assigned.
3. Select *Modeling object* → *Reassign....* A dialog box appears.
4. In the dialog box, specify whether the object to be reassigned is to be on the same or a lower level of the hierarchy.
5. Activate *Reassign....*



No other object can be assigned on the same level as a root and no other objects can be assigned subordinate to an entity type.

switch to the hierarchy of another data model

If you wish to display or edit the data model hierarchy of another data model, you can switch to this data model as follows:

1. Select *Modeling object* → *Other model...* A dialog box appears.
2. The procedure to be used varies according to whether you know the full name of the data model, only part of the name, or whether you do not know the name at all:

Complete name known

Enter the name of the data model you require in the dialog box.

Part of name known

Enter the part of the name you know in the entry field and use * to represent the part you do not know (for example, uni*). Click on the arrow to the right of the entry field. A hit list of all objects corresponding to the pattern you entered is displayed.

Position the cursor on the model you require and activate *Transfer*. The name of the selected data model now appears in the entry field.

Name unknown

You can display a list of all existing data models. Enter * in the first search field of the line *Data model* in the ABAP/4 Repository Information System.

Alternatively, you can search the ABAP/4 Repository Information System for a specific data model by entering selection options.

Position the cursor on the model you require in the hit list and click on *Transfer*. The name of the selected data model now appears in the input field.

3. Select *Continue* to display the hierarchy of the data model you require.

carry out consistency checks from the data model hierarchy

carry out consistency checks from the data model hierarchy

To display the screen *Check Consistency of Data Models* from the data model definition or from the hierarchy, select *Utilities* → *Consistency checks*.

To carry out consistency checks, proceed as follows:

1. Define the data models to be checked.

You can check either a single data model or a set of data models. The name of the data model from which you called the screen *Check Consistency of Data Models* is defaulted as the first value in the selection line *Data model*. Just as in the ABAP/4 Repository Information System, you can select a certain set of data models here.

2. Specify whether the set of data models to be checked contains only *Application models* and/or *Business objects* or both.
3. Select the checks you require. You can select any combination of the four checks:
 - *All Checks*: All four checks are made.
 - Check: completeness
 - Check: existence of predecessors
 - Check: connectivity
 - Check: existence of hierarchy obj.
4. Select *Program* → *Execute* or *Program* → *Exec. + print* to make the consistency checks and to obtain the check log.

If you choose *Execute*, the check log will be output only to the screen. If you choose *Exec. + print*, you can specify a printer to which you wish to send a printout of the log.



The check log could be very long, depending on the number of data models checked.

switch to the graphic function from the data model hierarchy

switch to the graphic function from the data model hierarchy

1. If you want all objects of the hierarchy to be displayed in the graphic, you must select all these objects. To do so, select the root of the hierarchy with *Sel./desel. sub-tree*.

If you wish to display a number of different objects of the hierarchy in the graphic, for example, only submodel *Persmodel* and entity type *Uni1400000* (remuneration class) from data model *Unimodel*, position the cursor on the each of the required objects in turn and activate *Sel./desel. sub-tree* (for data models, in this case *Persmodell*) or *Select/deselect node* (for entity types, in this case *Uni1400000*).

2. Then select *Utilities* → *Graphics*.



In the case of larger data models, such as FI, it could take a considerable time to generate the graphics.

switch to the graphics positions of modeling objects from the data model hierarchy

switch to the graphics positions of modeling objects from the data model hierarchy

To display or change the graphics positions of hierarchy objects, proceed as follows:

1. Position the cursor on the predecessor of the object(s) whose graphics positions are to be displayed or maintained.
2. Select *Goto* → *Display options*. You can toggle between display and change modes with *Display* <-> *Change*.
3. Enter the new values for those objects whose positions are to be maintained under *Line* and *Column*.
4. Save your entries.

call the graphics function

From within the Data Modeler

From the Data Modeler you can access the graphics as follows:

- From the data model definition

In the *Data Modeler: Initial Screen* select the modeling object *Data model*, enter the name of a data model and choose either *Display* or *Change*. In the definition screen for the data model, select *Utilities* → *Graphics*. Wait until the graphic has been fully generated. The data model is then displayed graphically (along with all submodels).

If you called the graphics in maintain mode, the graphic will also be displayed in maintain mode.
- From the data model hierarchy

In the *Data Modeler: Initial Screen* select the modeling object *Data model hierarchy*, enter the name of a data model and choose either *Display* or *Change*. The hierarchy list is displayed. From this list select all data models and entity types you wish to be included in the graphic.

Use the function *Sel./desel. sub-tree* on data models in the hierarchy list to select all dependent objects for loading into the graphic. If you wish individual objects from a selected sub-tree to be omitted from the graphic, you can deselect these objects with *Select/deselect node*.

To start the graphics, select *Utilities* → *Graphics* in the hierarchy list. Wait until the graphic has been fully generated. All the data models and entity types in the hierarchy list that were selected are displayed in the form of a graphic.

If you called the graphics in maintain mode, the graphic will also be displayed in maintain mode.
- From the entity type definition

In the screen *Data Modeler: Initial Screen*, select *Entity type*. Enter the name of an entity type and choose *Display* or *Change*. The definition of the entity type appears.

Choose *Utilities* → *Graphics* to branch to the graphics. The entity type is then displayed in the display area of the graphics window.

From the Object Browser of the ABAP/4 Development Workbench

In the initial screen of the R/3 System, select *Tools* → *ABAP/4 Development Workbench*.

From the Object Browser of the ABAP/4 Development Workbench you can access the definition screen for a data model or the hierarchy list. From there, you can call the graphical display with the *Graphics* function. When you access the graphics from the Object Browser, they are displayed initially in display mode.

From the ABAP/4 Repository Information System

In the initial screen of the R/3 System select *Tools* → *ABAP/4 Development Workbench*.

In the ABAP/4 Development Workbench select *Overview* → *Repository Infosys*.

call the graphics function

Using the ABAP/4 Repository Information System you can search for data models or entity types. A list of data models or entity types is displayed. Select the data models or entity types you require from the list followed by the function *Utilities* → *Graphics*. Wait until the graphic has been fully generated. The selected data models or entity types are displayed in the graphic.

change the section of the screen displayed in the graphics function

Zoom function

Choose *View* → *Zoom in* to enlarge the objects displayed on the screen.

Choose *View* → *Zoom out* to reduce the size of the objects displayed on the screen.

Choose *View* → *Complete view* to zoom out until the complete graphic is displayed. All objects in the graphic are now visible.

You can also change the value entered in the zoom field in the toolbar using the arrows to the right of the zoom field. You can also simply overwrite the value displayed with another value and confirm the new value with `ENTER`.

Selecting a section

If you press the Control key and hold down the left mouse button, you can draw a frame around the section of the graphic you require in the graphics display area. When you release the mouse button, this section will be displayed in the display area.

Centering on selected objects

Select one or more entity types and select *View* → *Center on sel. obj.* The section containing the selected entity types is then shown in the display area.

Find

Select *Edit* → *Find* to display a dialog box containing a list of search texts. If you click on one of the texts, the displayed section of the graphic is shifted so that the relevant entity type is located in the section.

You can also enter a search pattern in the dialog box. Confirm your entry by pressing `ENTER` so that the list in the dialog box is updated to correspond to the search pattern.

select objects in the graphics function

select objects in the graphics function

Determining the selection type

You can specify the selection type with *Settings* → *Select (type)*: *with border, inverted, pick-marked, with color*.

Selecting

Selecting/deselecting within a graphic

Aim	Procedure
To select 1 object	Click on the object with the left mouse button.
To select/deselect several objects	Holding the SHIFT button pressed down, click on the objects one after another with the left mouse button.
To select all objects in the graphic	Select <i>Edit</i> → <i>Select</i> → <i>Select all</i> .
To select the environment, i.e. the direct predecessors and successors of an object (linked via relationship or generalization/specialization) and the object itself are to be selected	First select the object or objects whose environment is to be selected. Then select <i>Edit</i> → <i>Select</i> → <i>Select environment</i> .
To select the complement (the complement set of the previously selected entity types is selected)	First select the objects whose complement is to be selected. Then select <i>Edit</i> → <i>Select</i> → <i>Select complement</i> .
To deselect all objects	Select <i>Edit</i> → <i>Select</i> → <i>Deselect all</i> . Alternatively, you can click on a 'free area' outside all the models.

display/change objects in the graphics function

Displaying definitions of individual objects

If you double-click on an object, the definition screen for the object is displayed. The screen is automatically positioned in the foreground. At this point, the graphic is not ready for input. Select *Back*, *Exit* or *Cancel* to return to the graphic.

When you double-click on an object, the definition screen is always called in display mode.

You can display the definitions for the following objects by double-clicking:

- data model (colored-in area)
- entity type (white rectangle)
- relationship (narrow black line)
- specialization category (blue triangle)

Displaying/changing definitions/object components

The function *Edit* → *Change* is active only in the maintain mode of the graphics program. Select an object and choose this function. Just as when you double-click on an object, the definition screen for the object is now displayed. However, this time the definition is called in maintain mode.

Selecting *Goto* → *Entity type*, *Relationship*, *Specializ. category* or *Data model* also takes you into the definition screens for the objects. For entity types you can also call the attributes, Dictionary assignments, comment, example, and aliases. Select the object you require and choose a function. The corresponding screen is displayed automatically in the foreground. At this point, the graphic is not ready for input. You can now make your changes. (If you are still in display mode, you will need to select *Display*↔*Change* first.) Once you have saved your changes, you can return to the graphic with *Back*, *Exit* or *Cancel*. Changes relevant to the graphic, such as the entity type short text or data model color are updated when you return to the graphic.



The graphic is only updated if the change is made in the screen called directly from the graphic. Changes made several steps away from the graphic will not cause the graphic to be updated.

Editing several object definitions in sequence

The graphics program allows mass editing of object definitions. This is done by selecting several objects of the same class (entity types, relationships, specialization categories, data models) and choosing the corresponding function under *Goto* (*Entity type* → *Definition* or alternatively *Relationship*, *Specialization category* or *Data model*). The definition screen for one of the selected objects appears. In this screen, select *Goto* → *Next* <Object>. In this way you can 'scroll through' the definitions of all selected objects and, if required, edit them in maintain mode. Once all selected objects have been processed, selecting *Goto* → *Next* <Object> returns you to the graphic.

display/change objects in the graphics function**Entity type: displaying hypertext**

If you select an entity type followed by the menu options *Goto* → *Entity type* → *Hypertext*, the hypertext for this entity type is displayed. The hypertext contains the definition of the entity type and links to any other texts for this entity type (such as a comment). From the hypertext you can jump directly to the other texts.

Entity type: Data Browser (data display)

If you select an entity type followed by the menu options *Goto* → *Entity type* → *Data Browser*, you can display the data (entities) for the entity type that was taken from the assigned table or view. You branch to the *Data Browser* screen for the relevant table or view. In this screen you can enter the selection criteria and display the selected data.

create objects in the graphics function

Creating an entity type and/or inserting it into a data model

1. Click with the left mouse button on the entity type in the insertion area (in the lower right-hand corner of the screen). A phantom frame appears around the mouse pointer.
2. Position this phantom frame in the display area on the data model to which you wish to add an entity type.
3. A dialog box appears.

You now have two options:

 - a) **You can create a new entity type and insert it into a data model:**

Enter a new entity type ID in the dialog box. A pop-up window appears.

If you wish to create a new entity type, confirm this by selecting *Yes*. The screen *Create Entity Type: Definition* appears.

Make the necessary entries and save the entity type.

Select *Back* to return to the graphic.
 - b) **You can insert an existing entity type into a data model:**

Enter the name of an existing entity type (input help) in the pop-up window and confirm your entry with *Cont*. You are returned to the graphic.
4. The entity type is now included in the graphic.

If the entity type was inserted into a data model, positioning mode will now be activated for this data model. The data model is displayed in the color *magenta* and all entity types of the data model can be moved. After inserting an entity type, you can move it directly to the desired position and save the positions in the graphic. (To exit from positioning mode, either select *Edit* → *Position cursor* → *Positioning on/off* or switch to display mode.)

Using this procedure you can insert more entity types into the same or other data models.

Creating a relationship

1. Choose *Edit* → *Connect*. The mouse pointer is transformed into a pencil symbol.
2. With the left mouse button held down, draw a line from the source entity type to the target entity type.
3. When you reach the target entity type, release the mouse button. The screen *Create Relationships: Definition* appears. Enter the necessary information and save the relationship.

After saving you are returned automatically to the graphic. The relationship you have created appears as a black line extending from the source to the target entity type.

Creating a specialization category

1. To begin with, select the entity type for which you wish to define a specialization category in the display area.

create objects in the graphics function

2. Click on the blue triangle in the insertion area. The mouse pointer is transformed into a phantom frame.
3. Position this phantom frame to the right of the selected entity type in the display area. A dialog box appears.
4. In this dialog box, enter the number of the specialization category.
5. The screen *Maintain Specialization Category* is displayed. Make the necessary entries and save your specialization category.
6. Click on *Back* to return to the graphic. The specialization category appears there as a blue triangle that is connected to the selected entity type by a broad blue line.

Creating specializations

1. To begin with, create the associated entity types (see *Creating an entity type and/or inserting an entity type into a data model*).
2. Specify the specialization category for the generalizing entity type (see above under *Creating a specialization category*).
3. Connect the specialization category to the associated entity type. To do so, select *Edit → Connect*. The mouse pointer is transformed into a pencil symbol.
4. With the left mouse button held down, draw a line from the specialization category (blue triangle) to the corresponding entity type. When you reach the entity type, release the mouse button.
5. The screen *Create Specialization* appears. Save the specialization.
6. You are returned automatically to the graphic. The specialization appears there as a broad blue line from the specialization category to the corresponding entity type.

delete objects in the graphics function

Deleting an entity type

1. Select the entity type.
2. Choose *Edit* → *Delete*. A pop-up window appears.
3. Confirm that you wish to delete with *Cont.*
4. If references still exist, they are displayed in a list.

If you double-click on a reference, the corresponding screen (definition or hierarchy) is displayed. You can delete the reference here.

Return to the list with *Back* and repeat the procedure until all the references are deleted.
5. If dependent objects still exist, they are displayed in a list.
6. If you wish to delete the entity type and its dependent objects, click on *Delete*.

Deleting a relationship

1. Select the relationship.
2. Choose *Edit* → *Delete*. A pop-up window appears.
3. Confirm that you wish to delete with *Cont.* The relationship is deleted and disappears from the graphic.

Deleting a specialization category

If there are no specializations assigned to the specialization category, proceed as follows:

1. Select the specialization category.
2. Choose *Edit* → *Delete*. A pop-up window appears.
3. Confirm that you wish to delete with *Cont.*

Otherwise, the procedure is the same as that described for deleting references in *Deleting an entity type*.

Deleting a specialization

1. Select the specialization (the blue line between the specialization category and the corresponding entity type).
2. Choose *Edit* → *Delete*. A pop-up window appears.
3. Confirm that you wish to delete with *Cont.* The specialization is deleted and disappears from the graphic.

change the positions in the graphics

change the positions in the graphics

To maintain entity type positions, proceed as follows:

1. In the graphics maintain mode, select the data model whose entity types you wish to move.
2. Select *Edit* → *Position cursor* → *Positioning on/off*. The selected data model is now displayed in the color *magenta* and all entity types of the data model can be moved. (If the data model has not yet been assigned a task in the Workbench Organizer, a pop-up window informing you of this is displayed at this point.)
3. You are now in positioning mode and can move the entity types in the data model (with left mouse button held pressed down).
4. After you have finished positioning the entity types, you can save the entity type positions relative to the data model with *Edit* → *Position cursor* → *Save positions*.

You can display and change the numerical values for these positions in the hierarchy list for the data model.

5. By selecting *Positioning on/off* you can return to the maintain mode. The edited model is now returned to its original color.

use the graphics utilities Get environment, Suppress and Reload

Get environment

1. Select one or more entity types whose environment you wish to load.
2. Choose *Utilities* → *Get environment* and one of the functions *Predecessors*, *Successors* or *Predec. + successors*.
3. If no predecessors or successors are found, a message informing you of this is displayed.
If only one predecessors or successor is found, it is displayed directly on the graphic.
If several predecessors/successors are found, the screen *Get Environment: Choose Entity Types* is displayed. The screen contains a list from which you can select the predecessors/successors you wish to be included in the graphic.
4. Choose *Transfer* to include the selected predecessors/successors in the graphic. You are returned to the graphic. The predecessors/successors that were transferred are now displayed.

Suppress

The *Suppress* function enables you to remove entity types from the graphics display. (This function does not physically delete the entity types.)

1. Select all entity types you wish to suppress from display.
2. Choose *Utilities* → *Suppress*.

Reload

The function allows you to branch to the ABAP/4 Repository Information System to search for other data models or entity types, which you can then transfer to the graphic. When you do this, however, the previous contents of the graphic disappear from display.

1. Choose *Utilities* → *Reload* and *Data models* or *Entity types*. The corresponding selection screen of the ABAP/4 Repository Information System is displayed.
2. Enter your selection criteria and choose *Execute*.
3. A hit list is output. Select the objects you require and load them into the graphic with *Transfer*.

use the graphics utilities **Compress/Expand data model, Adjust, Visibility and Restrict area**

use the graphics utilities **Compress/Expand data model, Adjust, Visibility and Restrict area**

Compressing/expanding data models

1. Select one or more data models from the graphic.
2. Choose *Edit* → *Compress*.

Each selected data model is compressed to form a node, which has the same color as the data model and contains the data model short text. All ingoing and outgoing relationships are compressed to a single line each.

3. With *Edit* → *Expand* you can restore the original status of the selected data models.

Adjust

1. Choose *Edit* → *Adjust*. A dialog box appears.
2. You can specify here whether you wish to adjust only the selected nodes and links, or whether you wish to adjust the complete data model. After making your selection, choose *Cont*.

The automatic layout mechanism is activated and optimizes the layout of the graphic with respect to the positioning of the nodes and the paths taken by the links.

Visibility

You can use this function to make previously selected nodes (entity types, specialization categories) and links (relationships, specializations) invisible.

1. Select the objects you wish to make invisible.
2. Choose *Edit* → *Visibility* → *Hide*. The selected nodes and links are no longer displayed in the graphic.
3. To make the hidden nodes and links visible again, choose *Edit* → *Visibility* → *Show*.

The hidden objects must have been selected first. (If in doubt, start by selecting *Edit* → *Select* → *Select all*.)

Restricting the visible area

This function allows you to restrict the visible area to the selected entity types. Direct predecessors and successors are shown as black dots and are labeled with the relevant entity type ID.

1. Select the objects that are to be displayed.
2. Choose *Edit* → *Restrict area* → *On*. Now only the selected nodes and their connections are displayed.
3. Choose *Edit* → *Restrict area* → *Off*. The original status is restored.

change the settings in the graphics function

Header

Settings → *Header* → *Change...* calls a dialog box. Here you can enter a new text for the second header line. If you select *Transfer*, this text now appears in the graphic.

Change partitioning

You can specify here how the inner frames are to be partitioned in maintain mode.

1. Choose *Settings* → *Change partitioning*. A screen is displayed in which you can specify how the screen is to be partitioned.
2. Click on the partitioning you require and choose *Apply*.

Change sequence

You can specify here the sequence of the areas (display, navigation, and insertion area) to be used in maintain mode. Proceed as follows:

1. Choose *Settings* → *Change sequence*. A screen is displayed in which you can specify the sequence.
2. Click on the sequence you require and choose *Apply*.

Movability on/off

1. To make it possible to move entity types, select *Settings* → *Movability on/off*. A pop-up window is displayed confirming that this function has been activated.
2. To move an entity type, depress the left mouse button and drag the entity type to the desired position.

print the graphic**print the graphic**

1. Choose *Graphics* → *Print...* A dialog box appears in which you can enter your printer settings.
2. Enter the name of the *Destination printer*.
3. Specify the number of pages over which the graphic is to be distributed. Before you can do this, you must have selected the option *Fit to page(s)*. Select the number of pages by double-clicking on the appropriate number of cells in the raster.

If you wish to work with a default size, select the option *Default size*. The raster disappears and the system determines the number of pages.

4. Determine whether the *Cutting marks* are to be switched on or off.
5. Choose *OK*.

If, in place of *OK*, you select the option *Preview*, a print preview of the graphic is displayed in the pop-up window. You can then change your settings if necessary.