WHITE PAPER

# ACI 4th Dimension: The widest scope of built-in database features

4TH DIMENSION is the most evolved database environment in today's market. Years of experience in database design, interface building, and constructive feedback from power users and professionals have clearly defined the 4D principles:

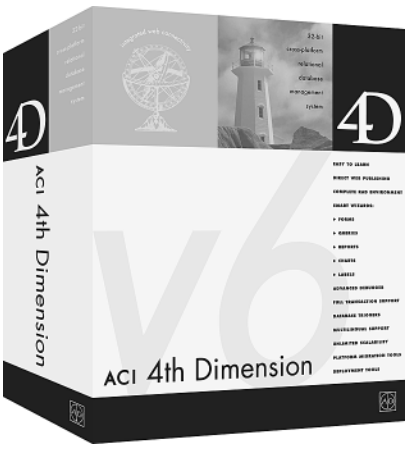
- Automate everything which can be automated to guarantee ease of use, security, and of simplicity of administration.
- Deliver an efficient graphical object-based model—thus, drastically reducing the need for code.
- Provide the widest possible range of capabilities with built-in features to ensure full consistency in database design and operation and to facilitate database maintenance.
- Offer easy, straightforward code re-use.
- Offer transparent cross-platform, Web, and client/server scalability.

Features described in this document apply to:

- 4D for Windows and for Macintosh: any application created on one platform can be used as-is on the other.
- 4D Server: 4D Server is the client/server version of 4D. Any 4D application can be deployed into a client/server architecture. By default, 4D Server has the same built-in capabilities as 4th Dimension, plus additional server specific characteristics (which will be detailed in another White Paper).

4D, 4D Client, and 4D Server share the same interface tools, the same language, and the same information management scheme. Using 4D you can debug code for your server's stored procedures on one single laptop, even on an airplane! Or if you use 4D Server as your multi-user development system, you can create a 4D stand-alone vertical application.

This document is a survey of the several hundreds of built-in features and options that 4th Dimension provides through menus or dialogs that are immediately available to users and developers. These features can be used as-is. 4D does not require you to recreate programmatically what you have already performed graphically. For each section, we have listed both the high- and low-level commands related to the theme described. You would use high-level commands if you need to call 4D built-in capabilities within a fully customized application. For example, using the command will start a new process. On the other hand, you would use low-level commands to communicate messages between processes.



# Contents

# Database environment

4D provides automated and customizable global database environment controls in three main areas:

- External environment controls
- Internal environment controls
- Localization

These tightly integrated environment controls are the key elements that allow your database to operate seamlessly and trouble-free.

## THE 4D EXTERNAL ENVIRONMENT MANAGER

The 4D External Environment Manager automatically checks your working environment. For instance, 4D automatically warns you if you run out of disk space or if the appropriate network has not been installed, etc. The 4D Manager can also pass information to developers for possible programmatic controls.

A significant part of environmental information is available in the 4D About Box and includes the following:

### Database context

A 4D application is composed of a structure file and a data file. This separation allows increased flexibility for upgrading applications created by vertical developers. Conversely, you can deliberately associate a data file and a structure file (WEDD).

- Tables, records, and privileges information
- Data structure file location
- Data file location

### Monitor

- Screen size
- Number of colors

### Network settings

- Computer name
- User name

### Volume information

Knowing volume information allows for easy optimization of disk space and for making logical decisions regarding possible needs for data segment sizing (which is performed through a simple dialog).

- Name
- Size
- Portion used by application
- Free space on volume

Programmatically, you can manipulate environmental parameters using commands related to the System Environment:

Count screens, Current machine, Current machine owner, FONT LIST, Font name, Font number, Gestalt, Menu bar height, Menu bar screen, SCREEN COORDINATES, SCREEN DEPTH, Screen height, Screen width, System folder, Temporary folder, SET SCREEN DEPTH

## INTERNAL ENVIRONMENT CONTROLS: DATABASE GENERAL PROPERTIES

Database global properties for the current database can be defined in a Properties window. You can override these properties, if necessary, at the form level or programmatically.

### General

- Choosing a startup environment: Design, User, Custom Menus.

  4D provides three main environments. The Design environment is where you create your application and underlying database. The user environment is the generic 4D end-user interface. The custom menu environment where you execute custom applications.
- Progress indicators: Numbers or thermometers
- Show toolbar
- Drag-and-drop highlight mechanisms (frame, pattern, or both)

- Web Server: Publish database at startup

  Database publishing on the Web is a built-in feature of 4D (see section on 4D Web Services).

- TCP port number: You can run several Web servers on the same computer and select a different TCP port for each Web server.

**Data control**

- Mandatory log file: The log file is designed to keep track of changes within the database since the last back up. If necessary, the database can be restored to the state it was in at a previous point in time.
- Deletion controls activate deletion controls for related tables
- Automatic transaction during data entry: This option is designed for 4D First users upgrading to 4D.
- Customizable settings for flushing data buffers

**Data access**

- Allow 4D Open connections

  4D Open is the application programming interface (API) for 4D that allows non-4D Clients to connect to 4D Server, and that enables creation of distributed client/server architectures.

- Structure access group privileges: Gives the specified group permission to enter the Design environment (see Passwords System)
- Hide/Display User List in Password dialog box
- User List in alphabetical order

**Platform interface**

- Automatic, Mac OS, Copland, Windows 3.1, Windows 95/Windows NT (See section on 4D cross-platform management.)
- Default font: System font or any other font available in list
- Message font and font size
- Edit style sheets: Style sheets for each available platform.

**Table and field defaults**

- Color Names or background: 4D allows color customization to organize fields and tables in the Design environment
- Icons for field types: You can display icons to indicate a field's type

**Method editor**

- Display methods in either a textual or flowchart representation.

  Note: Programming in a flowchart was implemented in 4D several years ago. It is maintained mostly for compatibility reasons.

- Show or hide key words at the bottom of the Method Editor
- Old startup method scheme and old file procedure scheme: To ensure full compatibility between existing applications and 4D V6, you have the ability to execute features found in previous versions of 4D.

**Design printing**

- Print titles: Select whether or not to print titles when you print the structure, forms, methods or passwords.

**Scheduler**

Using the scheduler, you can define priority level for each of your servers running on a pre-emptive OS such as Windows NT.

- Number of ticks between calls to OS
- Maximum number of ticks per call to OS
- Minimum number of ticks per call to OS

**Data cache and memory**

You can configure cache and memory settings to gain the best performances.

- Use new memory allocation scheme on Macintosh
- Maximum cache

- Application main memory: Allows you to define how much memory to allocate for your application while running under Windows.
- Screen update control: You can accelerate screen updates (at the expense of memory)

#### Client/server and Web server connections timeout

You can customize timeouts for client/server and Web server connections. However, 4D releases and closes automatically all objects used in the database even in case of an unintentional quit.

#### Display formats and entry filters

In addition to an impressive list of possible display formats for all fields (see section on Forms), 4D allows customized display formats and entry filters displayed in the object properties regular window. These formats and filters behave as style sheets. Any change is carried through all the forms using that format or that entry filter.

#### Using Customizer Plus

Customizer Plus is a utility provided with 4D that enables further customization. In particular, Customizer Plus allows preliminary customization of the General Database Properties window described above.

#### Additional preference settings

- Keyboard: You can customize keyboard shortcuts for main actions (data entry validation, cancelation, and adding a record in a sub-table)
- Windows: Controlling the placement and size of windows
- Preferences: Control over the size of the stack for the seven main processes, cursors, printing modes, digit precision, etc.
- Font: Control over fonts when printing methods
- Cache: Optimizing memory management for a specific application

#### Version controls

- Forcing the link between the data file and the application to avoid access to an incorrect data file.
- Forcing updates of all external extensions on all the 4D Client machines. This features applies to 4D Server automated application services and will be discussed in the 4D Server White Paper.

#### Network components

4D includes a virtual network manager that allows simultaneous access to network protocols (TCP/IP, ADSP, IPX). 4D can also emulate a network in memory, for example, you can simulate client/server using only one machine for testing or development.

- Optimizing ADSP communication between 4D Server and 4D Client
- Customizing port number for IPX and TCP access
- Timeout management

#### Localization

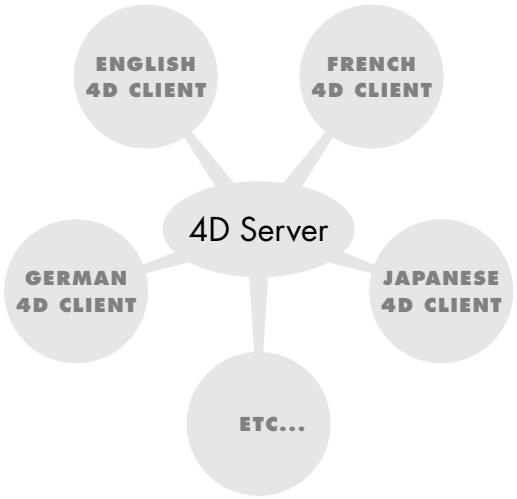
- Adapting ASCII internal maps to country-specific operating systems (for instance Windows in Greek)
- Automate display of command names in a local language (for example, running an application written with English commands with German or French commands).
- Using 4D with Script Manager

See following section on localizing applications.

### LOCALIZING APPLICATIONS

4D provides easy customization for multi-lingual or international applications. The same 4D application can be used in several languages with no additional development efforts:

- 4D tables and fields are tokenized and can be interpreted in the language used to open the application.
- 4D includes built-in management of resources. Therefore, instead of hard coding the menu text, menu items, buttons, and objects forms, you can use resource references within your forms and use 4D Insider to easily translate your string resources.
- Within a 4D Server client/server architecture, French users can access a database originally designed for English users, because localization occurs on the client side. Since interface management, menus, alerts, and commands sit on the client side, the same application can be accessed by multi-lingual clients. Each client benefits from all the characteristics proper to a specific language: comparison strings, diacritical marks, sorting order, display formats for date, time, telephone, currencies, etc.



# Cross-platform management and built-in Web support

Because 4D is a true cross-platform comprehensive development tool, companies that use 4D save the time and the cost associated with maintaining multiple development environments.

- A 4D application created for one platform can be deployed immediately on another. 4D provides native support for Windows 95, Windows NT, Macintosh, and Power Macintosh.
- Any single-user application can be deployed as a Web server using any commercial browser and scales easily to a workgroup client/server solution using 4D Server - which can also act as a Web server. More details are provided on Web services in the following section and in the 4D Internet White Paper.

## CROSS-PLATFORM MANAGEMENT

Cross-platform management is managed automatically by 4D. Any 4D application can be compiled by 4D Compiler to execute on either Macintosh, on Windows or both. Compiling can be optimized to take advantage of platforms specifics, such as math co-processors.

### Global appearance

Users and developers can specify the appearance of their applications by selecting a platform appearance. Supported platforms are:

- Automatic
- Mac OS
- Copland
- Windows 3.1
- Windows 95/Windows NT

You can let 4D automatically adjust the application interface depending on the machine used to execute it. Or you can select an interface style regardless of the platform on which the application is running. For example, if you run on Windows NT, you are allowed to have an application with a Copland look and feel, or the reverse.

Platform appearance can be defined at these levels:

- Database level
- Form level
- Object level

Because of the object-base nature of platform inheritance, the form level overrides for that form (and that form only) the choice previously made at the database level. A choice at the object level overrides for that object (and that object only in a given form), the choice previously made at the form level.

### Font, font size and font style

Style sheets allow you to create database specific style sheets for fonts, font sizes, and font styles. These styles can be applied to any or all database objects throughout the database—thus, allowing you to perform any modification in one step.

You can also manually control the appearance of each form and even each object within a form. You can also create style sheets for forms and objects. For example, you could create a style sheet for group boxes and set a specific platform appearance as one of the properties of any group box on your forms.

### Screens and windows

4D automatically identifies the screen and window types appropriate to a given platform and makes all necessary adjustments since objects created on Macintosh look smaller when viewed on Windows and vice versa. You can also perform these adjustments manually on a per form basis, and select the scaling ratio you want for objects in your form.

### GUI metaphors

Just because 4D runs on multiple platforms does not mean that you are restricted to only those interface elements that are common to all platforms. 4D makes all elements available to you regardless of the platform. For instance, if you prefer to use a combo box instead of a pop-up menu, you are free to do so, even if you are on a Macintosh.

### OS and environmental differences

Because OS offers platform dependent features, 4D offers commands to deal with such occurrences. For example, Apple Events does not work on Windows, and DDE is not available on the Macintosh. This does not mean that you cannot take advantage of these platform differences; you can use a control flow structure allowing 4D to execute the appropriate method for the platform on which the application is running. However, in all cases, before you choose to implement platform dependent features, make sure there is no built-in 4D feature capable of performing that task. In most cases, you will find that there is one.

## BUILT-IN WEB SUPPORT

4th Dimension includes an integrated Web server. No CGI or additional software is required to serve your data via the Web. Web handling is treated as a standard database task within the 4D architecture. (A white paper on all the 4D built-in Web services is available on the ACI US Web site at www.acius.com.)

### Plug and play

- Automatic interface management
- Automatic publishing: Database publishing on the Web is an automatic property. You publish your database by checking a box, selecting a menu or using a built-in command.

### Main characteristics

- On-line, transparent HTML translation: 4D transparently and dynamically translates forms and design components into HTML pages.
- Dynamic access to records and data: 4D handles Web browsers as standard clients of the 4D database engine. Using 4D Connectivity Plug-ins created by ACI or 4D Partners, you can select, insert, update, and delete data from SQL databases or legacy systems. The Web browser benefits from the powerful 4D features.
- Automatic session maintenance and context: Throughout the URLs of the Web pages, 4D maintains unique context and subcontext ID numbers ensuring complete synchronization between the current Web page displayed in the browser and the context of the 4D database connection.
- Automatic multi-user management, record locking and release, transaction and rollback management.
- Web Process maintenance: A Web process is managed using the 4D process architecture.
- Optimized Web Server architecture: The 4D Web Server engine has the same capabilities as the 4D database engine.
- HTML and JavaScript encapsulation: You can encapsulate HTML and JavaScript code to customize your 4D development.
- Binding between HTML and 4D objects: 4D provides a simple, built-in scheme for binding HTML objects to your 4D variables—your objects only need to have the same name.

### Third-party products

- Communicating with existing commercial Web servers
- Encrypting HTTP connections
- Java libraries
- FTP, SMTP, POP3, NNTP, PING, and low-level TCP/IP support (See white paper on 4D Internet for details.)

For programmatic control over your Web database, you can use the following commands:

On Web Connection Database Method, SEND HTML FILE, SET HTML ROOT, SET WEB DISPLAY LIMIT, SET WEB TIMEOUT, START WEB SERVER, STOP WEB SERVER

# Multi-process management

Multi-processing, multi-tasking or multi-threading sound like highly technical words and tend to scare non-specialists. What it simply means is that your program can perform several tasks simultaneously. 4D delivers this capability both automatically and at the user's request. A list of all the database processes is viewed easily by choosing a menu command.

Multiple processes add a significant dimension to multi-windows applications since they not only allow multiple active windows, but also allow each of these windows to execute their functions simultaneously. Since Macintosh supports multiple monitors connected to the same computer, you can take advantage of 4D multi-processing and multi-windowing capabilities for optimized visualization of several active processes. For example, a chemical engineer can have one screen displaying a graph of temperature fluctuations during an experiment, while using another window to note physical changes in the chemical reaction.

## 4D AUTOMATICALLY GENERATED PROCESSES

Automatically opened by 4D

When 4D launches, it automatically creates three processes for managing all aspects of development and data access.

- User/Custom Menus process
- Design process
- Cache Manager process: flushing and caching data to disk

Because 4D is structured as a multi-processing system, it delivers an interactive relationship between its user and design environments. The benefit is that any modification you perform in the design of a form is automatically and immediately reflected at the user level. Interaction inside 4D allows you to develop and test a database at the same time. For instance, you can design the appearance of a 4D form for the Web and simultaneously verify the appearance of that form within the browser accessing the database.

### Automatically opened when useful

4D also automatically opens three additional processes only as needed, thereby preserving valuable system memory:

- Indexing process
- On Serial Port Manager
- Web server process
- Event Manager

## USER-DEFINED PROCESSES

Multi-processing allows 4D to execute several operations simultaneously. For instance, while one process is printing a selection of records, another can be importing data, and a third can used for data entry.

### Creating a process

- With New Process command
- By selecting a check box in the Menu Editor
- By selecting a check box when executing a method

### Capabilities

Launching a new process allows you to access all 4D capabilities. Multi-processing is similar to having several 4Ds in one. Describing process features would be like describing 4D itself - which is the purpose of this entire document. However, here is a short list of striking capabilities which 4D provides:

- Work with more than one active Window
- Manipulate several current selections simultaneously (see Information Management section)
- Work with more than one current record simultaneously
- Work with more than one current input and output form simultaneously
- Start a lengthy operation separately
- Create customized user interface devices (floating palettes, customizable tool bars, etc.)

- Complete control over communication between processes using inter-process variables and commands.
- 4D debugger allows you to simultaneously execute and debug a process. Each process can have its own debug window (see 4D Debugger below)

#### GRAPHICAL PROCESS ADMINISTRATION

4D provides a comprehensive information display including:

- Process number
- Process ID
- Current status of the process (executing, delayed, waiting for user event, waiting for input/output, waiting for internal flag, paused, aborted, hidden)
- Execution time since the process was started

**Within methods, use...**

Communication: CALL PROCESS, CLEAR SEMAPHORE, GET PROCESS VARIABLE, Semaphore, SET PROCESS VARIABLE

User interface: BRING TO FRONT, Frontmost process, HIDE PROCESS, SHOW PROCESS

General: Count tasks, Count user processes, Count users, Current process, DELAY PROCESS, Execute on server, New process, PAUSE PROCESS, Process number, PROCESS PROPERTIES, Process state, Processes, RESUME PROCESS, Stored Procedures

# Structure design

With 4D, you spend more of your time creating solutions than on managing database tables.

- Visual representation of tables and columns provide a color-coded Entity Relationship Diagram (ERD).
- Independent colors assigned to tables, fields, and relations provide a clear image of your business model.
- Straightforward display of tables, fields, field data types, and relations ease structure viewing and modification.
- Powerful, drill-down Explorer enables easy navigation though all 4D objects.

4D Server includes a complete multi-user development environment:

- 4D automatically manages check-in and check-out of all database objects.
- Objects previously checked out by other programmers can still be viewed and manipulated in read-only state.
- Database modifications are reflected in real time for each developer on the team. No synchronization is required. 4D updates are immediately available to all programmers as soon as they are made.
- Because 4D Server integrates a multi-user development environment with its database engine, developers can test and program simultaneously.

## TABLES

### Appearance

- Visible/invisible
- Vertical/horizontal resizing
- Vertical/horizontal scrolling
- Colors

### Access privileges

Access privileges are assigned to each table based on the built-in 4D password system. They allow the designer to control on a table by table basis loading, saving, adding, and deleting records.

### Triggers

Triggers allow centralized code execution upon common database actions, including loading, saving, adding, and deleting of records.

### Data segment management

4D supports files up to 128 gigabytes. The 4D data file can be segmented and partitioned such that a single data file can span multiple volumes. This segmentation can occur at any time regardless of the number of records in the database. 4D intelligently stores data in the available data segments.

### Table and structure-related commands

- Table: Current default table, Current form table, DEFAULT TABLE, INPUT FORM, OUTPUT FORM.
- Structure access: Count fields, Count tables, Field name, GET FIELD PROPERTIES, SET INDEX.
- Triggers: Database event, Trigger level, TRIGGER PROPERTIES

## RELATIONS

### Appearance

- Creating a relation between tables is done simply by drawing a line between two related fields. Alternately, relations can be created by entering information into the related fields. 4D will automatically display the relation graphically
- User selectable colors allows for easy viewing of relations and cascading relationships
- Graphical representation of relation is automatically reorganized if tables are moved within the Design editor
- Relations, tables and fields are highlighted if you select a relating field or click on a relation

### Many-to-one and one-to-many

4D automated relations establish joins automatically without writing code.

When one-related records are added, 4D enforces relational integrity automatically by assigning related values in another table or subform.

At the developer's request, 4D can automatically handle the deletion of records:

- Leaving related many intact
- Deleting related many
- Preventing one-records from being deleted if related-many records exist

All these features can be programmatically controlled using the following commands:

CREATE RELATED ONE, RELATE MANY, RELATE MANY SELECTION, RELATE ONE, RELATE ONE SELECTION, SAVE RELATED ONE.

For compatibility reasons: OLD RELATED MANY, OLD RELATED ONE, SAVE OLD RELATED ONE

## FIELDS

Field main properties are established during structure design. These properties are inherited automatically by any form in which the field is used. With the exception of indexing and uniqueness, most fields properties can be overridden at the form level.

4D transparently accommodates all international standards and local requirements. For instance, a numeric field could be used as dollars in the United States or DM in Germany. Style sheets allow for centralized management of field formatting throughout the database and, just like any field property, can be overridden within a specific form.

4D provides eleven different field types:

### Alpha

Up to 80 characters in length

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify,
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

### Text

Up to 32,767 characters

Can be overridden at form level:

- Mandatory, Display only, Can't Modify
- Visible/Invisible
- Color
- Tips and Ballon Helps

Note: For long texts, you can also use 4D Write, a word processor plug-in created for 4D.

### Real

19 significant digits, decimal precision of 18, scientific notation support

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

### Integer

All values between -32,768 and +32,767

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

**Long integer**

All values between -2,147,483,648 and +2,147,483,647

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

**Date**

Year 2000 compatible

Static

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

**Time**

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and balloon help

**Boolean**

Static:

- Indexed
- Unique

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Choice lists
- Tips and ballon help

**Picture**

Can be overridden at form level:

- Mandatory, display only, can't modify
- Visible/invisible
- Color
- Compression options: Animation, component video, graphics, photo JPEG, video
- Black and white, up to 256 levels of gray, and millions of colors.
- Tips and balloon help

Note: Picture fields support objects other than pictures: Sounds, 4D Productivity Plug-ins data, and any data, document, or file manipulated through an external area.

#### Subtable

A subtable generates a table nested within a field. Subtables are multi-valued fields used when the parent record contains a variable amount of data. Subfields inside the subtable can be any conventional type. Subtables are automatically sized inside the parent form and can be printed with variable frames.

- All fields available, color
- Tips and ballon help

### BLOB

Up to 2 gigabytes

Can be overridden at form level:

- Visible/invisible
- Color
- Tips and balloon help

#### Within methods, use...

String: ASCII, Change string, Char, Character Reference Symbols, Delete string, Insert string, ISO to Mac, Length, Lowercase, Mac to ISO, Mac to Win, Num, Position, Replace string, String, Substring, Uppercase, Win to Mac

Series: Average, Max, Min, On a Series, Std deviation, Sum, Sum squares,Variance

Math: Abs, Arctan, Cos, Dec, Exp, Int, Log, Mod, Random, Round, SET REAL COMPARISON LEVEL, Sin, Square root,Tan,Trunc

Date and Time: Add to date, Current date, Current time, Date, Day number, Day of, Milliseconds, Month of, SET DEFAULT CENTURY, Tickcount, Time, Time string, Year of

Picture: COMPRESS PICTURE, COMPRESS PICTURE FILE, LOAD COMPRESS PICTURE FROM FILE, PICTURE PROPERTIES, Picture size, SAVE PICTURE TO FILE

Boolean: False, Not, True

BLOB: BLOB PROPERTIES, BLOB size, BLOB TO DOCUMENT, BLOB to integer, BLOB to list, BLOB to longint, BLOB to real, BLOB to text, BLOB TO VARIABLE, COMPRESS BLOB, COPY BLOB, DELETE FROM BLOB, DOCUMENT TO BLOB, EXPAND BLOB, INSERT IN BLOB, INTEGER TO BLOB, LIST TO BLOB, LONGINT TO BLOB, REAL TO BLOB, SET BLOB SIZE, TEXT TO BLOB, VARIABLE TO BLOB

### CHOICE LISTS

A choice list shows the predefined values that can be easily assigned to most 4D fields and variable objects. Lists can be assigned at the database level and overridden at the form or object level. Lists are dynamic in nature and can be manually or programmatically modified.

#### Common uses for lists

- Provide the user items found in the list that are associated with an object
- Restrict the user to items in the list
- Exclude the items in the list from being entered

#### Specifications

- Fields can be attached to different lists depending on the form
- Hierarchical lists: a list can be linked to another list. For example, cats can be linked to mammals and mammals to animals.
- Icons can be attached to an item
- Items in a list can be sorted

#### Within methods, use...

ARRAY TO LIST, LIST TO ARRAY, BLOB to list, List to BLOB, SET CHOICE LIST

### AUTOMATIC CREATION OF 4D STRUCTURES

4D structures can be created automatically in the following situations:

- Using 4D standard import dialog

Through the standard import dialog, 4D can create tables by simply importing a DBF file.

- Using 4D Insider

  4D Insider is a point-and-click, drag-and-drop code servicing system enabling component builders and users to modularize or merge existing applications, and to examine database objects and their dependencies.

  1. Create complete 4D databases with previously created components

  2. Create tables based on an SQL2 description model

- Using 4D ODBC

  You can insert any table from ODBC compliant database applications within 4D.

- Using 4D SQL Server and 4D for ORACLE

  You can create a 4D structure based on existing Sybase, MS SQL Server, and Oracle tables.

### 4D EXPLORER

4D Explorer is a drill-down navigation tool for accessing database objects. 4D Explorer servesmultiple functions:

- As a research and investigation tool: 4D provides a preview area for all database objects. This area can be expanded, collapsed, and resized.
- As an interactive dashboard: 4D Explorer provides quick access for immediate object editing and enables the creation of new objects.
- As a point-and-click and drag-and-drop toolbox: 4D Explorer delivers easy access to all 4D objects, including constants, 4D commands and user-defined choice lists. It also ensures interaction between design elements. For instance, you can drag a field from a table to a form or a method, or create many to one and one to many relations by simply dropping a field from one table to another.

#### Tables

- Alphabetical view of tables
- Click: preview on selected table
- Double click: bringing table properties for editing
- Focus on the table inside the global structure and expand fields
- Double click on fields: bringing up field properties for editing or drag-and-drop into other 4D objects.

#### Forms

- Alphabetical view of forms within tables
- Click: Preview of forms for selected table
- Double click: bringing the forms for editing
- Clicking on New brings Form wizard for creation of a new form or reuse of an existing template
- Drag-and-drop form in a method

#### Methods

- Database Methods

  Click: displays the list of the six default database Methods.

  Double- Click allows addition to the default database Methods

- Forms and Triggers Methods

  Click on table: allows viewing the Trigger method or creating it

  Click on a Form: allows viewing the form method or creating it.

- Project Method
- Drag-and-drop methods to associate them with other 4D objects

#### Constants

Almost 500 constants grouped in categories which can be dragged and dropped in methods:

4D Environment, ASCII Code, BLOB, Clipboard, Colors, Communications, Database Engine, Database Events, Date Display Format, Days and Months, Events - Modifiers, Events -What, Expressions, Fields and variable types, Find Windows, Font Styles, Form Events, Functions Keys, Hierarchical Lists, ISO Latin Characters Entities, Open Windows, Picture Display

formats, Platform Interfaces, Platform properties, Process state, Query Destinations, Resources Properties, Screen Depth, Set RGB Colors, Standard System Signatures, TCP Port Numbers, Test Path Name, Time Display format, Window kind.

#### Commands

Almost 600 commands, organized topically, which can be dragged and dropped in methods:

Arrays, ASCII, BLOB, Booleans, Clipboard, Communications, Compiler, Data Entry, Database Methods, Time and date, Debugging, Drag and Drop, Entry Control, Error codes, Form Events, Form Pages, Graphs, Hierarchical Lists, Import and Export, Interruptions, Language, Math, Menus, Messages, Named Selections, Object Properties, OLE Services, On a Series, Pictures

Printing, Process(Communications), Process (User Interface), Processes, Queries, Record Locking, Records, Relations, Resources, Selection, Sets, String, Structure Access, Subrecords, System Documents, System Environment, Table, Transactions, Triggers, User Interface, Users and Groups, Variables, Web Server, Windows

#### Lists

Viewing and editing user-defined choice lists

### PASSWORD EDITOR

Like all the other parts of 4D, the 4D Password editor is in itself a complete environment. The password system allows you to define clear distinctions between designers, administrators, and users, and reflect the exact privileges that groups or individuals have inside the database.

#### Properties

- Specify users of the database
- Create groups of users with different levels of access to the database
- Nest groups of users within other groups to create a hierarchy of users
- Specify the group which owns the objects that each user creates
- Assign a start-up method for each user
- Specify a group owner

#### Scope

After you create access groups, you can fully control access to:

- The Design environment
- Table properties and operations
- Record operations
- Input and output forms
- Methods
- Menu items
- Plug-ins and extensions

#### Within methods, use...

CHANGE ACCESS, CHANGE PASSWORD, Current user, DELETE USER, EDIT ACCESS, GET GROUP LIST, GET GROUP PROPERTIES, GET USER LIST, GET USER PROPERTIES, Is user deleted, SET GROUP PROPERTIES, SET USER PROPERTIES, User in group.

### 4D FORM WIZARD

4D is one of the most elaborate rapid development solutions on the market. With point-and-click ease, the 4D Form Wizard instantly generates a variety of forms allowing immediate access to your data. It also does much more. Unlike most wizards in the industry, 4D form wizard does not simply expect users to answer questions. It also provides a panel of options and techniques optimizing form design and reuse, and lets the user preview the effects of his choice in real time. The 4D Form Wizard acts as an interactive, powerful and customizable form design tool.

#### Tables, fields, template selection

- Form Type (Detail, List, for input/output and printing)
- Template selection (13 default examples, including Web aware templates)

- Saving your own forms as templates.
- Choosing fields from other tables
- Reordering fields
- Grouping fields in a group box and naming the group box

**Styles**

- Styles for each type of object (fields, mandatory fields, unique indexed fields, related fields, group box text, form title, Information, check boxes and radio buttons)
- Label editing: font, font size, font style, style sheets.
- Field editing: font, font size, font style, style sheets.

**Platform interface**

- Automatic, Inherit from form, Copland, Mac OS, Windows 3.1, Windows 95.
- Appearance: None, Plain, Dotted, Raised, Sunken, Double
- Foreground, Background, None

**Options**

- Form size: Width, height, screen sizes (automatic, low resolution windows, 9", 13", 15", 17", 20" Macintosh, PowerBook or Windows)
- Automatic adjustment to fields
- Integrated alert systems if all buttons cannot be placed because of form size.
- Label Location: No Label, In front of fields, Above fields
- Record number/Record count
- Form Title
- Background picture (25 patterns by default)
- Picture Library: allows you to store graphics that you can use as elements in forms, toolbar icons, picture menu items. This picture library enables you to use a graphic in several places while storing it in only one and helps to maintain transparent cross-platform compatibility.
- Associated menu bar
- Automatic multiple pages if the number of fields selected do not fit on one page

**Buttons**

- Choosing a family of buttons (11 per default)
- Viewing the appearance of each family of buttons before using it
- Choosing location for buttons on the form: Top (left, center, right), Bottom (left, center, right), Right side (top, center, bottom), Left side (top, center, bottom).
- Built-in button actions: Cancel, Delete Record, First Page, Last Page, Last record, Next Page, Next Record, Previous Page, Previous Record). You do not need to write a method to specify what happens when the user clicks on the button.

**Subforms**

- Object in the subform
- Form for subform
- Display only, selectable, enterable (to allow data entry in list form), double-clickable (to allow data entry in full page), add subform buttons
- Choosing a form design

### FORM EDITOR

Forms created with the 4D Form Wizard can be used directly or further enhanced using the form editor. The Form editor can also be used to create forms from scratch.

The 4D Form Editor is a complete environment palette capable of competing with advanced drawing packages. Its object-based concept allows point-and-click and drag-and-drop control on all events, objects, fields, variables, graphics, related tables and subforms that may appear on a form. This saves hours of programming efforts and facilitates the reuse of the forms or form templates.

4D allows complete control over each object within a form, ranging from its platform appearance to the execution of scripts attached to that object. Each object is identified by a name, which is automatically assigned by 4D but can be manually overridden.

**General features**

- Grid definition, Grid on/off
- Customizable ruler units (points, centimeters, inches)
- Customizable form scaling ratios (from Macintosh to Windows, from Windows to Macintosh, fixed ratio, rescale pictures).
- Automatic page navigation
- Master page
- Optimized picture management for background picture or picture buttons (using the Picture Library)
- Show/Hide Tools and Objects Palettes
- Intelligent copy and paste includes all object properties and the event methods associated with them.
- Specific menu can be attached to a specific form
- Control lines and markers for output forms, reports and labels

**Form properties**

- Platform Interface: Inherited from database, Automatic, Copland, Mac OS, Windows 3.1, Windows 95
- Access and owner modification privileges
- Form event controls (25 items)
- Automatic or manual window sizing
- Automatic or manual window resizing. Resizing options include: resizable, fixed width, fixed height, maximum width or height, minimum width or height

**Object types**

- Draw objects (rectangle, rounded rectangle, oval, lines)
- Form grid (dialog for number of row and columns)
- Group box (and label)
- Text objects
- Fields and standard variables
- Button
- Highlight button
- 3D Push Button
- Invisible Button
- Picture Button
- Button grid
- Radio Button,
- 3D Radio Button
- Combo box
- Hierarchical list
- Menu/Drop down list
- Check box
- 3D Check box
- Radio Picture
- Tab control
- Pop up
- Hierarchical pop up
- Picture menu
- Scrollable area
- Graph
- Thermometer
- Dial
- Ruler
- Subforms
- Plug-ins or external objects

**Object arrangement**

- Right, Left, Bottom, Top, Center Vertical, Center Horizontal Alignment
- Vertical, Horizontal distribution
- Group/Ungroup
- Move to back/to front
- Smart object duplication
- Customizable tab and entry order control

**Object Properties**

- Object name
- Coordinates
- Resizing and repositioning options: Grow horizontally and/or vertically; move horizontally and/or vertically. Automatic resizing causes the object to grow as the form is enlarged or become smaller as the window is reduced. When automatic repositioning is on, 4D tries to keep the object in view as the user reduces the size of the window. If the user resizes a row of buttons so that some buttons become obscured, automatic repositioning tries to move the buttons so that they remain in view.
- Color: Background/foreground
- Pattern, Border, Line Width
- Platform Interface (Inherited from form, Copland, Mac OS, Windows 3.1, Windows 95
- Appearance: None, Plain, Dotted, Raised, Sunken, Double
- Foreground, Background, None
- Attributes: tabbable, draggable, show focus, droppable
- Print with variable frame
- Display with vertical scrollbar
- Font attributes font, font size, style, style sheet (editable), justification

**Data controls**

- Enterable
- Mandatory
- Auto spellcheck
- Object list
- Object method
- Choice Lists
- Required lists
- Excluded lists
- Minimum values
- Maximum values
- Default values
- Tips and Help Balloons
- Control over which events (17 items) trigger method execution
- Object Method

**Data display**

4D offers a wide variety of display formats for fields. In addition, users can create their own display formats and entry filters as well as display formats and entry filter style sheets.

- Text fields and text variables

  Printing with a variable size frame

  With or without scroll bar display
- Real, Integer, Long integer fields and variables

  Positive, Negative and Null

  18 predefined display formats

  User-definable formats and styles
- Date fields and variables

  Several default display formats

  All international and local calender formats
- Time fields and variables

  Five display formats: HH:MM:SS, HH:MM, Hour Min,

HH:MM AM/PM, HH:MM AM/PM.

All international and local formats

- Booleans fields and variables

  As a pair of radio buttons

  As a check box

  User-definable labels

- Picture fields and variables

  All compression options and color definition

  Cut and Paste formats supported: PICT, EMF, WMF, BITMAP

  Display formats supported: PICT, Quicktime, embedded WMF, embedded EMF

  Using ACI-Pack (free 4D utility product): PICT, BMP, WMF, EMF, JPEG

  Display: Truncated (centered), Truncated (non-centered), Scaled to Fit, Scaled to Fit (proportional), On Background

- Entry filters for numeric, date, time, and alpha fields and variables. Entry filters control the way users are allowed to enter data in 4D objects. For example, an entry filter for telephone numbers will guide data entry and make it easier for fast search on the area code. Entry filters, like any other object property can be overridden programmatically.

**Within methods, use...**

Arrays: ARRAY BOOLEAN, ARRAY DATE, ARRAY INTEGER, ARRAY LONGINT, ARRAY PICTURE, ARRAY POINTER, ARRAY REAL, ARRAY STRING, ARRAY TEXT, ARRAY TO LIST, ARRAY TO SELECTION, COPY ARRAY, DELETE ELEMENT, DISTINCT VALUES, Find

in array. INSERT ELEMENT, LIST TO ARRAY, SELECTION TO ARRAY, Size of array, SORT ARRAY, SUBSELECTION TO ARRAY, Using the element zero of an array

Drag-and-drop: Drag and drop position, DRAG AND DROP PROPERTIES

Form and object events: On Load, On Unload, On Validate, On Display Details, On Activate, On Deactivate, On Outside Call, On Close Box, On Menu Selected, On Open Details, On Close Details, On Clicked, On Double Clicked, On Keystroke, On Getting Focus, On losing Focus, On Data Change, On External Area, On Drop, On Drag Over, On Printing Header, On Printing Details, On Printing Break, On Printing Footer

Form pages: Current form page, FIRST PAGE, Form Pages, GOTO PAGE, LAST PAGE, NEXT PAGE, PREVIOUS PAGE

Hierarchical lists: APPEND TO LIST, CLEAR LIST, Copy list, Count list items, DELETE LIST ITEM, GET LIST ITEM, GET LIST ITEM PROPERTIES, GET LIST PROPERTIES, INSERT LIST ITEM, Is a list, List item parent, List item position, Load list, New list, REDRAW LIST, SAVE LIST, SELECT LIST ITEM, SELECT LIST ITEM BY REFERENCE, Selected list item, SET LIST ITEM, SET LIST ITEM PROPERTIES, SET LIST PROPERTIES, SORT LIST

Object properties: BUTTON TEXT, DISABLE BUTTON, ENABLE BUTTON, FONT, FONT SIZE, FONT STYLE, SET CHOICE LIST, SET COLOR, SET ENTERABLE, SET FILTER, SET FORMAT, SET RGB COLOR, SET VISIBLE

User interface: BEEP, Caps lock down, GET HIGHLIGHT, GET MOUSE, Get platform interface, HIGHLIGHT TEXT, INVERT BACKGROUND, Last object, Macintosh command down, Macintosh control down, Macintosh option down, PLAY, Pop up menu, POST CLICK, POST EVENT, POST KEY, REDRAW, SET CURSOR, SET FIELD TITLES, SET PLATFORM INTERFACE, SET TABLE TITLES, Shift down, Windows Alt down, Windows Ctrl down

Windows: CLOSE WINDOW, DRAG WINDOW, ERASE WINDOW, Find window, Frontmost window, GET WINDOW RECT, Get window title, HIDE TOOLBAR, HIDE WINDOW, MAXIMIZE WINDOW, MINIMIZE WINDOW, Next window, Open external window, Open window, REDRAW WINDOW, SET WINDOW RECT, SET WINDOW TITLE, SHOW TOOLBAR, SHOW WINDOW, Window kind, WINDOW LIST, Window process

## MENU EDITOR

By default, 4D provides a standard set of menus that include functions to perform database operations. Alternatively, menus can be designed and used in a custom environment. The 4D Menu editor provides a simple interface in which you name the menu bar, the menu items, and indicate the name of the method associated with the menu item.

### General specifications

- Access to each menu bar can be controlled by a password.
- Multi-process: each running process can have its own menu bar
- Each form can have its own menu bar.
- Up to 32,767 menu bars
- Automatic inclusion of File and Edit menus
- Menus can be previewed while being designed
- Drag-and-drop rearranging of menus and menu items
- Connected menus: the same menu can be attached to several menu bars. Modifying one instance thus modifies all occurrences in a single step.

### Interface management

- Keyboard equivalents
- Disabling and enabling of menu items
- Separator lines
- Font control
- Style control (bold, italic, underline, outline, shadow). Bold, outline and shadow are displayed on Macintosh only.
- Associated icons with a menu item. The icon is used as a button in the toolbar that is displayed whenever the menu is displayed. Pictures and icons can be dragged and dropped from the Picture Library or copied from the Clipboard.

### Within methods, use...

APPEND MENU ITEM, Count menu items, Count menus, DELETE MENU ITEM, DISABLE MENU ITEM, ENABLE MENU ITEM, Get menu item, Get menu item key, Get menu item mark, Get menu item style, Get menu title, HIDE MENU BAR, INSERT MENU ITEM, MENU BAR, Menu selected, SET ABOUT, SET MENU ITEM, SET MENU ITEM KEY, SET MENU ITEM MARK, SET MENU ITEM STYLE, SHOW MENU BAR

# Programming environment

The 4D programming environment saves development time by delivering a high quality interface and by providing outstanding readability when writing and debugging code. 4D's modular programming capabilities adapt to a wide variety of programmatic needs, which can occur using a powerful relational database for mission critical applications—locally, remotely, and on the Internet. The strength of the 4D language is that it drastically reduces the amount of code involved to perform a task. The 4D language excels in six major ways:

- Smooth integration with 4D graphical environment: 4D language automatically understands what has been graphically designed, thus eliminating redundancy and additional efforts. 4D is a consistent environment where all parts interact naturally.
- Code consistency: The same code, including triggers and procedures, runs 4D as a standalone application, on 4D Client and 4D Server.
- Code reduction: All-inclusive commands deliver a wide range of capabilities at once. For example, with one command (MODIFY SELECTION), you can present a list of records and allow the user to double-click a record for data input. Data is formatted according to the current output form and any appropriate object methods are run when records are displayed. Double-clicking a record opens the current data entry form and allows the user to modify it. Again, any object method is run and any formatting, choice lists, entry filters, or other data entry controls are applied when necessary.
- Code maintenance and reuse: 4D allows designers to create generic routines to automate application programming for easy reuse and modification.
- Openness: 4D open architecture allows seamless blending with any external application.

Using 4D Server, you can perform multi-user development (see above Structure Design)

**METHOD EDITOR**

- Multi-windowing, multi-processing, multi-users
- Access privilege controls
- Customizable source code editor allows separate colors for all the following elements: fields, tables, process variables, methods, commands, externals, errors, comments, local variables, parameters, interprocess variables, constants.
- On-line help provided for each command
- Automatic indentation
- Brace matching
- Go to line
- Find/Replace/Find next
- On-the-fly syntax checking
- Unlimited user comments
- Direct editing of methods from within the Debug window
- Drag-and-drop support: tables, fields, forms, constants and commands from the Explorer
- Point-and-click tables, fields, forms, constants, commands and keywords from inside the Method Editor.
- Copy/Paste all or part of a method
- Object browser list of commands presented alphabetically or topically in hierarchical menus, user-defined, plug-ins and external extensions commands
- Automatic display of 4D keywords and commands in bold and user's defined methods in italics
- Macro-support for commands allows you to type the first few characters or use the wildcard character

## 4D LANGUAGE COMPONENTS

### General characteristics

- Almost 600 high level and low level commands
- Complete event-base model
- If and Case conditional structures
- Three looping structures: For, While, Repeat
- Unlimited methods, functions, and external methods: they are made available as standard 4D objects wherever they could be used.
- Parameter passing which provides the ability to pass a variable or fixed number of parameters to a method
- Full support for recursive code
- Full support for pointers
- Unlimited extendibility: 4D integrates C, C++ and Pascal code within any 4D method
- Application Programming Interface (API): 4D Open
- Unlimited interoperability. DLL, DDE, OLE, ODBC and AppleScript support, 4D Plug-ins and Third Party Extensions

### 4D methods

4D has five method types. Each method type manages a specific part of the database. The technique for writing each method type is identical, thus providing power users and developers with coherent code throughout the application.

- Object methods

  An object method is attached to an object in a form. When the object is copied, the object method bound to the object is automatically copied with it. The method is automatically invoked by 4D when the object is used. Objects methods are triggered by the users actions. Each of these events can be filtered to control when the method should be executed.

- Form methods

  A form method is attached to a form and monitors the use and behavior of that form. The method is invoked automatically by 4D when the form is used. Form events can be filtered to control when the method should be executed.

- Table methods and triggers

  A table method is attached to a table. Triggers are invoked automatically when you manipulate the records of the table.

- Project methods

  A Project method is independent from any object, form or table. Project methods are reusable and available for use for any other method. For instance, instead of writing the same method several time for several objects performing the same action, you can simply write a generic project method and refer to that project method as an attribute for these objects.

- Database methods

  Database methods are associated with the application as a whole and are automatically executed by 4D when you open the database. Six database methods are automatically available: On Startup Database, On Exit Database, On Web Connection, On Server Startup, On Server Shutdown, On Server Open Connection, On Server Close Connection.

### Variables

- Process
- Interprocess
- Local variables
- One and two dimensional arrays
- Pointers

### Operators

- Bitwise Operators: and, or (inclusive), or (exclusive), left bit shift, right bit shift, bit set, bit clear, bit test
- Comparison operators:

  For strings: Numeric, Date, Time: equality, inequality, greater than, less than, greater than or equal, less than or equal

  For pointers: Equality, inequality
- Date operators: Date difference, day addition, day subtraction
- Logical operators: And, or
- Numeric operators: Addition, subtraction, multiplication, division, longint division, modulo, exponentiation
- Picture Operators: horizontal concatenation, vertical concatenation, exclusive superimposition, inclusive superimposition, horizontal move, vertical move, resizing, horizontal scaling, vertical scaling
- String Operators: concatenation, repetition
- Time Operators: addition (Time+Time), addition (Time+Number), subtraction (Time-Time), subtraction (Time-Number), multiplication, division, longint division, modulo.

## 4D DEBUGGER

4D Debugger has been designed to be used easily by both beginners and expert programmers. It is a complete point-and-click, drag-and-drop graphical environment for code analysis and correction. 4D Debugger can be displayed from any error window and can be programmatically invoked using the TRACE command at the beginning of your routines.

4D Debugger can be called at anytime. It fully supports the 4D multi-processing architecture by maintaining a separate debugger for each individual process.

### Control toolbar buttons and information

- Name of the method currently being debugged
- The reason the debugger was invoked
- Step out
- Step into process
- Step into
- Step over
- Edit
- Abort and Edit
- Abort
- No trace

### Resizable panes

The debugger window is fully resizable and allows the programmer to easily resize each pane within the debugger. All columns within panes are also resizable.

- Watch
- Call chain
- Custom watch
- Source code

### Watch pane

- General information about the system, the 4D environment and execution environment
- Expression column, displaying the names of the objects or expressions
- Value column displaying the current value of the corresponding object or expression
- Hierarchical lists displaying tools to modify the value: information, line objects, variables (process, interprocess, local), parameters, self pointer, constants, fields, semaphores, sets, processes, named selections.
- Speed menu for: Collapse all, Expand all, Show field and table numbers, Show types, Show icons, Sorted tables and fields, Show integers in hexadecimal

**Call chain pane**

- Hierarchical list for easy tracking and control over embedded methods
- Double-clicking on the name of a method displays source code for the method
- Clicking the node next to the method's name expands or collapses parameter values and function result

**Custom watch pane**

- For easy evaluation of any expression, including any object returning a value.
- Inserting a new expression by drag-and-drop from the Watch pane or the Call Chain pane, or point-an-click in the Source code pane
- Speed menu for: Collapse all, Expand all, Insert command, Delete all, Show field and table numbers, Show types, Show icons, Sorted tables and fields, Show integers in hexadecimal

**Source code pane**

- Displaying the source code of the method being traced
- Program counter indicating the line on the verge of being executed
- All object values can be easily viewed by placing the cursor over that object.
- Break Point Properties Window includes:

  Location

  Type: Persistent, temporary

  Break when following expression is true

  Number of times to skip before breaking

  Break point is disabled
- Break List Window: For managing break points and creating additional breaks to your code by catching call to 4D commands

**Additional 4D debugging tools**

- Syntax Error Window: Debugging an application can also be achieved from within the application before it is compiled. In case of an error, a Syntax Error window displays a message indicating the type of error found. Using the buttons of the Syntax Error window, developers can either halt procedure execution or continue procedure execution. Thus, developers can skip any portion that cannot be executed, automatically display the debug window, or directly open the Method editor to immediately correct the error.
- Plug-ins debugging tools: When programming plug-ins using 4D for ORACLE, 4D SQL Server, 4D ODBC, and 4D Write, developers can benefit from the internal 4D debuggers and from the additional debuggers with those plug-ins.
- 4D Compiler: 4D Compiler includes a complete error description system for fast correction on compiled code. See White Paper on 4D Compiler.

**Within a method, use...**

Language: Command name, Count parameters, EXECUTE, Getpointer, Is a variable, Nil, NO TRACE, RESOLVE, POINTER, Self, TRACE, Type

Interruptions: FILTER EVENT, ON ERR CALL, ON EVENT CALL

Resources: ARRAY TO STRING LIST, CLOSE RESOURCE FILE, Create resource file, DELETE RESOURCE, GET ICON RESOURCE, Get indexed string, GET PICTURE RESOURCE, GET RESOURCE, Get resource name, Get resource properties, Get string resource, Get text resource, Open resource file, RESOURCE LIST, RESOURCE TYPE LIST, SET PICTURE RESOURCE, SET RESOURCE, SET RESOURCE NAME, SET RESOURCE, PROPERTIES, SET STRING RESOURCE, SET TEXT RESOURCE, STRING LIST TO ARRAY.

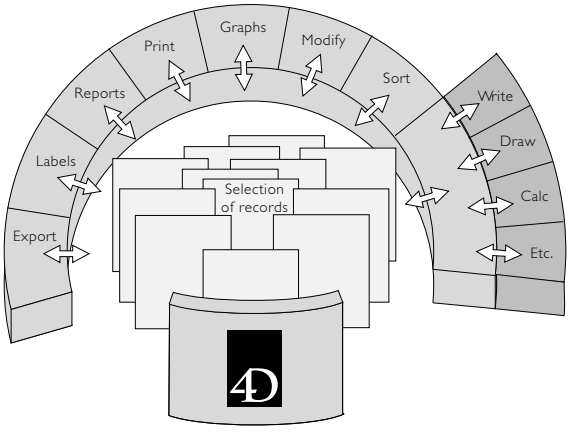
Variables: CLEAR VARIABLE, LOAD VARIABLES, SAVE VARIABLES, Undefined

Compiler: C_BLOB, C_BOOLEAN, C_DATE, C_GRAPH, C_INTEGER, C_LONGINT, C_PICTURE, C_POINTER, C_REAL, C_STRING, C_TEXT, C_TIME, IDLE

# Information management

## DATA ACCESS AND NAVIGATION: 4D CONCEPT OF CURRENT SELECTION

When a query is processed, the resulting selection of records is maintained automatically and treated as a database object. This selection of records is referred to as the Current Selection. Because current selections are maintained by 4D, the results from a single query can be used in a variety of operations without requiring the query to be re-executed. For example, the same selection can be used to print reports & labels, display charts, export data or even as the basis for subqueries.




### Current selection and user interface

- Multiple operations on a single selection (i.e., reports, labels, charts, exports, mail merge, global update, etc.)
- Live data from a current selection can be displayed on-screen as a list. Double-clicking can display detailed information or enable data modification in a multi-level, drill-down technique.
- When a current selection is displayed on-screen as a list, records can be selected manually by the user. Single, continuous and random record selection is available.
- Current selections can be traversed in a variety of manners. For example, navigation within a selection can be implemented using standard 4D or user-defined navigation controls such as First Record, Last Record, Previous Record, and Next Record buttons.

### Current selections in multi-process and multi-user situations

- 4D automatically maintains a current selection for each table/process/user combination. This allows each process for each user to manipulate data independently and concurrently. Current selections are maintained whether the user is connected to a browser or a 4D client.
- 4D provides automatic record locking for the current record in a selection. Once the record is released, it becomes available for modification from any other process or user. The 4D current selection/current record management avoids the problems of database replications and synchronization changes in multi-user and multi-processing situations.

### Within methods, use...

Selection: ALL RECORDS, APPLY TO SELECTION, Before selection, DELETE SELECTION, DISPLAY SELECTION, End selection, FIRST RECORD, GOTO SELECTED RECORD, LAST RECORD, MODIFY SELECTION, NEXT RECORD, ONE RECORD SELECT, PREVIOUS RECORD, Records in selection, REDUCE SELECTION, SCAN INDEX, Selected record number.

Sets: ADD TO SET, CLEAR SET, COPY SET, CREATE EMPTY SET, CREATE SET, DIFFERENCE, INTERSECTION, Is in set, LOAD SET, Records in set, REMOVE FROM SET, SAVE SET, Sets, UNION, USE SET.

Named selections: CLEAR NAMED SELECTION, COPY NAMED SELECTION, CUT NAMED SELECTION, USE NAMED SELECTION.

Subrecords: ALL SUBRECORDS, APPLY TO SUBSELECTION, Before subselection, CREATE SUBRECORD, DELETE SUBRECORD, End subselection, FIRST SUBRECORD, LAST SUBRECORD, NEXT SUBRECORD, ORDER SUBRECORDS BY, PREVIOUS SUBRECORD, QUERY SUBRECORDS, Records in subselection.

## QUERY AND SORT

When a query is executed, 4th Dimension automatically uses indexes if present. In compound queries on mixed indexed and non-indexed fields, 4th Dimension searches the indexed fields before the non-indexed fields thus providing optimal performance. Queries are not case sensitive by default and can understand foreign language diacritical marks where necessary.

4D's built-in general query and sort editors can be used directly or included in a fully customized application using only one high level command.

### General Query Editor

- Compound search criteria
- Relational searches
- Automatic choice list display
- Subtable searches
- Indexed fields shown in bold
- Saving and loading of the search expression
- Optimized use of indexes
- Query within the current selection
- Three conjunctions: and/or/except
- Eight comparison operators: equal to, not equal to, greater than, greater than or equal to, less than, less than or equal to, contains, does not contain

### Query by example

- Uses current data entry form
- Can be multiple pages
- Uses choice lists
- All comparison operators are available
- Criteria can be viewed in the General Query editor

### Query and modify

- Extremely fast search method
- Query on indexed fields only
- Immediate display of data entry form
- Multiple pages available

### Query by formula

- Entire programming language at the user's disposal
- Can use any valid boolean expression including:

  4D functions, user defined functions, external functions, etc.
- Saving and loading of query expressions

### Order-By Editor

- Up to 30 levels
- Ascending and descending order per level
- Displays field list and sort area
- Each level can be a field or a formula.
- Formulas can be edited in the Formula Editor

### Within methods, use...

ORDER BY, ORDER BY FORMULA, QUERY, QUERY BY EXAMPLE, QUERY BY FORMULA, QUERY SELECTION, QUERY SELECTION BY FORMULA, SET QUERY DESTINATION, SET QUERY LIMIT

## DATA INPUT AND OUTPUT

- Edit data in both list and detail forms
- Built-in and user defined data controls
- Relational data entry and modification (one to many, many to one and many to many) across tables, or through embedded subtables.
- Referential data integrity control for data entry, modification, and deletion.
- Global update using the Formula Editor which provides all the tools needed to perform a global update on a selection of records (editing area, operators toolbar, comparison operators, braces, quotes, tables drop-down list, fields list, command list, save and load formula button).

- Multi-processing compatible with data entry: data can be edited in one process while other processes perform operations such as importing, generating reports, or creating new indexes.
- Automatic record locking and record release during multi-process /multi-user data entry
- Print a single record or a selection of records
- Mail merge (with 4D alone or using 4D Write)
- Import/export (see below)
- Graph, labels, quick reports, etc. (See data access and navigation, below.)

**Interacting with peripherals for input and output**

- Bar code readers (using either the serial port or the ADB port)
- Fax machines (using 4D commands or third-party plug-ins)
- Card readers, magnetic strips, credit cards, and name badges (using the 4D built-in serial port commands)
- Optical character readers: 4D accepts data from a scanner, processes the results, and produces PostScript compatible output.
- Telephone systems (using third-party plug-ins)
- Touch screens (acts as a keyboard)
- Data entry via voice recognition (using third-party plug-ins)
- Multi-media input and output systems
- Analog input/output devices: I/O analog interfaces and IEEE type cards allow 4D to be used with industrial computing systems. 4D can control or access information from a wide variety of machine tools and measurement instruments. This capability is used in many areas, including controlling oil-drilling platforms, manipulating television antennas, and managing medical instruments.

**Interacting with SQL databases and ODBC-compliant products**

Thanks to its multi-processes architecture and its multiple network management capabilities, 4D can perform simultaneous queries on various servers, merge and display the results to the user with a unified interface either inside 4D Client or a Web browser accessing 4D. Use the following 4D Plug-ins:

- 4D SQL Server
- 4D for ORACLE
- 4D ODBC

**Interacting with legacy systems and other devices**

- Legacy systems: Using 3rd party plug-ins
- Connects with a Newton or other PDA using third-party plug-ins

**Interacting with other desktop applications**

- With any ODBC compliant product using 4D ODBC
- With any application using 4D commands, common scripting languages such as AppleScript, standard inter- application communication layers or customized third-party extensions calling 4D Engine.
- Mail services (MS Mail, QuickMail and others, using third-party plug-ins)
- On-line information network services
- Excel, Word, Quark Express, Frame Maker, PageMaker, etc.

**Within methods, use...**

Data entry: ADD RECORD, ADD SUBRECORD, DIALOG, Modified, MODIFY RECORD, MODIFY SUBRECORD, OId.

Records: CREATE RECORD, DELETE RECORD, DISPLAY RECORD, DUPLICATE RECORD, GOTO RECORD, Modified record, POP RECORD, PUSH RECORD, Record number, Records in table, SAVE RECORD, Sequence number.

Subrecords: ALL SUBRECORDS, APPLY TO SUBSELECTION. Before subselection, CREATE SUBRECORD, DELETE SUBRECORD, End subselection, FIRST SUBRECORD, LAST SUBRECORD, NEXT SUBRECORD, ORDER SUBRECORDS BY, PREVIOUS SUBRECORD, QUERY SUBRECORDS, Records in subselection.

Record locking: LOAD RECORD, Locked, LOCKED ATTRIBUTES, READ ONLY, Read only state, READ WRITE, UNLOAD RECORD.

Transactions: CANCEL TRANSACTION, In transaction, START TRANSACTION, VALIDATE TRANSACTION.

Entry control: ACCEPT, CANCEL, FILTER KEYSTROKE, GOTO AREA, Keystroke, REJECT.

Messages: ALERT, CONFIRM, GOTO XY, MESSAGE, MESSAGES OFF, MESSAGES ON, Request.

Communications: RECEIVE BUFFER, RECEIVE PACKET, RECEIVE RECORD, RECEIVE VARIABLE, SEND PACKET, SEND RECORD, SEND VARIABLE, SET CHANNEL, SET TIMEOUT, USE ASCII MAP OLE.

Services: OLE_DEL OBJECT, OLE_EXEC ACTION, OLE_INS DIALOG, OLE_Insert file.

Printing: ACCUMULATE, BREAK LEVEL, Level, PAGE BREAK, PAGE SETUP, PRINT FORM, PRINT LABEL, PRINT RECORD, PRINT SELECTION, PRINT SETTINGS, Printing page, REPORT, SET PRINT PREVIEW, Subtotal

## IMPORT/EXPORT

4D's built-in import/export dialogs can be used directly or be included in a fully customized application using just one high level command.

### ASCII map editor

ASCII maps are used during importing, exporting, or serial communications whenever the base character set is different between machines (i.e. EBCDIC to ASCII conversions).

- Input or Output ASCII Map
- Save/Load ASCII maps

### Import editor

With 4D's multi-processing environment, importing and exporting can take place in the background using a separate process.

- Graphical interface for selecting the fields to import
- User control over field and record delimiters
- Input form selection allows importing using a 4D form
- ASCII map selection
- SYLK, DIF, Text import
- dBase: import data or directly create a new table within 4D
- Automatic indexing of imported data

### Export editor

- Graphical interface for selecting the fields to export
- User control over field and record delimiters
- Format (text, DIF, SYLK)
- Add line feed
- SYLK, DIF, text
- ASCII map selection

### Within methods, use...

EXPORT DIF, EXPORT SYLK, EXPORT TEXT, IMPORT DIF, IMPORT SYLK, IMPORT TEXT

## LABEL WIZARD

The 4D Label Wizard is built-into the 4D user environment. The Label Wizard is also available in custom applications by calling the command PRINT LABEL.

Note: You can create even more sophisticated labels using the 4D form editor.

### Label design

- Label design preview area
- Drag-and-drop fields from the current table and related tables
- Static text entries
- Static graphical objects (rectangle, rounded rectangle, oval, lines)
- Static graphics
- Background and foreground colors
- Patterns, borders, line width
- Field formatting
- Font, style, size, justification
- Right, Left, Bottom, Top, Center Vertical, Center Horizontal Alignment
- Vertical and horizontal distribution.
- Duplicate and object layering
- Form list lets you bypass the Label Wizard and use a form to print labels.
- Automatic handling of blank fields and lines

### Labels page

- Label page preview area
- Orientation and label order buttons
- First label placement
- Label across and down boxes
- Label and page size control
- Margin control
- Auto resize
- Horizontal and vertical gap control
- Unit control (points, millimeters, centimeters, inches)
- Multiple labels per record lets you print more than one copy of each label.
- Standard page control lets you choose the page dimensions and margins for standard commercial label paper
- Method entry lets you choose a method to run at print time
- Save and then later reload label designs

## QUICK REPORT EDITOR

To create sophisticated non-columnar reports including subforms and embedded graphs, use the 4D form editor. For simpler tabular reports, use the Quick Report editor. The Quick Report editor is available in the user environment or in a custom application through the call of a single command.

- Point-and-click, drag-and-drop report definition
- Loading and saving a Quick Report definitions
- Columns display fields and formulas, either from the current table or from related tables.
- Display formats for fields
- Access to the Formula Editor
- Automatic summary calculations (totals, subtotals, minimum, maximum, count, average)
- Up to ten sort levels. Ascending and descending order
- Control of fonts, size, style, and justification for each cell and text attributes
- Control automatic widths, repeated values, and column sorting on a per column basis.
- Columns indicators between columns
- Current page width is clearly indicated
- Header, detail, break and total rows
- Configurable page headers and footers including current time, current date, and page number
- Report framing
- User-definable formulas
- Print to a printer, a disk file, or to a graph. Can be used as an ad-hoc export system.

## GRAPHS

4D includes an integrated graphing system which allow you to create graphs directly from your data without using other products. Graphs can be plotted using fields and/or variables. You can use the 4D Chart wizard in the user environment or within a custom application.

### Graph display

- Displayed in the current window or in a separate window
- Can be displayed in a data entry form where values within a subtable can be presented as chart. Consequently, each record can contain a different graph or graphs.
- Drag-and-drop fields from current table or related tables
- Adding references to other 4D fields and calculated expressions
- Creating hot links with other 4D plug-ins (4D Write, 4D Calc, 4D Draw)
- Includes three-dimensional charting

### Two-dimensional graphs

- Area: 6 styles and 3 options (stacked, stacked proportional, horizontal)
- Column: 8 styles and 5 options (stacked, stacked proportional, horizontal, overlap, gap width)
- Line: 3 styles and 2 options (stacked, horizontal)
- Pie: option Start Angle, exploding wedges
- Picture: 3 options (stacked, stacked proportional, horizontal, overlap, gap width)
- Polar
- Modifying depth on two-dimensional graph

### Three-dimensional graphs

- Column: 2 styles and 3 options (category gap/width, series gap/width, tops only)
- Line: option series gap/width
- Area: option series gap/width
- Surface: 2 styles and option tops only
- Triangle: 2styles and 3 options (series gap/width, flipped, plot zero values)
- Spikes: 2 styles and option oval/square heads
- Changing perspective on a three-dimensional graph

### Graph customization

- Add static graphical objects (rectangle, rounded rectangle, oval, lines, polygons, arrowheads)
- Static text
- Object alignment aids (Right, Left, Bottom, Top, Center Vertical, Center Horizontal Alignment, Vertical, Horizontal distribution, Group/ungroup, Move to back/to front, Duplicate)
- Document size (width and height)
- Rulers (points, centimeters, inches)
- Copy/Cut/Paste/Duplicate
- Graph resizing
- Axes label customization (label positions, orientation, formats)
- Tick mark customization
- Scale control of the values axis
- Reversing the order of data points on an axis
- Positioning the origin, on the Y axis, on the X axis
- Axis title control
- Show/hide grid lines
- Legend customization: hide/display, position, legend order, text
- Attributes manipulations on chart objects

### Within methods, use...

GRAPH, GRAPH SETTINGS, GRAPH TABLE, 4D Chart commands

ACI US, INC. 20883 STEVENS CREEK BLVD. CUPERTINO, CA 95014
TEL 408.252.4444 FAX 408.252.0831 HTTP://WWW.ACIUS.COM