

VisualAge Pacbase



The Metamodel and the Repository

Version 3.0

Note

Before using this document, read the general information under "Notices" on page v.

According to your licence agreement, you may consult or download the complete up-to-date collection of the VisualAge Pacbase documentation from the VisualAge Pacbase Support Center at:

<http://www.ibm.com/software/ad/vapacbase/support.htm>

Consult the Catalog section in the Documentation home page to make sure you have the most recent edition of this document.

Second Edition (January 2002)

This edition applies to the following licensed programs:

- VisualAge Pacbase Version 3.0

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at: <http://www.ibm.com/software/ad/vapacbase/support.htm> or to the following postal address:

IBM Paris Laboratory
1, place Jean-Baptiste Clément
93881 Noisy-le-Grand, France.

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

© Copyright International Business Machines Corporation 1983,2002. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices v

Trademarks. vii

Chapter 1. Introduction 1

Purpose of the Manual 1

Goals and Advantages of VisualAge Pacbase 1

Chapter 2. The Repository Structure . . . 3

Repository Definition 3

Specifications Database 3

Libraries. 3

Sessions 4

Chapter 3. The Metamodel 5

Metamodel Definition 5

Composition 5

Design and Development Metamodels. 5

Administration Metamodel 8

Re-use Principle 9

Cross-referencing Principle 9

Data Consistency 9

Metamodel Extensibility 9

Documentation Principle 10

Chapter 4. Services 11

Search for Instances. 11

Search by Keywords 11

Search by Lists 11

Search by cross-references 12

Metamodel's On-Line Help 12

Journalization of Updates. 12

Access Security 13

Library Level. 13

Instance Level 13

Notices

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

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk NY 10504-1785, U.S.A.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Paris Laboratory, SMC Department, 1 place J.B.Clément, 93881 Noisy-Le-Grand Cedex. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

IBM may change this publication, the product described herein, or both.

Trademarks

IBM is a trademark of International Business Machines Corporation, Inc. AIX, AS/400, CICS, CICS/MVS, CICS/VSE, COBOL/2, DB2, IMS, MQSeries, OS/2, PACBASE, RACF, RS/6000, SQL/DS, TeamConnection, and VisualAge are trademarks of International Business Machines Corporation, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and/or other countries licensed exclusively through X/Open Company Limited.

All other company, product, and service names may be trademarks of their respective owners.

Chapter 1. Introduction

Purpose of the Manual

This manual describes the basic principles of the Repository and the Metamodel.

Their implementation, which can be performed via the Administrator/Developer workbench, the VisualAge Pacbase WorkStation or the character-mode interface, is not documented in this Manual. Please refer to the on-line help of the Administrator/Developer workbench, to the 'VisualAge Pacbase WorkStation User Interface' Guide or to the 'Character-Mode User Interface' Guide.

Goals and Advantages of VisualAge Pacbase

VisualAge Pacbase (also abridged as VA Pac) enables you to design and develop new applications as well as maintain already existing applications.

These applications can be:

- eBusiness applications,
- On-line applications,
- Batch applications.

With VisualAge Pacbase, several development tools are integrated into a single package.

VA Pac provides the development environment for applications which must be:

- Shared: distribution of processes and data;
- User responsive: easy to use and to modify;
- Portable: independent of target environment, and adaptable to new technology.

VisualAge Pacbase assists everyone involved in development:

- Analysts,
- Programmers,
- Administrators,
- Project leaders.

Chapter 2. The Repository Structure

Repository Definition

The Repository is the sole storage location of the data handled during the design, development and integration processes.

Moreover the Repository features facilities to organize, maintain, and secure this data.

Specifications Database

The Repository concept materializes through the existence of Specifications Databases.

These Databases store all the data handled throughout the life cycle of applications.

Several Databases can co-exist in a company. For example a Database can be dedicated to one of the company's activities or department...

Databases are managed by an Administrator who must see to their consistencies and updates.

Libraries

A VA Pac Database is structured in Libraries.

Libraries are hierarchically structured on a maximum of nine levels. The hierarchical structure is determined when Libraries are created and can only be modified by the deletion or creation of a Library.

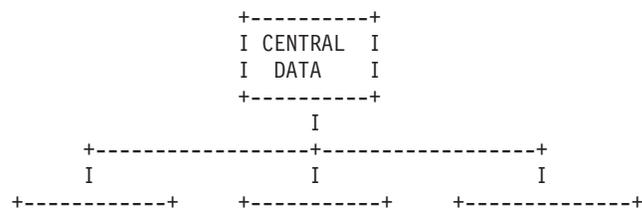
The hierarchical level of a library corresponds to the general nature of the information it contains.

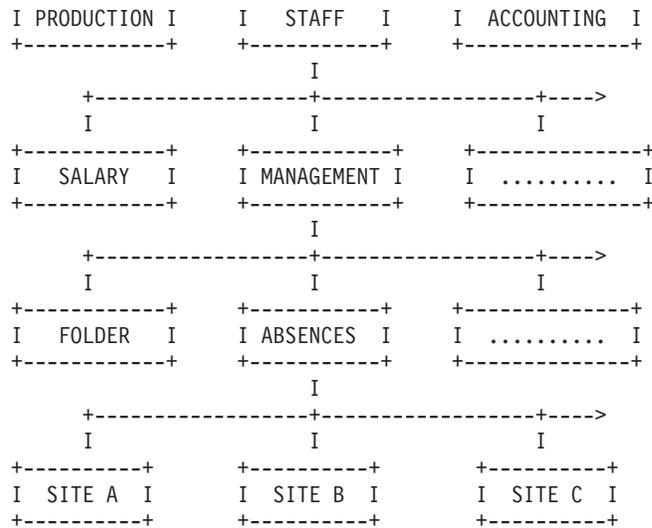
A specification is described in the user's connection Library but also in the higher Libraries of which this Library is dependent.

Any modification in a given Library immediately impacts all its dependent Libraries.

VA Pac shows the user the elements which belong to the selected Library, as well as those which belong to the higher level Libraries and, optionally, those which belong to the lower Libraries.

Example of the Libraries Hierarchy in a Database





Sessions

Sessions make up the time scale of the Database. Data is dated via them.

The principle of sessions makes it possible to manage various versions of the same application: the Administrator freezes the Database, creating a snapshot of the current session.

The result is the creation of two types of frozen sessions:

- the initial frozen session for the application as it was when it was put into production. This session can be accessed in read mode only and then never changes.
- the test frozen session, dedicated to the maintenance of the application put into production. This session can then be accessed in read/write mode.

With VA Pac, this versioning does not entail any physical duplication of specifications. So the source of your production applications is always up-to-date and accessible in real time.

Sessions are represented by four-digit numbers, the value being 0001 at the time the Database is created. This value is incremented daily, when the first user logs on to the Database.

However for a better identification of sessions, frozen sessions can be assigned names.

During maintenance, you can:

- access the initial session at any time,
- perform corrections in the test session,
- request the transfer of data from the test session to the current session or to another test session.

You can display the discrepancies between two sessions of a Database. This consultation mode enables you to compare the lines from a frozen session against those from another session. After comparing these lines, you can align the frozen session with the reference session, if necessary.

Chapter 3. The Metamodel

Metamodel Definition

The VA Pac Metamodel defines the structure of the data involved in the logical description of applications.

It reduces as much as possible the recognition of the physical requirements during the specification phase. As a result the software architecture is chosen at generation time only.

The VA Pac Metamodel makes it possible to describe applications in a non-redundant, systematic, and exhaustive way.

Moreover it enables you to analyze the impact of any modification before making it. This makes the maintenance easier.

Before using VisualAge Pacbase, you must then master its Metamodel.

Composition

The Metamodel is composed of entities, which are typed, structured and reusable elements.

Entities are grouped to constitute applications.

There are several Metamodels:

- two Metamodels accessible to all users:
 - the design Metamodel,
 - the development Metamodel.
- a Metamodel accessible to Administrators only:
 - the administration Metamodel.

Design and Development Metamodels

Together, these Metamodels cover the whole life-cycle of applications.

The Design Metamodel

The design Metamodel groups the entities managed by the design methodology in use as well as the relations between these entities (and their cardinalities in some cases).

It can be used only via the Pacdesign module of the VA Pac WorkStation.

Please refer to the on-line help of the design methodology in use.

Refer also to the 'VisualAge Pacbase WorkStation User Interface' Guide for a detailed description of the interface.

The Development Metamodel

The entities which belong to the Development Metamodel are grouped according to their function.

- DESCRIPTION OF DATA.

- DATA ELEMENT

This entity is used to identify and describe all the basic units of information used by the applications during at least one stage in the development of a project.

For more information, see the on-line help of the Developer workbench or otherwise the 'Data Dictionary' Manual.

- SEGMENT

This entity is an ordered and structured list of Data Elements.

For more information, see the on-line help of the Developer workbench, or the 'Data Dictionary' Manual.

- LOGICAL VIEW

This entity is an ordered and structured list of Data Elements, in an eBusiness application.

For more information, see the on-line help of the Developer workbench, the 'Developer's Documentation - eBusiness Applications series: Business Logics' for the VA Pac WorkStation interface or the 'Pacbench C/S: Business Logic and TUI Clients' for the character-mode interface.

- DATA STRUCTURE

This entity groups a logical set of Segments.

For more information, see the on-line help of the Developer workbench, or the 'Data Dictionary' Manual.

- DATABASE BLOCK

This entity is used to logically and physically describe hierarchical, relational or network Databases.

For more information, see the on-line help of the Developer workbench, or the 'Data Dictionary' Manual.

- DOCUMENTATION

- TEXT

This entity is a narrative presentation of information, which describes and documents applications during their entire life cycles.

For more information, see the 'VA Pac WorkStation User Interface' Guide or the 'Character Mode User Interface' Guide.

- DOCUMENT (VA Pac WorkStation and character-mode interface)

This entity enables you to create personalized documentation (long-range plans, analysis reports, functional and technical specifications...) by calling instances managed in the Specifications Database.

For more information, see the 'Personalized Documentation Manager' Manual.

WARNING The USER MANUAL entity is no longer maintained nor documented. For information, contact the technical support.

- DOCUMENT (AD workbench)

This entity is entirely new in VA Pac 3.0. It enables you to edit instances' lists and descriptions.

For more information, see the on-line help of the Developer workbench.

- PARAMETERIZED INPUT AID (P.I.A.)

This entity is used to standardize the input in the instances's comments, Database Block descriptions...

For more information, see the on-line help of the Administrator/Developer workbench or the 'Character Mode User Interface' Manual.

- EBUSINESS APPLICATIONS DEVELOPMENT

- ELEMENTARY COMPONENT.

This entity supports the processing of applications developed with the eBusiness module of the Developer workbench.

For more information, see the on-line help of the Developer workbench, the 'Developer's Documentation - eBusiness Applications series: Business Logics' for the VA Pac WorkStation interface or the 'Pacbench C/S: Business Logic and TUI Clients' for the character-mode interface.

- FOLDER.

This entity groups Logical Views associated with Elementary Components.

For more information, see the on-line help of the Developer workbench, the 'Developer's Documentation - eBusiness Applications series: Business Logics' for the VA Pac WorkStation interface or the 'Pacbench C/S: Business Logic and TUI Clients' for the character-mode interface.

- APPLICATION.

This entity, which is available via the Developer workbench only, describes the technical components required for the correct operation of the application.

For more information, see the on-line help of the Developer workbench.

- COMMUNICATION MONITOR.

This entity describes the communication between the client and the server used by the application.

For more information, see the on-line help of the Developer workbench, the 'Developer's Documentation - eBusiness Applications series: Business Logics' for the VA Pac WorkStation interface or the 'Pacbench C/S: Business Logic and TUI Clients' for the character-mode interface.

- INITIALIZATION - TERMINATION MONITOR.

This entity implements specific processing before and after the execution of a request associated with a Folder.

For more information, see the on-line help of the Developer workbench, or the 'Developer's Documentation - eBusiness Applications series: Business Logics' for the VA Pac WorkStation interface.

- BATCH APPLICATIONS DEVELOPMENT

- PROGRAM and MACRO-STRUCTURE.

These entities support the processing of batch applications.

For more information, refer to the on-line help of the Developer workbench and to the 'Batch Applications' Manual.

- REPORT

This entity is used to describe the reports output by the batch procedures.

For more information, refer to the on-line help of the Developer workbench and to the 'Batch Applications' Manual.

- ON-LINE SYSTEMS DEVELOPMENT

- SCREEN.

This entity supports the processing of applications developed with the OLSD function.

For more information, see the 'On-Line Systems Development' Manual.

- DIALOG

This entity groups a logical family of Screens and enables you to specify options common to all the components associated with the Dialog.

For more information, see the 'On-Line Systems Development' Manual.

Moreover you can customize this Metamodel by defining Meta Entities, User Entities and User Relations (refer to the on-line help of the Administrator workbench, or to the 'Dictionary Extensibility' Manual for the character-mode interface).

Administration Metamodel

This Metamodel, which operates in a single session and Library, can be accessed by Administrators only, via a dedicated Workstation. It contains the following entities, which are documented in the on-line help of the Administrator workbench:

- BASE.

This entity is used to define the Databases managed on site.

- PROFILE.

This entity is used to define initial parameters (authorizations...). valid when the user uses the product.

- USER.

This entity is used to:

- customize the authorizations defined in the user's profile,
- assign additional data and parameters specific to the user,
- define the generation-print commands specific to the user.

- COMMAND LINES.

This entity is used to define the commands before/after the job for the generation procedure.

- PARAMETERS.

This entity is used to specify the VA Pac parameters tables:

- Error messages,
- Text labels,
- Special characters.

- SECURITY.

This entity is used to define the site's security systems and VA Pac access keys.

- PAC/TRANSFER

This entity is used to enter the input of the PacTransfer module.

- META ENTITIES and USER RELATIONS

These entities are used to customize the Metamodel.

- LIBRARIES.

This entity is used to indicate the characteristics of a Library.

- CONFIGURATION ENVIRONMENTS

This entity is used to indicate physical targets for the generation procedure.

- INPUT AIDS

This entity is used to standardize the input in the entities's comments, Database Block descriptions...

- DOCUMENTS

This entity is entirely new in VA Pac 3.0. It enables you to edit instances' lists and descriptions.

Re-use Principle

An instance which already exists in the Repository can be used by other instances as many times as necessary, without being redefined.

For example, a Data Element, which is defined only once, can be used in the description of one or more Programs or Segments, or in one or more Screen Maps.

This principle:

- saves time in the development stage,
 - decreases the chances of errors,
 - simplifies the maintenance stage since there is only one check point.
-

Cross-referencing Principle

The same instance can be used as many times as necessary by other instances. So links between used instances and using instances are automatically, systematically and immediately created as soon as an instance is called in another one.

These cross-references are then automatically updated in real time.

This principle:

- Enables you to have an overall view of the Metamodel,
 - Enables you to assess, via a search, all the impacts that a modification is bound to entail,
 - forbids the deletion of an instance which is linked to another one.
-

Data Consistency

Data consistency is guaranteed because:

- two instances of the same entity cannot have the same code (in the same Library),
 - an instance cannot be deleted if it is used by other instances, Refer to subchapter 'Cross-Referencing Principle'.
 - read/write access can be reserved on certain instances or Libraries/sessions to certain users. Refer to chapter 'Services', subchapter 'Access Security'.
-

Metamodel Extensibility

VisualAge Pacbase manages standard entities which represent specific concepts in the normal life-cycle of a project.

However customized entities can be defined and their links with standard or other customized entities can be indicated.

The VisualAge Pacbase Metamodel can then be customized to meet the specific needs of a company.

For details, refer to the on-line help of the Administrator workbench.

Documentation Principle

The Documentation function enables you to document all the information contained in the Repository via:

- COMMENTS.

Refer to the on-line help of the Administrator/Developer workbench, to the 'VA Pac WorkStation User Interface' Guide or to the 'Character Mode User Interface' Guide.

- TEXTS which explain the functional aspects of applications.

The input of text in the VA Pac WorkStation and in the character-mode interface is very different, although compatible, because the WorkStation offers the interface of a text processing.

Refer to the 'VA Pac WorkStation User Interface' Guide or to the 'Character-Mode User Interface' Guide, as well as to the 'Personalized Documentation Manager' Manual.

- INPUT AIDS, which are preformatted input masks defined according to the specific needs of applications. These Input Aids constitute documentation standards.

Refer to the on-line help of the Administrator/Developer workbench or to the 'Character-Mode User Interface' Guide.

- DOCUMENTS, which structure the information contained in the Database in order to print documents whose readers are the applications' end-users. These documents contain information relevant for these end-users.

Refer to the 'Personalized Documentation Manager' Manual.

Chapter 4. Services

Search for Instances

Search by Keywords

Keywords are character strings associated with instances.

They enable you to find the instances which are 'marked' in this way in the Database. You can then find a piece of information or check that the piece of information you want to create does not already exist.

To know how to define keywords, associate them with instances and carry out a search by keywords, refer to the on-line help of the Administrator/Developer workbench, to the 'VA Pac WorkStation User Interface' Guide or to the 'Character-Mode User Interface' Guide.

KEYWORD TYPES

There are two types of keywords:

- Implicit keywords, which are automatically made up from the instances' names. If these names are clear enough, finding instances related to a topic is really easy.

Example: To find all the instances related to dates, you can carry out a search on the 'date' keyword. Then you will find the texts named 'Date Parameters', 'Date Input', the Data Element named 'Beginning date', the Program named 'Computation of payment date'...

- Explicit keywords, entered on the instances' Definitions, according to the company's needs, to indicate additional management criteria.

Example: a Data Element with the 'VAL' keywords is bound to be validated ; a Program with the 'PROD' keyword has been put into production, etc.

KEYWORDS THESAURUS

All keywords are automatically integrated into a thesaurus.

This thesaurus contains all the keywords and indicates the number of uses of each one.

This thesaurus can be complemented by the keywords' definitions or the assignment of synonyms.

Search by Lists

Search by Lists

The instances of each entity can be ordered in lists, which vary according to the sort criterion you specify.

You can choose to display the list by code (this criterion is always available, on any entity), name, type...

To make your search more efficient, you can also vary the scope of the target Libraries: you can extend the search to all the Libraries or narrow it to a specific Library...

Search by cross-references

Since an instance can be used as many times as necessary by other instances, it is useful to rapidly make the connection between these instances.

When you carry out a search by cross-references, you benefit from a global and coherent view of the Database. You can view all the cross-references between instances.

Cross-references represent a major asset in the evolving or corrective maintenance stage because they enable you to immediately assess the impacts of the update of an instance.

For example, if you modify a Data Element, the Segments, Reports and Programs in which it is used will be automatically modified accordingly and you may have to change the Reports' maps if the format of the Data Element has changed...

To know how to implement the search by cross-references, refer to the on-line help of the Administrator/Developer workbench, to the 'VA Pac WorkStation User Interface' Guide or to the 'Character-Mode User Interface' Guide.

Metamodel's On-Line Help

FOR THE ADMINISTRATOR/DEVELOPER WORKBENCH

You can access the on-line help from any location in the WorkStation via the general table of contents of the on-line help, or from the windows and input fields.

FOR THE VA Pac WORKSTATION OR THE CHARACTER-MODE INTERFACE

You can access the on-line help from any input window.

Two types of help are available for each window:

- window help, which describes the purpose of the window, its general characteristics and its uses.
- input field help, which describes the coding rules, possible values and error messages of each input field of the window.

To know how to access the on-line help, refer to the 'VA Pac WorkStation User Interface' Guide or to the 'Character-Mode User Interface' Guide.

Journalization of Updates

When the Administrator creates the Database, he/she can activate an option to store all the update transactions in a journal.

This journal, shared by all the Database users, contains one line for each update transaction. This line contains:

- The code of the user who carried out the update,
- The user's work context,
- The time of the update,

- The action code of the updated line,
- the element which was updated.

All users can view the journal.

Moreover, the user can select transactions by entering selection criteria (ex: view the transactions performed by one user, view the transactions carried out after a specific time...).

Some utilities use this journal.

Access Security

Library Level

Since all of the company's projects are stored in a sole Repository, some pieces of information require to be secured.

So the Database Administrator assigns user profiles, i.e. authorizations valid for each user profile (developer profile, project leader profile...). The Administrator can modify them for some users specifically.

Authorizations can be general on the Database or specific on each Library.

For detail, refer to the on-line help of the Administrator workbench.

Note: The Administrator can restrict the users' accesses by putting the VA Pac Database under DSMS monitoring. For explanations, refer to the on-line help of the Administrator workbench or the DSMS Reference Manual.

Instance Level

Since data can be handled by many users, a timestamp manages concurrent updates via an optimistic lock.

If a user has updated a data item and if he/she wants to store it in the Repository, two cases are possible:

- This data item has not been updated meanwhile by another user. The update is then stored immediately.
- The data item has been updated meanwhile by another user (very rare case). A message displays the name of the user who carried out the update. The user can then choose to override this update with his/her own update.

Moreover, beside this automatic timestamp control, a data item can be explicitly locked to prevent its use and update (ex: a Data Model under development, a Segment not validated yet...).

The other users will only be able to view the locked instances.