



Glossary

COPYRIGHT

1989 - 2000 by **ZUCCHETTI TOOLS** S.r.l.

All rights reserved.

This publication contains information protected by copyright. This publication may not be reproduced, stored, or transmitted, in any form or by any means without the prior permission of the publisher.

TRADEMARKS

All produced trademarks are ownership of the holder and are acknowledged by this publication.

ZUCCHETTI TOOLS S.r.l. SOFTWARE TECHNOLOGY

PADOVA - BELLARIA - RIMINI

E-mail: clabr@codelab.it

Web Address:

<http://www.zucchettitools.com>

<http://www.codepainter.com>

<http://www.codelab.it>

Summary

Introduction	1
1.1 Introduction	1
Design Glossary.....	3
2.1 Design Glossary Elements	3
Codify Glossary.....	13
3.1 Codify Glossary Elements	13
Front End Glossary	19
4.1 Front End Glossary Elements.....	19

Chapter 1

Introduction

1.1 Introduction

This Glossary has been divided in three chapters refering back to the three main **CODEPAINTER REVOLUTION** sections; namely Design, Codify and Front End. Its aim is explaining those terms that within the tool have specific meanings. Further, it has not been integrated within one of the guides, so that you can always easily refer to it, no matter if you are doing your project analysis, using CodePainter or simply using the software application.

Terms headings are **bold**. Within the text you can also find words in *italic*, meaning that those terms are included in the glossary.

Chapter 2

Design Glossary

2.1 Design Glossary Elements

Allow Nulls Option

The field value may be null. This option is not available in *Primary Keys*.

Alternate Key

Alternate keys are *Candidate Keys* that are not *Primary Keys*. An entity can have many *Candidate Keys*.

Attribute

An attribute is a noun or property that describes an entity. Attributes are often values, adjectives or dates. All data stored in an attribute must be of the same kind and having the same properties. After you have identified the main entities, you need to identify the attributes (read fields), i.e. the kind of information that must be stored for each entity.

Autonumber Button

It opens a dialog window where you can define automatically increasing numbers for field groups.

GLOSSARY

Business Object

A business application can be divided into three fundamental sections: user interfaces, business rules and database tables. A Business Object is a set of business rules needed to aggregate data for user interfaces, as well as to disaggregate data to read and write database tables: e.g. in an user interface an invoice is seen as an unique document, but its data is stored into two different database tables (see *Master/Detail Entity*).

Candidate Key

A candidate key is any possible unique identifier for a row within a database table. A candidate key can be either a singleton or a composite key. Every entity must have at least one candidate key, whereas a table can have more than one. *Attributes* in *Candidate Keys* do not allow nulls.

CardinalityCODEPAINTER **REVOLUTION** provides a set of well-defined relationships and a set of well-defined entities, therefore the cardinality is determined by the relationship type and the entity type. When dealing with simple *Relationship Links* the cardinality is one-to-many (including zeros on both sides), no matter what the kind of entity you choose. When dealing with *Parent/Child Relationships* the cardinality is one-to-one - when you establish a link between two Master File Entities; or one-to-many - when you establish a link between a *Master File Entity* and a *Detail File Entity* (or a *Master/Detail Entity*).

Codify Phase

The Codify Phase starts once you have defined all entities in your project. During this phase you can define all entity's details required to build the final application.

Company Name Option

The database table name contains the company name. This is used when you need different database tables for each company.

CPROWNUM Field

While defining an entity, it could be difficult to identify a correct *Primary Key*. To deal with this problem you can use CPROWNUM. This field is managed by CODEPAINTER **REVOLUTION** and will always be a valid *Primary Key*. It is particularly useful for Detail File entities and Master/Detail entities. By default the field is not shown.

Create Record If It Does Not Exist Option

If the *Foreign Key* of the linked table does not exist a new database table row will be created.

Data Name Field

In this field you need to define the symbolic name of the database table.

Database Table Entity

This entity has a database table only, and not a user interface. It could be useful to link this kind of entity with a *Routine Entity* or with a *Dialog Window Entity*. This entity is commonly used as a working table or to store historical data.

Dataflow Link

This link is used during the *Design Phase* to graphically represent how data flows through the entities. At one link end you need to define a Routine Entity. **CODEPAINTER REVOLUTION** will not generate any code associated to this link, as it is used for documentation purpose only.

Design Phase

When you are asked to build a new SW application you normally interview your client to make a list of requirements that the application must fulfill. After having gathered the required information, you start defining your entities and their links. This is what the Design Phase is all about. **CODEPAINTER REVOLUTION** gives you a user-friendly interface to build your application plan easily. During this phase, you can define links with *Referential Integrity*, fields, database tables, etc.

Detail File Entity

Medium to high sized business organizations are usually divided into business units and/or departments. To manage this kind of organizations, a series documents for internal use only are required. Usually, the main part of these documents is the body, meaning that header and footer are empty or collapsed in the body rows. You can manage this kind of data using the Detail File class. The Detail File Entity is made of a data-entry form and a database table. The *Primary Key* of the table has a fixed field (representing the collapsed footer and header information) and a varying one (representing each row of the body of the document). Please note that the Detail File Entity is a special case of the *Master/Detail Entity*. A typical example is the list of "Things To Do" per worker. The database table could have the following fields: Date, Department

GLOSSARY

No, Worker No., Worker Name, Task to Complete, and Status of the task. The *Primary Key* could be made of: Date + Department No. + Worker No. The collapsed information would be Date and Department No. While programming with **CODEPAINTER REVOLUTION**, you can notice that it is quite common to create a *Parent/Child Relationship* between a *Master File Entity* and a *Detail File Entity*. In this case, the document's header and footer information is stored in the Master File table, while the related body information is stored in the Child Detail File table. To understand when to use a Detail file linked through a *Parent/Child Relationship* to a *Master File Entity* and when to use a *Master/Detail Entity* you need to have a careful look at the problem itself. In the first case, during the analysis, we notice that we have two different entities, i.e. two different *Business Objects*; in the second case, we have a single entity, i.e. a single *Business Object*.

Dialog Window Entity

This entity has a user interface and not an associated database table. Usually, this entity is linked to other entities from which it retrieves or stores data. This entity cannot be used as origin of a link, which is different from an event

Eighth Index Field

see Second Index Field.

Eleventh Index Field

see Second Index Field.

Entity

An entity is a noun - a person, place, or thing. An entity is the subject or object of an action, and may be related to other entities. An entity can be tangible, such as a person, place, or animal, or intangible, such as a banking transaction, a division in a company, or a payroll period.

Event Link

When an Event link is established, the execution of the entity will depend on an event. For example, if management requires a monthly report orders received, every month a procedure must be triggered to compute it. These links can be created between any two entities in the design plan. The *Database Table Entity* represents an exception to the rule because it cannot be triggered by an event.

External Table Option

This option is used to take the database table of an entity from another application, which cannot be updated.

Externally Linkable Option

This option allows to link entities in the current project with entities of other projects.

Fifth Index Field

see Second Index Field.

Foreign Key

A foreign key exists when one or more *Attributes* in an entity refer to a *Primary* or *Alternate Key* of another entity. When the foreign key is a composite key and it is not relating to the primary entity, then all foreign key *Attributes* must be null.

Fourteenth Index Field

see Second Index Field.

Fourth Index Field

see Second Index Field.

Key

A key is a set of columns that define a table. There are four types of keys: *Candidate Keys*, *Primary Keys*, *Alternate Keys*, and *Foreign Keys*. Along with the four types of keys, there are two basic key classes: composite and singleton. A composite key is a key composed of more than one *Attribute*. Composite keys force joins to be more complex. A singleton key is any key made up of only one *Attribute*.

Master File Entity

When dealing with business/commercial applications, you usually need to have files containing information about clients, suppliers, accounts, etc. You can manage this kind of data using the Master Files class. The Master File Entity has a database table and a data-entry form. This entity should be used when records must be processed one by one (i.e. each record is processed on its own): e.g. to insert a new client in the file you need to write a single row in the database table. Similarly, when we want to remove a client from the file

GLOSSARY

you need to delete a single database table row; and again, to change a client's name in you need to modify a single row. **CODEPAINTER REVOLUTION** allows you to define database table name, fields and keys. Moreover it allows you to edit the data-entry form. When you generate the project this information is gathered together and an application is built. This application will use the defined data-entry form to insert, remove, modify and display database table rows.

Master/Detail Entity

Business transactions are normally represented by documents such as invoices, shipping bills, purchase orders, expense statements, etc. The main characteristic of these documents is that they have at least three different sections: header, body and footer. This implies that three different kinds of records must be managed. This can be achieved using the Master/Detail class. The Master/Detail Entity has a data-entry form and two database tables (Master table and Detail table). The Master table has the document's header and the footer, i.e. all unrepeated fields. The Detail table contains the document's body (all repeated fields) along with all unrepeated fields composing the Master table's *Primary Key*. Let's consider an invoice as example. A typical header could have Customer Name, Customer Address, Customer Phone No., Order No., and Date. A typical footer may include payment details. We could choose the Order No. as the *Primary Key* of the Master table in which all information will be stored. A typical body of an invoice could have product code, description, shipped quantity, and price per unit. The Detail table will store this information together with the Order No; i.e. the *Primary Key* of the Master table. Note that at least one repeated field must be part of the Detail Table *Primary Key*. For certain kind of documents, e.g. invoices, it could be difficult to find out which repeated fields could uniquely identify a body's row. To deal with this problem, **CODEPAINTER REVOLUTION** introduces the *CPROWNUM Field*. This special numeric field automatically counts the document's row becoming the Detail Table unique key.

Menu option

This option creates a menu item to activate the designed entity.

Ninth Index Field

see Second Index Field.

Output Entity

This entity reads data from database tables and displays or prints it. Examples

of output entities are: *Zooms*, Graphs, Reports, Word documents and Excel documents.

Parent/Child Integrity

When you delete a database table row in the Parent Entity the corresponding rows in the Child Entity will be automatically deleted, too. Moreover, you can insert a row in the Child Entity database table if corresponding rows in the Parent Entity exist.

Parent/Child Relationship Link

The Parent/Child Relationship is a particular kind of relationship between two entities: the Parent and the Child. The Child entity must have either a composite primary key made of the Parent Entity primary key and a series of Child Entity *Attributes*, or its primary key is identical to the Parent Entity primary key. When this link is used CODEPAINTER REVOLUTION automatically establishes *Parent/Child Integrity*. Commands for inserting and modifying rows in the Child Entity must be called within the Parent Entity window.

Physical Name Field

In this field you can define the physical name of the database table.

Primary Key

A primary key is made of one or more *Attributes* that uniquely identify a database table row. A *Candidate Key* can become an entity's primary key.

Prototype Option

You can create a prototype program for entities defined in the design plan. Warning: deselect this option before running the *Codify Phase*, otherwise you will lose all changes made using the various painters.

Read Data From.. Link

It is used to let *Routine Entities*, *Dialog Window Entities*, and *Output Entities* Read Data From other entities. Integrity rules are not included. It is mainly used during the *Design Phase* in order to have more detailed project documentation.

Referential Integrity

Refers to rules ensuring that primary and foreign keys remain consistent

GLOSSARY

whenever database table rows are inserted, updated, or deleted.

Relationship Link

A relational database has the ability to relate or associate information about different entities in one database table. A relationship is normally a verb, an action, or an ongoing interaction between entities. Relationships stand between entities and describe their interaction. In **CODEPAINTER REVOLUTION**, you can connect two entities with a Relationship link. This kind of link can be created in the Design Phase and will ensure *Referential Integrity* between the database tables. If you want to avoid *Referential Integrity*, you can create a link directly in the *Codify Phase*.

Repeated Option

This option defines fields belonging to the document body, meaning that the field has multiple values within the document.

Routine Entity

Routine Entities are pieces of code without user interface windows. Any relevant procedure or function or batch can be represented with this entity within the *Design Phase*. During the *Codify Phase*, you can write your code in a portable (with respect of **CODEPAINTER REVOLUTION** host languages) and powerful programming language. The routine entity can be connected to other entities using the *Event Link*, the *Read data from.. Link* and the *Dataflow Link*.

Second Index Field

The *Primary Key* is the main search criterion for a database table but sometime, when dealing with large databases, a query could turn-up to be slow. To speed-up your queries, **CODEPAINTER REVOLUTION** allows you to define up to ten different indexes.

Seventh Index Field

see Second Index Field.

Sixth Index Field

see Second Index Field.

Template

It represents the file containing the source code used to generate your application procedures. The source code contained is just a skeleton of the

final code and must be integrated with user specified names and options.

Tenth Index Field

see Second Index Field.

Third Index Field

see Second Index Field.

Thirteenth Index Field

see Second Index Field.

Twelfth Index Field

see Second Index Field.

Zoom

It is used to display a query in tabular mode.

Chapter 3

Codify Glossary

3.1 Codify Glossary Elements

Always Enabled Option

When this option is deselected, the button will be enabled in the editing mode only.

Application Files

Set of files automatically produced by CODEPAINTER REVOLUTION. To run the application, execute the file "cp3start".

Autonumber Menu Item

This option opens a dialog window in which you can define automatically increasing numbers for field groups.

Bind To Design Entity Option

The current object is bound to the corresponding design entity. This option must be deselected when you define an entity that uses data from another one, e.g. when you want to create a *Master File* from the master database table of a *Master/Detail Entity*.

GLOSSARY

Calc/Link Depends On

The field value is recalculated every time the value of one of the variables defined in the dialog box is changed.

Codify Phase

The Codify Phase starts once you have defined all entities in your project. During this phase you can define all entity's details required to build the final application.

CPROWNUM Field

While defining an entity, it could be difficult to identify a correct *Primary Key*. To deal with this problem you can use CPROWNUM. This field is managed by **CODEPAINTER REVOLUTION** and will always be a valid *Primary Key*. It is particularly useful for *Detail File* entities and *Master/Detail* entities. By default the field is not shown.

Create Record If It Does Not Exist

If the *Foreign Key* of the linked table does not exist a new database table row will be created.

Design Phase

When you are asked to build a new SW application you normally interview your client to make a list of requirements that the application must fulfil. After having gathered the required information, you start defining your entities and their links. This is what the Design Phase is all about. **CODEPAINTER REVOLUTION** gives you a user-friendly interface to build your application plan easily. During this phase, you can define links with *Referential Integrity*, fields, *Database Tables*, etc.

Edit Under Condition Option

This option gives you the possibility to edit a field only if an expression is TRUE.

Fifth Index Field

see Second Index Field.

Fourth Index Field

see Second Index Field.

Group Menu Item

Entities, which are logically related, can be added in, modified and removed from groups. This organization will affect the way files are split in the different directories.

Hide Under Condition Option

This option gives you the possibility to hide a field when an expression is TRUE.

Large Application Option

Setting this option, *Project Files* are organized in different subdirectories, which are created under the main project directory. Definition files are saved in `..\ProjectName`, the FoxPro source code in `..\ProjectName\vfsrc`, the Visual Basic source code in `..\ProjectName\vbsrc`, and the Java source code in `..\ProjectName\Javasrc`. The final application is saved in `..\ProjectName\exe`.

Local Option

When this flag is active, the variable is excluded from the calling procedure (i.e. the parent object) of the *Dialog Window Entity*.

Manual Block Menu Item

Manual blocks identify areas where you can write your own code. CODEPAINTER REVOLUTION will leave these areas unchanged even when the code is regenerated.

Modal Object Option

Editing on a modal window must be terminated before switching to other windows. This option allows you to create a modal object.

Numb. Of Search Criteria Option

When browsing a *Database Table* you can use more than one search criterion. This option allows you to specify how many search criteria are going to be used.

O.S. Field

In this field you can define the Operating System under which your application must run.

GLOSSARY

Print_Prg Field

In this field you can define the name of the procedure called when, in the querying mode, the "F2" key is pressed.

Project File

File used to define the application.

Project File

File used to define the application.

Ref. Field

In this field you can define the name of the optional business object variable pointing to an object.

Repeated Option

This option defines fields belonging to the document body, meaning that the field has multiple values within the document.

Revision Counter Field

This field shows how many times the object has been modified.

Second Index Field

The *Primary Key* is the main search criterion for a database table but sometime, when dealing with large databases, a query could turn-up to be slow. To speed-up your queries, CODEPAINTER REVOLUTION allows you to define up to ten different indexes.

Seventh Index Field

see Second Index Field.

Sixth Index Field

see Second Index Field.

Third Index Field

see Second Index Field.

Zero Filling Option

When this option is selected and you insert a number (different from zero) in a character field, zeroes will be added to the left hand side of the field until the length limit is reached. This is useful when the *Autonumber Option* is selected as well.

Zoom On Zoom Field

In this field you can define the name of the routine that must be called when, in the zooming mode, the zoom button is pressed.

Zoom

It is used to display a query in tabular mode.

Chapter 4

Front End Glossary

4.1 Front End Glossary Elements

Application Files

Set of files automatically produced by CODEPAINTER REVOLUTION. To run the application, execute the file "cp3start".

Group Menu Item

Entities, which are logically related, can be added in, modified and removed from groups. This organization will affect the way files are split in the different directories.

Large Application Option

Setting this option, *Project Files* are organized in different subdirectories, which are created under the main project directory. Definition files are saved in `..\ProjectName`, the FoxPro source code in `..\ProjectName\vfsrc`, the Visual Basic source code in `..\ProjectName\vbsrc`, and the Java source code in `..\ProjectName\Javasrc`. The final application is saved in `..\ProjectName\exe`.

Project File

File used to define the application.