VisualAge Pacbase 2.5

PACBASE ACCESS FACILITY
REFERENCE MANUAL

DDPAF000251A

Note

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

According to your license 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.software.ibm.com/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.

First Edition (August 1998)

This edition applies to the following licensed program:

- VisualAge Pacbase Version 2.5

Comments on publications (including document reference number) should be sent electronically through the Support Center Web site at:
http://www.software.ibm.com/ad/vapacbase/support.htm
or to the following postal address:
IBM Paris Laboratory
VisualAge Pacbase Support
30, rue du Château des Rentiers
75640 PARIS Cedex 13
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.

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:

Intellectual Property and Licensing
International Business Machines Corporation
North Castle Drive, Armonk, New-York 10504-1785
USA

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 information which has been exchanged, should contact:

IBM Paris Laboratory
SMC Department
30, rue du Château des Rentiers
75640 PARIS Cedex 13
FRANCE

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.

TABLE OF CONTENTS

1. INTRODUCTION

1.1. INTRODUCTION TO THE PAF FUNCTION

INTRODUCTION TO THE PACBASE ACCESS FACILITY (PAF) FUNCTION

The PACBASE Access Facility (PAF) function allows the user to extract information from the VisualAge Pacbase Database via SQL statements. This extracted information can then be used to update the VisualAge Pacbase Database.

RELATIONAL VIEW OF THE DATABASE

The PAF Function provides a relational description of the standard VisualAge Pacbase metamodel and of the WorkStation metamodel. This type of description is a prerequisite for the formulation of SQL queries.

A relational model is a tabular structure of data organized in columns. Relations between objects (data bases, table spaces, tables, views, or indexes - but not the data itself) are correspondences between the table columns.

In the PAF tabular model:

. Each entity (standard entities and User Entities) is described by a set of tables.

. Information about an entity definition or description is described by specific columns.

. Cross-references are described in two ways:

- By direct relations between two columns of different Tables,

- By "virtual" tables, whose columns identify the two tables to be related and represent the cross-references.

DESCRIPTION OF TABLES

The tables and columns of the VA Pac-PACLAN-PACLAN/X metamodel are described in detail in the reference manual: PAF Tables/ Host (Ref: DD PAG).

If you ever need to write PAF programs concerning method entities managed by the WorkStation, the edition is performed from the GENERATION AND PRINTING COMMANDS screen (CH: GP), with PCM command in the COM field. The method code -one character only- will be input in the ENTITY field.

List of possible values for methods:

M stands for Merise
D stands for YSM
Y stands for Yourdon
A stands for SSADM
O stands for OMT
F stands for IFW

You will then have the description of the tables and columns for the required metamodel.

DATABASE UPDATING

Refer to subchpater "UPDP - BATCH mode" (chapter "PUF - PACBASE UPDATE FACILITY").

INTRODUCTION TO THE SQL LANGUAGE

The Structured Query Language (SQL) is the standard query language used with Relational Databases. Its syntax is similar to that of the English language.

This language is used to formulate queries, retrieve selected data, and update the database.

An SQL query is defined by a SELECT statement (See Subchapter "SYNTAX OF THE SQL-PAF LANGUAGE").

The tables in a Relational Database are composed of rows (equivalent to records of a file).

An SQL query defines a subset of the information contained in a database without specifying when or how this information was obtained and processed.

In order to integrate SQL with the procedural languages used in business-oriented computing, the concept of the cursor has been associated with an SQL query.

A cursor makes the extracted rows available to a PAF user program. The cursor is defined and declared with a DECLARE CURSOR statement, which includes a SELECT statement that identifies the extracted rows. When a cursor is used, the program can retrieve each extracted row sequentially: the cursor must be opened (with an OPEN statement) before any rows are retrieved. A FETCH statement is used to retrieve the cursor's current row, and can be executed repeatedly until all rows are retrieved. Then, the cursor must be closed with a CLOSE statement.

Once the cursor has been defined, the statements make the cursor similar to a file; the FETCH statement is the equivalent of a READ statement in COBOL.

STANDARD SQL QUERIES

With the PACBASE Access Facility (PAF), the Database can be accessed via the Structured Query Language (SQL), which is commonly used for Relational Databases.

Accesses are performed by the declaration and use of SQL cursors.

Each cursor is associated with a VA Pac context, i.e. Library and Database Session. Several cursors can be processed at one time in order to access information pertaining to several VisualAge Pacbase cross-references.

Queries can be written directly in COBOL, described in all batch or on-line user programs, or generated by VisualAge Pacbase.
In all cases, the query is processed by the PAF Translator program (before the COBOL compilation), which translates the SQL statements into CALL instructions used by the PAF Extraction Sub-Program.

THE PAF EXTRACTION SUB-PROGRAM

This Sub-Program accesses and extracts data from the Database. It retrieves the internal parameters built by the PAF Translator Program in order to perform the requested data extraction.

Extracted data is transmitted to the PAF user program in the Communication Area generated by the PAF Translator Program (COBOL Communication Area in the WORKING-STORAGE SECTION).

2. IMPLEMENTATION IN USER PROGRAMS

2.1. INTRODUCTION

INTRODUCTION

THE USER PROGRAM

PAF is implemented through a user on-line or batch program, either written directly in COBOL or generated by VA Pac. The PAF Extraction Sub-Program generates all the accesses to the Specifications Database.

In the user's program, the cursor(s) must first be declared via a DECLARE CURSOR statement in order to access the desired tables. For each declared cursor, the sequence of statements is as follows:

CONNECT Connect Cursor to a Visualge Pacbase context (User, Library, and Database Session),

OPEN Open Cursor, i.e. extraction from the Specifications Database,

FETCH Sequentially retrieve extracted rows,

CLOSE Close Cursor.

The SET statement allows for the dynamic modification of the PAF Translator Program's operating parameters.

The INIT and QUIT statements perform technical initialization and termination operations according to the extraction mode and the hardware in use (e.g. File OPEN/CLOSE in batch mode). They are not required in on-line mode.

THE PAF TRANSLATOR

SQL commands are inserted into a PAF user program and are translated into COBOL instructions before the COBOL compilation.

The PAF Translator transforms SQL statements into comment lines which precede the translated COBOL instructions.

The SQL-PAF DECLARE statement is translated into a declaration. Other SQL commands are translated into CALLs of the Extraction Sub-Program, except for the SET statement, which is not an SQL command and has a very special usage.

For more information, please refer to Paragraph "THE SET STATEMENT" in Subchapter "SYNTAX OF THE SQL-PAF LANGUAGE".

The PAF Translator is parameterized by a comment line inserted into the PAF user program following the IDENTIFICATION DIVISION line. This parameter line, the description of which appears on the next page, is automatically generated if the program is developed with PACBASE and if the 'EXP' operator is used on a Procedural Code (-P) line of the program.

For more details, please refer to Subchapter "PAF IMPLEMENTATION UNDER VISUALAGE PACBASE".

The PAF Translator parameter line is formatted as follows:

COLUMN	LENGTH	CONTENT
1	6	COBOL LINE NUMBER
7	1	* FOR COBOL COMMENT LINE
8	5	EXECUTION MODE: BATCH or ON-LINE
14	4	FIXED LABEL
19	3	LIBRARY CODE
23	5	SESSION NUMBER (& VERSION)
29	2	GENERATION VARIANT(S) (COBOL & MAP)
32	3	FIXED LABEL
36	1	DATABASE LANGUAGE CODE (A or F)
38	3	BATCH PROGRAM SKELETON ON-LINE PROGRAM SKELETON COBOL PROGRAM SKELETON
42	1	SKELETON LANGUAGE (A or F)
44	6	'SINGLE' OR 'DOUBLE' QUOTES DELIMITER

The Execution Mode is used to distinguish between batch and on-line. The Execution Mode that VA Pac takes into account depends on the generator implemented on-site. The Execution Mode allows the PAF Translator to declare, as appropriate, the work areas specific to on-line and to generate the calls to the Extraction Sub-Program.

The Generation Variant taken into account by VA Pac depends on the one specified on the Program Definition screen. It is used to adapt the generated syntax in function with the compiler.

The String Delimiter which VA Pac takes into account depends on what was specified on the Library Definiton screen. It allows the PAF Translator to recognize the string delimiter for both generation and source analysis.

The Library and Session parameters allow the PAF Translator to connect to the appropriate VA Pac Database when it is processing a Cursor dealing with a User Entity Occurrence: the columns associated with User Entity Occurrences depend on the description of the corresponding User Entity. Therefore, the PAF Translator has to read this User Entity in the Library and Session where it is described in order to validate the SQL query.

The PAF Translator is a bilingual program. The first specified Language Code applies to error messages generated by the Translator. The second refers to the language used in the mnemonic coding of the Tables and Columns which describe the VisualAge Pacbase Database.

This implementation of two language codes (English and French) allows a site to generate programs for another site using a different language code. The values taken by VA Pac for these two language codes are those of the AE and SG files (error messages and generation skeleton).

The line 2 generated by the VA Pac generator is read by the PAF generator, but is not reproduced in the translated source.

The SET statement can be used to easily modify these parameters anytime during the actual translation process. For example, it may be necessary to change the Library Code one or more times. However, in the case of an on-line program containing the INSERT and FETCHER statements (without DECLARE CURSOR), the SET statement must come before the previous statements for the initialization of the aerea (e.g., writing the statement in Working Storage Section).

THE PAF EXTRACTOR SUB-PROGRAM

The PAF Extractor Sub-Program manages accesses to the VisualAge Pacbase Database.

This sub-program retrieves the internal parameters built by the PAF Translator and performs the selected data extraction as follows:

* When a CONNECT statement is issued, the Extractor establishes the user's connection, for the specified Cursor, to the VA Pac Database (validates access authorization to Library and Database Session).

* When an OPEN statement is issued, the Extractor accesses the VA Pac Database, and stores the extracted rows in an temporary Work File. The number of extracted rows may be parameterized for each Cursor (SIZE parameter within the CONNECT Statement).

* When a FETCH statement is issued, the Extractor retrieves the extracted rows, one by one, from the Temporary Work File and transmits them to the user program Communication Area generated by the PAF Translator.

  Refer to Subchapter "DATABASE ACCESS OPTIMIZATION" which provides a detailed explanation of of the mechanism used by the Extractor Sub-Program to manage the OPEN and FETCH statements.

* When a CLOSE statement is issued, the specified Cursor is closed.

DAF extractor is also used in DUPD procedure which updates in batch mode DSMS database with a DAF Tables sequential file.
For more information, refer to operations manual

THE TEMPORARY WORKFILE

The purpose of the temporary Workfile is to store the rows extracted from the Specifications Database after an OPEN statement is issued (or a FETCH statement once all the extracted rows have been restituted to the user program).

Extracted rows are retrieved one by one from the temporary Workfile for each FETCH statement. The maximum number of extracted rows may be parameterized for each Cursor (via the SIZE parameter in the CONNECT statement).

For more details, refer to Subchapters "SYNTAX OF THE SQL-PAF LANGUAGE" and "DATABASE ACCESS OPTIMIZATION".

The temporary Workfile also contains technical parameters used by the PAF Extractor Sub-Program for Cursor management (See topic "SPECIFICATIONS FIELD" in Subchapter "THE TRANSLATED USER PROGRAM").

PHYSICAL DESCRIPTION

The temporary Workfile is an indexed sequential file with a variable format.

When this file is created for a batch job, its access key is composed of:

. A Cursor code,
. A Structure code,
. A Record number.

When this file is created for on-line use, it is used by all PAF applications and users, and its access key is composed of:

. A Conversation identifier,
. A Cursor code,
. A Structure code,
. A Record number.

CICS: In a CICS environment, there can be several PACBASE databases; one temporary Workfile is created for each database.

2.2. SYNTAX OF THE SQL-PAF LANGUAGE

SYNTAX OF THE SQL-PAF LANGUAGE

Due to the nature of the first release of PAF, i.e., as an extraction tool, the SQL statements to be used with PAF are limited to database queries only. Creation, Modification, and Deletion statements are not recognized.

GENERAL INFORMATION

To access the Specifications Database through the PAF function, the user has to declare, and then use, SQL Cursors.

For each table to be accessed, the user must declare a Cursor in the WORKING-STORAGE SECTION (DECLARE CURSOR statement). In the PROCEDURE DIVISION, the sequence of statements associated with a given Cursor is as follows:

CONNECT, OPEN, FETCH, and CLOSE.

OPEN, FETCH and CLOSE are standard SQL statements, while the CONNECT statement is specific to the PAF function. All four of these statements are designated as Cursor operation statements in SQL-PAF syntax.

A PAF user program can use up to 100 Cursors.

The INIT (initialization) and QUIT (termination) statements, which are independent of Cursors, must be issued, respectively, before and after any Cursor operation statements (compulsory in Batch mode).

NOTE: All SQL-PAF sentences must be coded starting in COBOL column 12, must begin with EXEC PAF, and must end with END-EXEC.

CURSOR DECLARATION

An SQL Cursor is declared in the WORKING-STORAGE SECTION of the PAF user program by means of a DECLARE CURSOR statement.

In the DECLARE statement, the following keywords should be noted:

SELECT, FROM, WHERE, AND, and OR.

The syntax of the DECLARE CURSOR statement is as follows (values between parentheses are optional):

```
EXEC PAF DECLARE <cursor-code> CURSOR FOR
SELECT * FROM <table-code> (WHERE <condition(s)>)
END-EXEC
```

where:

- <cursor-code> is the four-character cursor identifier,

- SELECT here applies to the whole table, and is used to retrieve all table columns; in other words, columns cannot be selected individually. Thus, the syntax is always SELECT *.

- FROM cannot be used with a JOIN clause, and is therefore followed by just one table code.

- <table-code> identifies the PAF Table. The tables and columns of the VisualAge Pacbase metamodel are described in detail in the PAF Tables/Host (Ref: DD PAG) Reference Manual. To edit tables and columns descriptions for method entities, refer to sub-chapter 'Introduction to PAF function', section 'Description of Tables'.

- WHERE does not allow SUBSELECTs.

- <condition(s)>: Each condition applies to a table column and is indicated in parentheses.

  Several conditions may be linked using the logical 'AND' and 'OR' operators. The total number of elementary conditions is limited to 50.

A condition is formatted as follows:

COLUMN OPERATOR OPERAND

where:

. COLUMN = column code

. OPERATOR may have the following values:

```
=   : equals
>   : is greater than
>=  : is greater than or equal to
<   : is less than
<=  : is less than or equal to
<>  : is different from
```

. OPERAND is either:

```
. Another column of the table,
. A COBOL constant,
. A PAF user program COBOL variable.
```

NOTE: Alphanumeric constants cannot exceed 60 characters. If this length is insufficient, an initialized COBOL variable can be used instead.

Numeric constants can only be unsigned integer constants and cannot exceed 18 digits. The PAF Translator does not validate the declaration of COBOL variables used as operands. Subscripted COBOL variables cannot be created.

Limitations related to the use of the SELECT clause are not restrictive, since the ability to manage several cursors at the same time makes up for the inconvenience of mono-table accesses. Furthermore, coding a very complicated SQL query is often a tricky matter. The purpose of the PAF Function is not to provide a comprehensive SQL interface, but to allow its users to access any data contained in a PACBASE Specifications Database. Data is accessed at the PAF user program level.

For more information please refer to Subchapter "EMBEDDED PAF CURSORS" in this Chapter.

CURSOR MANAGEMENT

Cursor operation statements are written in the PROCEDURE DIVISION of a PAF user program.

CONNECT: This is the first statement to be issued. It performs the connection to a VisualAge Pacbase context. This context can be modified as many times as needed.

The syntax of the CONNECT statement is as follows:

```
EXEC PAF CONNECT <cursor-code> TO
    USER = <user-code>
    PASS = <password>
    LIB = <library-code>
    SESSION = <session-number-and-version>
    NET = <sub-network-option>
    SIZE = <maximum-number-of-rows>
    IDENT = <transaction-code>
    BASE = <PACBASE-Database-code>
END-EXEC
```

The parameters to the right of the equal sign ('='), which are described below, must be either literals or COBOL variables.

USER VisualAge Pacbase user code

PASS VisualAge Pacbase user password

LIB VisualAge Pacbase library code

SESSION VisualAge Pacbase session number and version

NET VisualAge Pacbase Database sub-network option:

I, C, A, U, <, >, and Z

SIZE Maximum number of rows stored in the temporary Work File

IDENT Conversation transaction code

BASE 4-character Database code

All of the parameters in the CONNECT statement are required for the first cursor connection. The CONNECT statement is used to assign values to the parameters associated with the cursor during extraction. However, the user may modify certain parameters during a subsequent connection (for example, to use the same request in a different VA Pac context).

It is always possible to reconnect a cursor (for example to explore another sub-network with the same query). In order to request another VA Pac connection, the user only has to enter the parameters for the new connection and does not have to enter them in the order specified above.

EXAMPLE:

```
EXEC PAF CONNECT <cursor-code> TO
NET = <sub-network-option>
LIB = <library-code>
END-EXEC
```

NOTES ON THE 'IDENT' AND 'BASE' PARAMETERS

These two parameters are only used in on-line PAF user programs.

The IDENT parameter is a 25-character field which identifies the on-line conversation using the PAF function. It is also part of the access key in the temporary Work File. Its value depends on the monitor in use. For example, on CICS, it is advised to set this parameter with the DFHTERMID variable supplied by CICS. For complete information on the IDENT values, refer to "THE 'IDENT' PARAMETER" subchapter.

The BASE parameter contains the four-character database code for access to the VA Pac Database and the PAF Temporary Work File. (This parameter is used only with IBM hardware running on CICS).

NOTES ON THE 'SIZE' PARAMETER

The maximum number of extracted rows in the temporary Workfile may be exceeded in the case of the following tables:

- Segment Descriptions (SEGDEL),
- General Documentation of Entities (entDOC) or Entity Descriptions (entdscDOC) in which Parameterized Input Aids are called.

For these three types of tables, the extraction proceeds until the end of the Segment or PIA description. It stops when the SIZE limit (max. no. of rows) is reached or exceeded.

The SIZE parameter value must be greater than zero. Otherwise, the PAF Extractor Sub-Program automatically sets it to "1".

OPEN: When this statement is issued, the Extractor Sub-Program accesses the VA Pac Database and writes, in the temporary Workfile, the rows selected (and extracted) according to the cursor declaration.

A cursor can be opened if it is already connected and if it has not yet been opened.

The syntax of the OPEN statement is:

```
EXEC PAF OPEN <cursor-code> END-EXEC
```

FETCH: When this statement is issued, the Extractor Sub-Program sends the current retrieved row from the Intermediary Work File to the PAF user program.

The cursor must be opened and there may be as many FETCH statements as necessary.

The syntax of the FETCH statement is:

```
EXEC PAF FETCH <cursor-code> END-EXEC
```

NOTE: The standard syntax of the FETCH statement includes the keyword 'INTO', which allows each selected column to be associated with a COBOL field.

The PAF-SQL extractor extracts ALL columns into a Communication Area, generated in the WORKING-STORAGE SECTION when a DECLARE statement is issued.

Thus, the keyword INTO (followed by the list of target COBOL fields) is not used in the SQL-PAF FETCH statement.

CLOSE: Only an opened cursor can be closed: a CLOSE statement can be issued only after an OPEN or a FETCH statement.

The syntax of the CLOSE statement is:

```
EXEC PAF CLOSE <cursor-code> END-EXEC
```

INSERT: INSERT inserts data in the intermediary workfile (for example, a PAF extraction). These data must be stored initially in the pref-PUFENR field of pref-PAFCOM - "pref" being a prefix of 1 to 4 characters long.

Note:
At the time of an INSERT the workfile may already contain "70"-type recordings with the same identifier provided by previous transactions. Consequently, before inserting new data in the workfile, these recordings are re-read in order to determine the number of the statement which will be affected when the next recording is inserted (pref-NUMENR field of pref-PAFCOM).

CALPUF: This statement changes itself to call to the PUF subprogram.

FETCHER: It returns the errors one by one which could have been stored in the intermediary file by the PUF subprogram in the pref-PUFERR field of pref-PAFCOM. The user program is the one who foresees the restoration loop.

INSERT, CALPUF et FETCHER are independents from the cursors and exist only in on-line mode.

THE SET STATEMENT

The SET statement is used to modify a number of parameters used by the PAF Translator.

These parameters are initialized by the second line (a COMMENT line following the IDENTIFICATION DIVISION) in a PAF user program. This line is described in this chapter, in the "INTRODUCTION" Subchapter, Paragraph "THE PAF TRANSLATOR".

After the specified parameters are actually updated, the PAF Translator sends the SET statement as comments to the translated PAF user program. The SET statement is the only SQL-PAF statement which is not translated into COBOL.

The statement may be written in WSS or Procedure Division. However, if these parameters are not to be modified during the program, it is recommended to write the SET statement at the beginning of the Working Storage Section.

The syntax of the SET statement is as follows:

```
EXEC PAF SET
    LIB = <library code>
    DELIM = <delimiter>
    TYPE = <generation variant(s)>
    MODE = <execution mode>
    LINK = <calling mode>
    ROOT = <root>
    SESSION = <session-number-and-version>
    USRENT = <UEO parm>
    CTXCOL = <CTX parm (context)>
    BASPUF = <base-PUF>
    IDPUFL = <ident-PUF>
    UPDATE = <pref>
END-EXEC
```

The syntax of the SET statement is similar to that of the CONNECT statement. In particular, it is not necessary to give new values to the LIB, DELIM, TYPE, MODE and ROOT parameters and to enter them in the order given above.

THE PAF TRANSLATOR PARAMETERS

The values of these parameters are purely alphanumeric and do not require a string delimiter.

EXAMPLE: In order to modify the LIB parameter, using the library code "PFA", the syntax would be as follows:

EXEC PAF SET LIB = PFA END-EXEC

DESCRIPTION OF PARAMETERS:

LIB is a 3-character parameter. Its default value is the Library from which the PAF user program is generated.

It is used to declare a cursor for a table of User Entity Occurrences (UEO's) described by User Entities not found in the Library from which the PAF user program is generated.

DELIM is a six-character parameter. It is used to modify the delimiter using the following values:

SINGLE: for the sinqle quote ('),
DOUBLE: for the double quote (").

TYPE is a two-character parameter. It is used to modify the variants VA PAC TYPE OF TP MONITOR and MAP TO GENERATE/TYPE OF COBOL TO GENERATE.

EXAMPLE: When a PAF user program is written and generated with the PACBASE Batch Systems Development component in an IBM CICS/MVS environment (Variants '00' or 'X0'), in order to obtain an On-Line application to be executed in an IBM IMS/VS environment, the TYPE parameter value needs to be changed to '01' or 'X1' ('00' is automatically generated by the PAF Translator).

Refer to the ON-LINE SYSTEMS DEVELOPMENT and BATCH SYSTEMS DEVELOPMENT Reference Manuals for a list of the values of these variants.

MODE is a five-character parameter. It is used to modify the execution mode for PACBASE users who are using the Batch generator to generate on-line programs (value of MODE = 'TP').

LINK is a six-character parameter. It is used to specify the calling mode of the extractor (STATIC or DYNAM) for static or dynamic calls, respectively. For CICS, this parameter is ignored and the translator generates a LINK statement for the call to the extractor. For batch DOS, the call is always dynamic and is executed using module PACDYNAM. For BULL DPS7 and DPS8 hardware, the statement is ignored for on-line processing, in which case the call is always static.

ROOT is a two-character parameter. It is used to modify the first two characters of the external names of the extraction sub-programs. This parameter is called 'ROOT', because the standard way to parameterize these sub-programs is to assign the first two characters of the root of the VisualAge Pacbase database to the external names of these sub-programs.

SESSION is a five-character parameter. It is used to modify the session. Its default value is the session in which the PAF user program is generated.

USRENT (1 to 5 characters). This parameter is used to modify the codes of the standard columns in the UEO definition and description tables, in order to avoid the problem of codes conflicting with the user data elements called in the description of the associated user entity. The default codes of these columns are CUEO, LUEO, NUEO and WUEO for the UEO code, name, description line number and explicit keywords, respectively (see the "PAF TABLES/HOST" manual, Chapter "DESCRIPION OF PAF TABLES", Subchapter "USER ENTITY OCCURRENCES"). The USRENT parameter value replaces the 'UEO' string in each of these column codes:

USRENT = XXX renames these columns as CXXX, LXXX, NXXX and WXXX, respectively.

CTXCOL (1 to 5 characters). This parameter is used to modify the codes of the context columns of every PAF table, in order to avoid the problem of conflicting codes in the definition or description tables with user data elements called in the description of the associated user entity. The default codes of these columns are LCTX, SCTX, TCTX and for the library code, session number and hierarchical indicator, respectively (see chapter "IMPLEMENTATION IN USER PROGRAMS", subchapter "THE TRANSLATED USER PROGRAM").

The CTXCOL parameter value replaces the 'CTX' string in each of these column codes:
CTXCOL = YYY renames these columns as LYYY, SYYY, TYYY and NYYY, respectively.

It is important that the USRENT and CTXCOL parameters are NOT assigned the same values, in order to avoid additional conflicts.

The "UPDATE = pref" clause

Generates the description of the two workings which will be used at the time of the calls to the PAF extractor (pref-PAFCOM) or to the PUF program (pref-PUFCOM) with the help of the INSERT, CALPUF and FETCHER statements - "pref" being a prefix of 1 to 4 characters.

The "BASPUF = base-PFU" and "IDPUF = ident-PUF" statements

The PUF statements are independent of PAF cursurs. The PUF- specific identifiers must be declared in order to give access to the workfile.
- "base-PUF": VA Pac transaction code (4 chars.)
- "ident-PUF": identifier (1-25 chars.)

NOTE:
For the last two clauses, parameters on the right side of '=' may be constants (to be documented in ") or COBOL variables (for CONNECT).

TECHNICAL INITIALIZATION AND TERMINATION

The INIT and QUIT statements allow the Extractor to perform technical initialization and termination operations which depend on the Operating System in use. As a result, both statements must be issued, respectively, before and after other SQL-PAF statements.

The syntax of the INIT statement is:

```
EXEC PAF INIT END-EXEC
```

The syntax of the QUIT statement is:

```
EXEC PAF QUIT END-EXEC
```

2.3. DATABASE ACCESS OPTIMIZATION

THE WHERE CLAUSE

Using the WHERE clause improves overall performance by optimizing Database accesses and reducing the volume of extracted data stored in the SYSPAF intermediary file (only relevant data need be extracted).

1. Optimizing Database Accesses:

A Table access will prove more or less efficient according to the WHERE clause utilization.

EXAMPLE: Access to the SEGDEL Table

. SELECT * FROM SEGDEL WHERE CDEL = 'cccccc'

Data Element's cross-references to Segments
Equivalent on-line choice: EccccccXS

. SELECT * FROM SEGDEL WHERE CSEG = 'ssss'

Direct access to the Segment's Description
Equivalent on-line choice: SssssCE

. SELECT * FROM SEGDEL WHERE CSEG > 'ssss'

```
Prior access to the List of Segments, then -- for each
Segment -- an access to its Description is
performed.Equivalent on-line choice: LCSssss, S____CE
```

2. Decrease in Volume of Extracted Data (stored in the SYSPAF Temporary File):

EXAMPLE: Extraction of Programs which are macro-commands.

. Either select all the programs and then keep only those whose Column code TPGMNA = M,
. Or select in a WHERE clause only the programs whose Column code TPGMNA = M.

Extraction will obviously be quicker with the second solution.

CONCLUSION: It is highly recommended to use the WHERE clause and to use it with a maximum of restrictive paramaters.

THE SIZE PARAMETER

In order to optimize accesses to the VA Pac database, the PAF Extractor reads several rows in advance. Advance reading avoids a systematic Read each time a row is fetched; systematic Reads cause systematic resets (START) in the PACBASE Index (AN) file.

Advance reading may be seen as the logical equivalent of Input/Output BUFFERs used by File Access Methods implemented with any Operating System.

For each cursor, the user can parameterize the number of rows read in advance by using the SIZE parameter in the PAF function's CONNECT statement.

MANAGEMENT OF ROWS READ BY THE PAF EXTRACTOR

The PAF Extractor reads the Database and validates the row that has been read. If the result is valid, the row is stored in the temporary Workfile.

This process is repeated as long as the number of stored rows is less than the value of the SIZE parameter.

Processing ends when the Extractor READ function detects the last row for the declared cursor.

The Extractor READ function returns the number of rows actually read and an End-of-Cursor Indicator.

This Read function is systematically executed when a cursor OPEN statement is issued.

It may also be executed when a FETCH statement is issued. The purpose of a FETCH is mainly to read the temporary Workfile in order to retrieve a new row.

When the retrieved row is the last one to be read by the Extractor READ function, the function is executed once again if end-of-cursor has not been detected.

ACCESS OPTIMIZATION VIA THE SIZE PARAMETER

The SIZE parameter is a critical factor in the PAF Extractor's efficiency in terms of response time.

SETTING THE SIZE PARAMETER FOR AN ON-LINE PAF USER PROGRAM:

The value of the SIZE parameter should not be less than the number of rows fetched per screen (number of records displayed on the screen, for example). Based on a simple hypothesis that all rows which are read are also valid, the optimal value of the SIZE parameter is a multiple of the number of rows fetched per screen.

NOTE: If the on-line PAF user program includes screen branching operations, the optimal value of the SIZE parameter is equal to the number of rows fetched per screen.

SETTING THE SIZE PARAMETER FOR A BATCH PAF USER PROGRAM:

At first glance, it may seem appropriate to set the SIZE parameter to a large value in order to minimize the number of READs. However, a value which is too large would result in the dynamic creation of too many records for most indexed sequential file access methods.

The ideal in Batch is to sufficiently define the size of the Input/Output BUFFER for the temporary Work File so that the READ function causes only logical input-outputs.

2.4. THE 'IDENT' PARAMETER

THE 'IDENT' PARAMETER

The purpose of the IDENT parameter is to uniquely identify a conversation in a multi-user environment.

This identification is therefore closely linked to the TP Monitor under which the translated PAF user application will be executed.

The recommendations below take into account the standard variables supplied with the TP Monitor (terminal identifier, etc.).

CICS/MVS or DOS:

The EIBTRMID CICS variable identifies each terminal.

IMS/VS:

The name of the logical terminal is found in the IO/PCB field (S-IPCB-XNMTE variable in programs generated by the PACBASE OLSD function).

DPS7 and DPS8:

The variable associated with the SYMBOLIC SOURCE is found in the COMMUNICATION SECTION (7-CD01-XTERM in programs generated by the PACBASE OLSD function).

ICL: The system variable has no standard name. In programs generated by the VA PAC OLSD function, it is the TERMINAL-NAME field in the INPUT-MESSAGE.

UNISYS A SERIES:

The system variable is called COMS-IN-STATION.

MS-DOS/OS2 Environment (VA PAC OS2):

The temporary Workfile is managed locally, and on-line mode is identical to batch mode. Therefore, there is no IDENT parameter.

2.5. PAF IMPLEMENTATION UNDER VISUALAGE PACBASE

PAF IMPLEMENTATION UNDER VISUALAGE PACBASE

The DECLARE CURSOR clause must be entered in the WORKING-STORAGE SECTION.

It must therefore be inserted on Work Area (-W) lines in the Program or On-Line Screen.

"EXEC PAF" must start in the 5th position of the LEVEL OR SECTION field, and "END-EXEC" must be entered after the cursor declaration.

EXAMPLE:

```
   A LIN T LEVEL OR SECTION WORK AREA DESCRIPTION

            100       EXEC PAF DECLARE CU01 CURSOR FOR
            120       SELECT * FROM DELDEF
            140       WHERE FDELIL = 5 END-EXEC
```

Except for the DECLARE CURSOR clause, all SQL-PAF statements are written on the Procedural Code (-P) lines of the PAF user Program or On-Line Screen. An EXP PAF Operator generates EXEC PAF and END-EXEC calls, which are found before and after all SQL-PAF statements, respectively.

EXAMPLE:

```
   A SF LIN OPE OPERANDS

                  EXP OPEN CU01
```

will generate:

```
         EXEC PAF
         OPEN CU01
         END-EXEC
```

2.6. THE TRANSLATED USER PROGRAM

THE TRANSLATED USER PROGRAM

Before COBOL compilation, the PAF Translator transforms SQL-PAF queries into COBOL declarations and instructions. (Warning: PAF translator is incompatible with a COBOL formatting request).

The EXEC PAF .... END-EXEC sequences are commented out in the COBOL program.

WORKING-STORAGE SECTION

The following phrase:

```
DECLARE <cursor-code> CURSOR FOR SELECT * FROM <table-code>
```

generates the following data declarations in the WORKING- STORAGE SECTION, under Level 01 <cursor-code>-CURSOR, of the PAF user program:

- The Cursor Management field,
- The Specifications field, where the query is translated,
- The Communication Area, i.e., the selected PAF Table.

The two fields which are accessible in the program (Cursor Management Field and Communication Area) are prefixed by the cursor-code. The Specifications field is generated as a FILLER.

EXAMPLE:

Extraction of Text Descriptions (TXTDSC Table) for Text Entity 'TEXT01', including Text Paragraphs greater than 'EE'. The cursor-code is TX04.

```
EXEC PAF  DECLARE TX04 CURSOR FOR SELECT * FROM
          TXTDSC WHERE CTXT = 'TEXT01' AND CPAR > 'EE'
END-EXEC
```

The following fields are grouped under the level:

```
01    TX04-CURSOR
```

CURSOR MANAGEMENT FIELD

```
05      TX04-SAVE.
  10    FILLER        PIC X(06) VALUE 'TX0401'.
  10    TX04-TABCOD   PIC X(10) VALUE 'TXTDSC    '.
  10    TX04-RETCOD   PIC 9(00002).
  10    TX04-ORDER    PIC X(00001).
  10    TX04-FI       PIC X(00001).
  10    TX04-FT       PIC X(00001).
  10    TX04-CUSRCU   PIC X(00008).
  10    TX04-CPSWCU   PIC X(00008).
  10    TX04-CLIBCU   PIC X(00003).
  10    TX04-CSESCU   PIC X(00005).
  10    TX04-CNETCU   PIC X(00001).
  10    TX04-NRECCU   PIC 9(00006).
  10    TX04-INTERN   PIC X(00034).
```

TABCOD Table code.

RETCOD Extractor Return Code.

"0": No error detected. Other values are presented in Chapter "ERROR MESSAGES", Subchapter "THE EXTRACTOR".

ORDER Each PAF statement is identified by a number:

1 INIT
2 CONNECT
4 OPEN
6 FETCH
8 CLOSE
9 QUIT

FI End of cursor READ:

1 Read of cursor's last row,
0 Otherwise.

FT End of cursor processing:

1 Cursor FETCH beyond its last row,
0 Otherwise.

NOTE: FI is set to '1' in two cases:
. The FETCH command returns the last corresponding record to the cursor selection.
. No data is selected by the OPEN command.

If a FETCH command is executed when FI = '1', the FT indicator is returned to '1' without the new record being used.
The OPEN command can also change the FI indicator to avoid a useless FETCH.

The other fields contain user parameters related to the CONNECT statement. The values of these parameters are automatically generated in the PROCEDURE DIVISION by the PAF Translator when the CONNECT statement is encountered.

CUSRCU VisualAge Pacbase user code

CPSWCU VisualAge Pacbase user password

CLIBCU VisualAge Pacbase library code

CSESCU VA PAC database session number (and version)

CNETCU VisualAge Pacbase database sub-network option

NRECCU Max. number of records in the Intermed. Work File

INTERN Internal usage field

NOTE: The Cursor Management field is called <cursor-code>-SAVE since, for an on-line PAF user program, this field has to be saved when the calling program returns control to the monitor. The On-Line PAF Extractor actually backs up a representation of the query in the Specifications field <cursor-code>-TECH (see next paragraph) in the Temporary Work File.

For an on-line PAF user program, the management field is the same as in batch but with two preceding fields, which are used to identify the database and the terminal (see description of the CONNECT statement):

```
10    TX04-IDENT      PIC X(25).
10    TX04-BASE       PIC X(4).
```

The contents of both fields, <cursor-code>-SAVE and <cursor-code>-TECH, must not be modified.

SPECIFICATIONS FIELD

This field is generated at the following level:

```
05      TX04-TECH.
```

This field is a variable length FILLER, where the query is translated for the PAF Extractor. The PAF user cannot access this field.

COMMUNICATION AREA

```
05      TX04.
  10    TX04-LCTX     PIC X(00003).
  10    TX04-SCTX     PIC 9(00004).
  10    TX04-TCTX     PIC X(00001).
  10    TX04-NCTX     PIC X(00001).
  10    TX04-CTXT     PIC X(00006).
  10    TX04-CPAR     PIC X(00002).
  10    TX04-CLIN     PIC 9(00003).
  10    TX04-TLIN     PIC X(00001).
  10    TX04-DLINTX   PIC X(00060).
  10    TX04-CDEL     PIC X(00006).
```

Whatever Table is selected, the first four columns are always generated. They specifiy the origin of the extracted line, i.e.:

LCTX Library code where the extracted line is defined.

SCTX Session Number when the line was last modified.

TCTX Version of session when the line was last modified.

NCTX Line Source Indicator:

= The line is extracted from the library to which the cursor is connected.

> The line is extracted from a higher-level library than the cursor connection library.

< The line is extracted from a lower-level library than the cursor connection library.

The codes of these four columns may be changed by using the SET statement with the 'CTXCOL' parameter.

The other columns are specific to the selected table.

PROCEDURE DIVISION

For each SQL-PAF statement in the PROCEDURE DIVISION, the following operations occur:

- the TX04-ORDER field is filled in,
- the Extractor Sub-Program is called and the entire TX04-CURSOR field is passed to it.

When a CONNECT statement is encountered, the Translator sends the user's parameters to the Cursor Management field.

When an OPEN statement is encountered, the Translator fills in (in some cases under specific conditions) the Specifications field (-TECH) when the cursor depends on one or more COBOL fields.

2.7. EMBEDDED PAF CURSORS

EMBEDDED PAF CURSORS

LIMITATIONS OF THE SQL-PAF SYNTAX

The SQL-PAF syntax uses a sub-set of the SQL language. In particular, cursors cannot be defined with embedded SELECT clauses. Embedded SELECT clauses are useful in obtaining information conditioned by a sequence of cross-references such as, a List of Data Elements called in Segments used in Programs.

However, this limitation is not too restrictive since you can declare several cursors (maximum number = 100). When a cursor depends on a COBOL field which belongs to the Communication Area of another cursor, both cursors act as one cursor using embedded SELECT clauses.

EMBEDDED CURSORS: EXAMPLE

Suppose a user wants to obtain, for each Data Structure in the VA Pac Database, the list of Programs that use it.

This query involves the following Tables:

DSTDEF Data Structure Definition,
PGMDST Program Call of Data Structures,

and the following Columns:

```
CDST Data Structure code,
CPGM Program code.
```

The SELECT clause of a standard SQL query is written as follows:

SELECT * FROM PGMDST WHERE
CDST = (SELECT CDST FROM DSTDEF)

With the SQL-PAF syntax, the same query uses two embedded cursors:

DECLARE LCD CURSOR FOR SELECT * FROM DSTDEF
DECLARE PGCD CURSOR FOR SELECT * FROM PGMDST
WHERE CDST = LCD-CDST

The first cursor, coded LCD, provides the list of all Data Structures. The second cursor, coded PGCD, provides the list of Programs using the Data Structure coded LCD-CDST, i.e., the code of the Data Structure currently fetched in the first cursor.

In the PROCEDURE DIVISION, after both cursors are connected, the LCD cursor is opened. As long as LCD-FT is not equal to one (i.e., the LCD cursor is still open), each time a FETCH statement is issued on the LCD cursor, an OPEN statement will be issued on the PGCD cursor. The PGCD cursor is issued. This OPEN allows the code of the next Data Structure to be moved to the LCD-CDST field.

After all FETCH statements are issued for the PGCD cursor, the cursor is closed, and another FETCH statement is issued for the LCD cursor.

In this way, you can simulate, through embedded cursor processing, a single cursor defined by embedded SELECT clauses.

REMINDER: Up to 100 cursors may be used by a PAF user program.

2.8. EXECUTION OF PAF USER PROGRAMS

EXECUTION OF PAF USER PROGRAMS

BATCH

Each time a Batch PAF user program is executed, you have to create and declare the temporary Workfile at the beginning of the job stream (and possibly delete the same file if it was previously created).

This file has the following characteristics:

. Indexed sequential access.

. Key length = 12.

. Maximum record size = 468.

. Average record size = 170.

Then, in order to execute the batch PAF user program, the VisualAge Pacbase Database files and the temporary Workfile have to be declared in the JCL. For more details, refer to the JCL examples in the operations manual.

ON-LINE

The temporary Workfile has the following characteristics when used with an on-line PAF user program:

. Indexed sequential access.

. Key length = 37, starting in position 2.

. Maximum record size = 539.

. Average record size = 200.

3. EXAMPLES OF PROGRAMS USING PAF

3.1. INTRODUCTION

INTRODUCTION

The purpose of this chapter is to present two examples of programs (batch and on-line) using PAF. Additionally, it suggests ways to use PAF programs (standard quality control, PACBASE database administration, etc.).

The first program example, 'PAFEX1', is a batch program. It builds a list of all the Properties (Data Elements whose Type = 'P') without relational names. Two cursors must be declared: one for the definition of Data Elements, and one for their descriptions. This batch program is a good example of how to use embedded cursors.

The second program example, 'PAFEX2', is an on-line program. It builds a list of screens that do not conform to a specific local standard (Data Element presentation, initialization character, screen or element help character). This program uses only one cursor (for the screen definition), and manages screen scrolling. This on-line program provides an example of how to transmit the PAF context between each iteration of a dialogue.

3.2. BATCH EXAMPLE

BATCH EXAMPLE

OBJECTIVE:

- To list the Properities without relational names.

PAF Cursor Declarations:

- CU01 selects 'P'-type Data Elements (Properties).

- CU02 (opened for each Element found by CU01) selects the description lines which contain a relational name.

Procedural logic:

- F02BA: PAF initialization.

- F02CA: CU01 Cursor Connection. The user code and password are hard-coded but could be obtained through a Read of an input file. The sub-network option is set to 'U', which indicates that only Properties in Library 'CIV' are taken into account. The CONNECT statement establishes the context for the PAF Read.

- FO2DA: CU02 Cursor Connection.

- F21BA: Opening the CU01 Cursor. This statement involves reading the 'P'-type Data Element Definition screens in Library 'CIV', and writing them in the PAF work file.

- F21CA: Fetching the definition screens, i.e., the screens are read one-by-one from the PAF work file.

- F21DA: As long as the end-of-cursor is not reached (CU01-FI = '0'), the CU02 Cursor is opened for each fetched element. This Cursor selects only the 'R'-type description lines of the fetched elements. Therefore, an immediate end-of-cursor (CU02-FI = '1') after the first FETCH means that this property does not have a relational name. In this case, a line is formatted and printed on a Report. The CU02 Cursor is closed so that it can be reopened for the next element of the CU01 Cursor.

- F79 : When all of the Properties have been FETCHED, the CU01 Cursor is closed and a QUIT statement is issued in order to close the database files and the PAF work file.

```
--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROGRAM DEFINITION....... PAFEX1                                            !
!                                                                             !
! PROGRAM NAME.......................: LIST PROPERTIES W/O SQL NAME           !
!                                                                             !
! CODE FOR SEQUENCE OF GENERATION....: PAFEX1                                 !
!                                                                             !
! TYPE OF CODE TO GENERATE...........: 0                                      !
! COBOL NUMBERING AND ALIGNMENT OPT..:                                        !
! CONTROL CARDS IN FRONT OF PROGRAM..:                                        !
! CONTROL CARDS IN BACK OF PROGRAM...:                                        !
! COBOL PROGRAM-ID...................: PAFEX1                                 !
! MODE OF PROGRAMMING................: P                                      !
! TYPE AND STRUCTURE OF PROGRAM......: B                                      !
! PROGRAM CLASSIFICATION CODE........: P   PROGRAM                            !
! TYPE OF PRESENCE VALIDATION........:                                        !
! SQL INDICATORS GENERATION WITH '-'.:                                        !
!                                                                             !
!                                                                             !
! EXPLICIT KEYWORDS..:                                                        !
!                                                                             !
! SESSION NUMBER.....: 2013        LIBRARY......: CIV    LOCK....:            !
!                                                                             !
! O: C1 CH: Ppafex1                    ACTION:                                !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! DATA STRUCTURES USED IN PROGRAM :    PAFEX1 LIST PROPERTIES W/O SQL NAME    !
!                                                                             !
! A DP CO : DL EXTERN OARFU BLOCK T  B M U RE SE L UNIT C  SELECTION F E R L PL!
!   PR    : PR PRLIST SSFOU     0 R      I                 A         I     1  !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!         :                                                                   !
!         :    STAT.FLD:             ACC. KEY:             RECTYPEL           !
!                                                                             !
! O: C1 CH: -CD                                                               !
--------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! WORK AREAS.........ENTITY TYPE P PAFEX1 LIST PROPERTIES W/O SQL NAME         !
!                                                                              !
! CODE FOR PLACEMENT..:      BA                                                !
! A LIN T LEVEL OR SECTION WORK AREA DESCRIPTION                        OCCURS!
!   100 * PROPERTIES LIST                                                      !
!   110       EXEC PAF DECLARE CU01 CURSOR FOR                                 !
!   120       SELECT * FROM DELDEF WHERE                                       !
!   130       TDEL = 'P'                                                       !
!   140       END-EXEC                                                         !
!   200 * LIST OF RELATIONAL NAMES OF A PROPERTY                               !
!   210       EXEC PAF DECLARE CU02 CURSOR FOR                                 !
!   220       SELECT * FROM DELDSC WHERE                                       !
!   230       CDEL = CU01-CDEL AND                                             !
!   240       TLIN = 'R'                                                       !
!   250       END-EXEC                                                         !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -W                                                                 !
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! PROCEDURAL CODE  P PAFEX1 LIST PROPERTIES W/O SQL NAME        FUNCTION: 02   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!          N   INITIALIZATION AND CONNECTIONS    05BL                          !
! - -- --- --- ------------------------------- ---- --------------------------!
!   BA     N   INITIALIZATION                    10BL                          !
!   BA 100 EXP INIT                                                            !
! - -- --- --- ------------------------------- ---- --------------------------!
!   CA     N   CONNECTION OF CU01                10BL                          !
!   CA 100 EXP CONNECT CU01 TO                                                 !
!   CA 110     USER    = 'USER'                                                !
!   CA 120     PASS    = 'PASS'                                                !
!   CA 130     LIB     = 'CIV'                                                 !
!   CA 140     NET     = 'U'                                                   !
!   CA 150     SESSION = SPACES                                                !
!   CA 160     SIZE    = 40                                                    !
! - -- --- --- ------------------------------- ---- --------------------------!
!   DA     N   CONNECTION OF CU02                10BL                          !
!   DA 100 EXP CONNECT CU02 TO                                                 !
!   DA 110     USER    = 'USER'                                                !
!   DA 120     PASS    = 'PASS'                                                !
!                                                                              !
! O: C1 CH: -P02                                                               !
!                                                                              !
-------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE  P PAFEX1 LIST PROPERTIES W/O SQL NAME        FUNCTION: 02   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!   DA 130     LIB     = 'CIV'                                                 !
!   DA 140     NET     = 'U'                                                   !
!   DA 150     SESSION = SPACES                                                !
!   DA 160     SIZE    = 1                                                     !
! - -- --- --- -------------------------------- ---- --------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -P02da                                                             !
!                                                                              !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE  P PAFEX1 LIST PROPERTIES W/O SQL NAME        FUNCTION: 21   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!          N   PROCESS                           05BL                          !
! - -- --- --- -------------------------------- ---- --------------------------!
!   BA     N   OPEN THE PROPERTY LIST            10BL                          !
!   BA 100 EXP OPEN CU01                                                       !
! - -- --- --- -------------------------------- ---- --------------------------!
!   CA     N   READ EACH PROPERTY                10DW CU01-FI = '0'            !
!   CA 100 EXP FETCH CU01                                                      !
! - -- --- --- -------------------------------- ---- --------------------------!
!   DA     N   SEARCH FOR RELATIONAL NAMES       15BL                          !
!   DA 100 *   OPEN RELATIONAL NAMES                                           !
!   DA 110 EXP OPEN CU02                                                       !
!   DA 200 *   READ RELATIONAL NAMES                                           !
!   DA 210 EXP FETCH CU02                                                      !
!   DA 300 *   NO RELATIONAL NAME : REPORT       99IT CU02-FT = '1'            !
!   DA 310 P   F8F                                                             !
!   DA 400 *   CLOSE RELATIONAL NAMES            99BL                          !
!   DA 410 EXP CLOSE CU02                                                      !
! - -- --- --- -------------------------------- ---- --------------------------!
!                                                                              !
! O: C1 CH: -P21                                                               !
!                                                                              !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! PROCEDURAL CODE  P PAFEX1 LIST PROPERTIES W/O SQL NAME        FUNCTION: 79   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!          N   DISCONNECT FROM PAF               05BL                          !
!      100 EXP CLOSE CU01                                                      !
!      110 EXP QUIT                                                            !
!      120 GFT                                                                 !
! - -- --- --- -------------------------------- ---- -------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! *** END ***                                                                  !
! O: C1 CH: -P79                                                               !
!                                                                              !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! REPORT DEFINITION.....: PRA                                                  !
!                                                                              !
! NAME..................: LIST PROPERTIES W/O SQL NAME                         !
!                                                                              !
! COMMENTS..............:                                                     !
!                                                                              !
! NATURE................: E REPORT                                            !
! PRINTER TYPE..........: L                                                    !
!                                                                              !
! LINE LENGTH...........: 132                                                  !
! FORMAT FOR TOTALS     : INTEGER........: 11                                  !
!                         DECIMAL PLACES.: 07                                  !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! EXPLICIT KEYWORDS..:                                                         !
!                                                                              !
!                                                                              !
!                                                                              !
! SESSION NUMBER.....: 2013        LIBRARY......: CIV    LOCK....:             !
!                                                                              !
! O: C1 CH: Rpra                        ACTION:                                !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! REPORT LAYOUT :         PRA LIST PROPERTIES W/O SQL NAME          LENGTH= 132!
!                                                                              !
! A LN CP S          1    1    2    2    3    3    4    4    5    5    6    6 !
!           1...5....0....5....0....5....0....5....0....5....0....5....0....5.!
!   00  1                                                                      !
!   03  2                          LIST OF PROPERTIES WITHOUT RELATIONAL NAME !
!   06  3                          ----------------------------------------- !
!   09  4    ------------------------------------------------------------------!
!   12  5    I  CODE  I           PROPERTY NAME              I INP.  FORM. I IN!
!   15  4    ------------------------------------------------------------------!
!   18  6    I        I                                      I             I   !
!   21  4    ------------------------------------------------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -L                                                                 !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! REPORT LAYOUT :         PRA LIST PROPERTIES W/O SQL NAME          LENGTH= 132!
!                                                                              !
! A LN CP S  6    7    7    8    8    9    9   10   10   11   11   12   12   13!
!           5....0....5....0....5....0....5....0....5....0....5....0....5....0!
!   03  2    E                                      PAGE:                     !
!   06  3    -                                                                !
!   09  4    ------------------------------------------------------------------!
!   12  5    INT. FORM. I U I        OUTPUT FORMAT        I PARENT I LIB I SESS!
!   15  4    ------------------------------------------------------------------!
!   18  6               I   I                             I        I     I     !
!   21  4    ------------------------------------------------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -L01C65                                                            !
--------------------------------------------------------------------------------
```

```
!                   APPLI  CICS/VSAM                        *PTJML.D474.CIV.2020!
! REPORT DESCRIPTION:       PRA LIST PROPERTIES W/O SQL NAME                     !
!                                                                                !
! A:   LINE LENGTH:    132 LI PAGE:   60 CAT TBL INST:    WR OPT:   SECTION: 00!
!      COMMENTS....:              CONDITIONS                                     !
!                                                                                !
! A CA LIN T TLI ST CP SKP FUSF COMMENTS      CONDITIONS                         !
!   BA 010        1 01 01*      HEADER        5-PR00-ALC NOT < 5-PR00-ALCM     !
!   BA 030          02 01                                                        !
!   BA 050          03 01                                                        !
!   BA 070          04 01                                                        !
!   BA 090          05 01                                                        !
!   BA 110          04 01                                                        !
! - -- --- - --- -- -- --- ---- ------------- ----------------------------------!
!   CA 010        2 06 01       DETAIL                                           !
! - -- --- - --- -- -- --- ---- ------------- ----------------------------------!
!   DA 010          04 01       FOOTER        5-PR00-ALC NOT < 5-PR00-ALCM OR  !
!   DA 020                                    CU01-FI = '1'                      !
! - -- --- - --- -- -- --- ---- ------------- ----------------------------------!
!                                                                                !
!                                                                                !
!                                                                                !
!                                                                                !
! O: C1 CH: -D                                                                   !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                        *PTJML.D474.CIV.2020
! REPORT CALL OF ELEMENTS  PRA LIST PROPERTIES W/O SQL NAME                      !
!                                                                                !
! A ST ELEM   L : STA C O W SOURCE    FLD CONDITION                        LIBR.!
!   01 XPAGE  0 : 112   M 5 PR00-APC                                       2013 !
! - -- ------ - : --- - - - ---------- -------------------------------- -----!
!   02 CDEL   0 :   3   M   CU01CDEL                                      2013 !
!   02 LDEL   0 :  12   M   CU01LDEL                                      2013 !
!   02 FDELT  0 :  51   M   CU01FDELT                                     2013 !
!   02 FDELI  0 :  65   M   CU01FDELI                                     2013 !
!   02 ODELUS 0 :  78   M   CU01ODELUS                                    2013 !
!   02 FDELO  0 :  82   M   CU01FDELO                                     2013 !
!   02 CDELP  0 : 112   M   CU01CDELP                                     2013 !
!   02 LCTX   0 : 121   M   CU01LCTX                                      2013 !
!   02 SCTX   0 : 127   M   CU01SCTX                                      2013 !
!   02 TCTX   0 : 131   M   CU01TCTX                                      2013 !
! - -- ------ - : --- - - - ---------- -------------------------------- -----!
!               :                                                              !
!               :                                                              !
!               :                                                              !
!               :                                                              !
!               :                                                              !
! *** END ***                                                                    !
! O: C1 CH: -CE                                                                  !
--------------------------------------------------------------------------------
```

3.3. ON-LINE EXAMPLE

ON-LINE EXAMPLE

OBJECTIVE:

- To list the Screens that do not conform to a particular local standard.

PAF Cursor Declarations:

- CU01 selects those Screen Definitions in which the Data Element presentation, initialization character, or Screen or Element help character differs from the standard.

Procedural logic:

- F02 : PAF initialization only for the first entry into the program (EIBCALEN = '0').

- F06CA: CU01 Cursor Connection. The user code and password are hard-coded but could be entered on a menu and passed via the COMMAREA. The VA Pac Database code must be specified (D474). In addition, the terminal code (EIBTRMID) is assigned to the PAF 'IDENT' parameter in order to ensure that the keys between conversations in the PAF file are unique. The 'SIZE' parameter is set to '10', which corresponds to the number of repeated lines on the screen. Only the number of screens necessary for display are read, in order to avoid on-line reads that are too long and may prove to be pointless if a user ends up exiting the transaction without consulting the whole list.

In this example, the F06 function is executed only on the first entry into the program. Thus, the CONNECT and OPEN statements are issued only once. This example does not take into account interactive modifications of selection criteria. This could be implemented by modifying specific screen-top category fields that are associated with the criterion of the SELECT statement defining the CU01 cursor.

- F06DA: Opening the CU01 Cursor. This statement causes screen definitions which do not conform to a local standard to be read and stored in the PAF work file.

- F06EA: The field to save the CU01 Cursor (CU01-SAVE) is transferred to a backup field in the COMMAREA.

- F4031: Closing the CU01 Cursor and executing the PAF CLOSE and QUIT statements.

- F52BA: Retrieving the CU01-SAVE field from the COMMAREA. If all records have been fetched, the Cursor is closed.

- F60PF: If the last record has not yet been fetched, and as long as the repetitive category of the screen is being processed, the next record is fetched.

- F60PQ: If the last record as been fetched, the "END" message is written and an exit from the iteration is performed.

- F75 : If no error is detected, the field to save the CU01 Cursor (CU01-SAVE) is transferred to a backup field in the COMMAREA.

```
--------------------------------------------------------------------------------
!                                                          *PTJML.D474.CIV.2020!
!                                                                              !
! DIALOGUE COMPLEMENT...................: PP PAF                                !
!                                                                              !
!                                                                              !
! COMMON AREA-DATA STRUCTURE CODE........: PF                                   !
!                                                                              !
! ERROR MESSAGE FILE CHARACTERISTICS.....:                                     !
!                      ORGANIZATION.....:                                      !
!                      EXTERNAL NAME....:                                      !
!                                                                              !
! FIRST SCREEN OF THE DIALOGUE...........:                                     !
!                                                                              !
! COMPLEMENTARY COMMON AREA LENGTH.......:                                     !
!                                                                              !
! CODE OF PSB OR SUB-SCHEMA..............:                                     !
!                                                                              !
!                                                                              !
! OPTIONS :                                                                    !
!                                                                              !
!                                                                              !
! SESSION NUMBER.....: 2013        LIBRARY......: CIV                          !
! *** END ***                                                                  !
! O: C1 CH: -O                          ACTION:                                !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! ON-LINE SCREEN DEFINITION..........: PAFEX2                                  !
!                                                                              !
! SCREEN NAME........................: LIST OF NON-STANDARD SCREENS            !
!                                                                              !
! SCREEN SIZE (LINES, COLUMNS) ......:   24       080                         !
! LABEL TYPE, TABS, INITIALIZATION...: * S      * 02                           !
! HELP CHARACTER SCREEN, DATA ELEMENT: * 11     * 12                           !
!                                                                              !
!                                     LABELS   DISPLAY INPUT ER.MESS. ER.FLD.  !
! INTENSITY ATTRIBUTE ...............:   N        N       N       B       B     !
! PRESENTATION ATTRIBUTE ............:   N        N       N       N       N     !
! COLOR ATTRIBUTE ...................:   W        W       W       W       W     !
!                                                                              !
! TYPE OF COBOL AND MAP TO GENERATE..:   0 * 0   IBM OS CICS (PROG. & MAP BMS) !
! CONTROL CARD OPTIONS FRONT & BACK..: * CC       (PROGRAM)   * KK       (MAP) !
! EXTERNAL NAMES ....................:   PF001P   (PROGRAM)     PF001M   (MAP) !
! TRANSACTION CODE...................:                                          !
!                                                                              !
!                                                                              !
! EXPLICIT KEYWORDS..:                                                         !
! SESSION NUMBER.....: 2013        LIBRARY......: CIV    LOCK....:             !
!                                                                              !
! O: C1 CH: Opf0010                    ACTION:                                 !
--------------------------------------------------------------------------------
```

```
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! ON-LINE SCREEN X-REF'S TO SCREENS FOR ON-LINE SCREEN : PAFEX2               !
!                                                                            !
! SCREEN : LIN  D.ELEM  P LN COL N P C HR VR  C P O SEG D.ELEM  W SEG D.ELEM LV!
! ---------------------------------------------------------------------------!
! ------- CALL OF  T  TYPE  (TITLE) -----------------------------------------!
! PAFEX2   020  PAFEX2  A 01 025 T     00 00                                 !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
!                                                                            !
! O: C1 CH: -XO                                                              !
------------------------------------------------------------------------------

------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! SCREEN CALL OF ELEM... PAFEX2 LIST OF NON-STANDARD SCREENS                 !
!                                                                            !
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES   . VALIDATION UPDATE .  DISPLAY      !
!       :        . P LN COL N L C HR VR  .  P V U UPD TARGET .  S SOURCE  LV !
! ...........................................................................!
!   010 : PFKEY  .          V            .    O V A          .    EN        !
!   011 :        .                       .      V S          .    01        !
!   012 :        .                       .      G            .    02        !
!   020 : PAFEX2 . A 01 025 T            .                   .              !
!   025 : PFVIEW .   02 025 V            .  R                .              !
!   040 :        .   02 015 L            .                   .              !
!   045 :        .      001 L            .                   .              !
!   060 : ODELL1 .   02 001 V            .  R                .              !
!   080 : ODELI1 .          V            .  P                .              !
!   100 : OSCRH1 .   01 001 V            .  P                .              !
!   120 : OSCRH2 .          V            .  P                .              !
!   130 :        .   01 001 L            .                   .              !
!   140 : RGROUP .   01 001 R 3      10  .                   .              !
!   160 : CSCR   .      003 P            .                   .    CU01      !
!   180 : LSCR   .      003 P            .                   .    CU01      !
!   200 : ODELLB .      003 P            .                   .    CU01      !
!                                                                            !
! O: C1 CH: -CE                                                              !
------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! SCREEN CALL OF ELEM... PAFEX2 LIST OF NON-STANDARD SCREENS                   !
!                                                                             !
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES   . VALIDATION UPDATE .  DISPLAY       !
!       :        . P LN COL N L C HR VR  .  P V U UPD TARGET .  S SOURCE   LV !
! ..........................................................................  !
!   220 : ODELIC .      003 P            .                   .    CU01        !
!   240 : OSCRHS .      003 P            .                   .    CU01        !
!   260 : OSCRHE .      003 P            .                   .    CU01        !
!   900 : ZGROUP . A 23 001 Z            .                   .                !
!   920 : ERMSG  .      001 P F          .                   .                !
!   940 :        . A 24 001 L            .                   .                !
!   960 :        .      001 L            .                   .                !
!   980 :        .      001 L            .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!       :        .                       .                   .                !
!                                                                             !
! O: C1 CH:                                                                   !
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! SCREEN CALL OF ELEM... PAFEX2 LIST OF NON-STANDARD SCREENS                   !
!                                                                             !
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES         . LABEL                        !
!       :        . P LN COL N L HR VR IN PR CO . T LITERALS                  !
! ..........................................................................  !
!   010 : PFKEY  .          V                 .                               !
!   011 :        .                            .                               !
!   012 :        .                            .                               !
!   020 : PAFEX2 . A 01 025 T                 .                               !
!   025 : PFVIEW .   02 025 V                 . I C                           !
!   040 :        .   02 015 L                 .   PLEASE ENTER THE STANDARD/  !
!   045 :        .      001 L                 .   SCREEN CHARACTERISTICS:/    !
!   060 : ODELL1 .   02 001 V                 . I S                           !
!   080 : ODELI1 .          V                 . I _                           !
!   100 : OSCRH1 .   01 001 V                 . I 11                          !
!   120 : OSCRH2 .          V                 . I 12                          !
!   130 :        .   01 001 L                 . A 079-                        !
!   140 : RGROUP .   01 001 R 3    10         .                               !
!   160 : CSCR   .      003 P                 .                               !
!   180 : LSCR   .      003 P                 .                               !
!   200 : ODELLB .      003 P                 .                               !
!                                                                             !
! O: C2 CH: -CE                                                               !
-------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
!                  APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! SCREEN CALL OF ELEM... PAFEX2 LIST OF NON-STANDARD SCREENS                   !
!                                                                             !
! A LIN : D.ELEM . PHYSICAL ATTRIBUTES         . LABEL                        !
!       :        . P LN COL N L HR VR IN PR CO . T LITERALS                  !
! ..........................................................................  .
!   220 : ODELIC .      003 P                  .                              !
!   240 : OSCRHS .      003 P                  .                              !
!   260 : OSCRHE .      003 P                  .                              !
!   900 : ZGROUP . A 23 001 Z                  .                              !
!   920 : ERMSG  .      001 P F                .                              !
!   940 :        . A 24 001 L                  .   ENTER: DISPLAY, PF1: SCRLL,/!
!   960 :        .      001 L                  .   PF2: EXIT, PF11: SCRN HELP,/!
!   980 :        .      001 L                  .   PF12: FIELD HELP/          !
!       :        .                             .                              !
!       :        .                             .                              !
!       :        .                             .                              !
!       :        .                             .                              !
!       :        .                             .                              !
!       :        .                             .                              !
!       :        .                             .                              !
!                                                                             !
! O: C2 CH: -CE                                                               !
-------------------------------------------------------------------------------

-------------------------------------------------------------------------------
!                    APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
!                                                                         S   !
! SEGMENT CALL OF ELEMENTS PF00 COMMUNICATION AREA                            !
!                                                                             !
! A LIN : ELEM.  INT.FORM. U OCC GR K CMD456 CONT VALUE/SFC UPD/TRGET DOC LIBR !
!   100 : PFSAVE X(115)    D                                                  !
!   110 : PFFRST X         D                                                  !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!       :                                                                     !
!                                                                             !
!                                                                             !
!                                                                             !
!                                                                             !
!                                                                             !
!                                                                             !
! O: C1 CH: Spf00CE                                                           !
-------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! WORK AREAS.........ENTITY TYPE O PAFEX2 LIST OF NON-STANDARD SCREENS        !
!                                                                              !
! CODE FOR PLACEMENT..:      BA                                                !
! A LIN T LEVEL OR SECTION WORK AREA DESCRIPTION                       OCCURS!
!   100 * LIST OF NON STANDARD SCREENS                                         !
!   110       EXEC PAF DECLARE CU01 CURSOR FOR                                 !
!   120       SELECT * FROM SCRDEF WHERE                                       !
!   130       ODELLB <> I-0020-ODELL1 OR                                       !
!   140       ODELIC <> I-0020-ODELI1 OR                                       !
!   150       OSCRHS <> I-0020-OSCRH1 OR                                       !
!   160       OSCRHE <> I-0020-OSCRH2 OR                                       !
!   170       END-EXEC                                                         !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -W                                                                 !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 02   !
!                                                                              !
! A SF LIN OPE OPERANDS                        LVTY CONDITION                  !
!          N   PAF INITIALIZATIONS             05IT EIBCALEN = 0               !
!      100 EXP INIT                                                            !
!      110 M   '0'     PF00-PFFRST                                             !
! - -- --- --- ------------------------------- ---- --------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -P02                                                               !
!                                                                              !
--------------------------------------------------------------------------------
```

```
--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 06   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!          N   PAF CONNECTION                    05IT PF00-PFFRST = '0'        !
! - -- --- --- ------------------------------- ---- --------------------------!
!   BA     N   SET FIRST TIME INDICATOR          10BL                          !
!   BA 100 M   '1'   PF00-PFFRST                                               !
! - -- --- --- ------------------------------- ---- --------------------------!
!   CA     N   CURSOR CONNECTION                 10BL                          !
!   CA 100 EXP CONNECT CU01 TO                                                 !
!   CA 110     USER    = 'USER'                                                !
!   CA 120     PASS    = 'PASS'                                                !
!   CA 130     LIB     = 'CIV'                                                 !
!   CA 140     SESSION = SPACES                                                !
!   CA 150     NET     = I-0010-PFVIEW                                         !
!   CA 160     SIZE    = 10                                                    !
!   CA 170     BASE    = 'D474'                                                !
!   CA 180     IDENT   = EIBTRMID                                              !
! - -- --- --- ------------------------------- ---- --------------------------
!   DA     N   OPEN CURSOR                       10BL                          !
!   DA 100 EXP OPEN CU01                                                       !
!                                                                              !
! O: C1 CH: -P06                                                             !
!                                                                              !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                        PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 06   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!   EA     N   SAVE CURSOR IN COMMAREA           10BL                          !
!   EA 100 M   CU01-SAVE PF00-PFSAVE                                           !
! - -- --- --- ------------------------------- ---- --------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -P06ea                                                           !
--------------------------------------------------------------------------------
```

```
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 40   !
!                                                                               !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                 !
!   31     N   END OF PAF PROCESSING             15BL                           !
!   31 100 EXP CLOSE CU01                                                       !
!   31 200 EXP QUIT                                                             !
! - -- --- --- ------------------------------- ---- --------------------------!
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
! O: C1 CH: -P40                                                                !
!                                                                               !
```

```
!                   APPLI  CICS/VSAM                       *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 52   !
!                                                                               !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                 !
!          N   REINITIALIZATION OF PAF AREAS     05BL                           !
!      100 M   PF00-PFSAVE CU01-SAVE                                            !
! - -- --- --- ------------------------------- ---- --------------------------!
!   BA     N   CLOSE IF END OF CURSOR            10IT CU01-FT = '1'             !
!   BA 100 EXP CLOSE CU01                                                       !
! - -- --- --- ------------------------------- ---- --------------------------!
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
!                                                                               !
! O: C1 CH: -P52                                                                !
!                                                                               !
```

```
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 60   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!   PF     N   FETCH SCREEN DEFINITION RECORDS  10IT CU01-FT = '0'            !
!   PF 100 EXP FETCH CU01                         AN CATX = 'R'                !
! - -- --- --- -------------------------------- ---- --------------------------!
!   PQ     N   DISPLAY END OF LIST MESSAGE      10IT CU01-FT = '1'            !
!   PQ 100 ERU 0001                                                           !
!   PQ 200 GFT                                                                !
! - -- --- --- -------------------------------- ---- --------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! O: C1 CH: -P60                                                               !
!                                                                              !
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
!                   APPLI  CICS/VSAM                      *PTJML.D474.CIV.2020!
! PROCEDURAL CODE       O PAFEX2 LIST OF NON-STANDARD SCREENS   FUNCTION: 75   !
!                                                                              !
! A SF LIN OPE OPERANDS                          LVTY CONDITION                !
!          N   SAVE CURSOR IF NO ERROR          05IT GR-EG = '1'              !
!      100 M   CU01-SAVE PF00-PFSAVE              AN CU01-FT = '0'            !
! - -- --- --- -------------------------------- ---- --------------------------!
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
!                                                                              !
! *** END ***                                                                  !
! O: C1 CH: -P75                                                               !
!                                                                              !
--------------------------------------------------------------------------------
```

```
-------------------------------------------------------------------------------
!                        LIST OF NON-STANDARD SCREENS                         !
!                                                                             !
!                        PAF EXTRACTOR VIEW: C                                !
!                                                                             !
!              PLEASE ENTER THE STANDARD SCREEN CHARACTERISTICS:              !
!                                                                             !
! LABEL TYPE........: S                  INIT. CHAR........: _                !
! SCREEN HELP CHAR..: 11                 FIELD HELP CHAR...: 12               !
-------------------------------------------------------------------------------
!  SCREEN   SCREEN NAME                          LABEL   INIT   SCREEN   FIELD  !
!  CODE                                          TYPE    CHAR   HELP     HELP   !
!                                                                CHAR     CHAR   !
!  XXXXXX   XXXXXXXXXXXXXXXXXXXXXXXXXXXXX       X       X      XX       XX     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
!  ......   .............................        .       .      ..       ..     !
! XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  !
! ENTER: DISPLAY, PF1: SCRLL,  PF2: EXIT, PF11: SCRN HELP,   PF12: FIELD HELP  !
-------------------------------------------------------------------------------
```

4. PUF - PACBASE UPDATE FACILITY

4.1. UPDP - BATCH MODE

INTRODUCTION TO PUF

A Specifications Database, with extractions of tables and columns carried out by PAF, can be updated by using the Pacbase Update Facility (PUF) procedure.

BATCH MODE - UPDP

From the sequential files, the UPDP procedure transforms the extractions in transactions used to update the Specifications Database.

For a description of this procedure, refer to the Operations Manual "Batch Procedures: User's Guide", Chapter "UPDP".

USER INPUT

The sequential file of input transactions is produced by a PAF extractor program. Its records mirror the PAF tables (described in the PAF Tables Manual).

Pos.	Length	Meaning
1	1	Transaction code (C, M, X, A or D, B)
2	10	PAF table code
12	299	PAF table contents (as described in the PAF Tables Manual).

UPDATE RULES

Update transactions are not sorted.

Each set of transactions impacting a library or session must be preceded by an ASSIGN table code line.

Pos.	Length	Value	Meaning
2	10	'ASSIGN'	Table code
12	8	uuuuuuuu	User code
20	8	pppppppp	Password
28	3	bbb	Library code
31	4	nnnn	Session number (blank=current)
35	1	'T'	Session status if test session
36	3	nnn	No line numbering
39	1	'F' or 'A'	Language code, useful if the transactions are not in the same language as the Database IN CASE OF A DSMS CONTROL OF THE DATABASE:
40	3	ppp	Product code
43	6	nnnnnn	Product number

When the update is performed while the TP is active (on platforms that support this function), the input transaction flow must be preceded by a CHECKP table code line.

Pos.	Length	Value	Meaning
2	10	'CHECKP'	Table code
12	4	nnnn	Number of transactions processed between two pauses or checkpoints
16	4	'UPDT'	Update procedure
20	2	nn	OS/2, UNIX, WINDOWS NT: Pause time, in seconds, between two update sets

4.2. ON-LINE MODE

ON-LINE MODE

In on-line mode, PUF updates, in real-time, large transactions in VisualAge Pacbase from a PAF user program. On certain platforms, this solves the problems of concurrent batch and on-line updating.

THE SERVICE

The PUF communication program receives and sends messages in middleware communication to satisty several applications.

The client application requests the following types of services:

* Entity extraction (download): the monitor sends the entity description in the communication area. There are two types of download:

- download with intention of updating: the download description is locked until the upload forbidding all other updating.

- download for consultation: no lock in place, the entity remains modifiable.

* Entity update (upload): the monitor receives an entity description which it transfers in the SYSPAF file when the last message is sent; it starts the uploading transaction and the request to unblock the entity. It then sends the erroneous transactions with errors. This service is stored in the CHOICE field, i.e. the communication field; it is 18 characters long and is coded as follows:

- UPDOWN (1 char.): D for download, U for upload, V V for block.

- SLASH (2 chars.): fixed value //

- COMET (1 char.): method code

- DLIST (2 chars.): LC for list by code, LT for list by type, LU for user list (only xxFCOM communication monitor)

- DSCODE (3 chars.): local entity code

- ENTITE (6 chars.): entity code

- DSCHXT (2 chars.): entity description selection ** for the whole of the entity, blank for the definition page

- UPDWV (1 char.): this variable depends on UPDOWN.

. in download = W is not intending to update and X if depending to update (lock positioned)

. in upload = U with unlocking at the end of updating, V without unlocking after updating.

* Locking/unlocking of a technical lock on certain entity descriptives.

THE PUF ON-LINE ENTRY POINT

The program suffixed by F000 is the distributor for the PUF on-line application. For the transactions transfer, the distributor uses the PA-suffixed workfile which is the PAF workfile. It sends the update transactions to different program folders.

The following recordings are used in the workfile:

- PA70 which contains the erroneous transactions sent by the user program to the distributor. They are cancelled by the distributor,

- PA80 which contains the erroneous transactions (with the gravity and label of the error), sent by the distributor to the user program. After analysis, the user program cancels PA80.

In the case of an ABEND, the distributor sends a message in the communication field to warn the user program. The message appears in the following format:

- ABEND

- return code (2 chars.):

  31 = problem with the PA file
  32 = problem with the VisualAge Pacbase files
  99 = problem with a program

- external name of the file or the program (8 chars.).

THE PAF CLIENT

The procedure for using PUF on-line is described in Chapter "Implementation in User Programs", Subchapter "Syntax of the SQL-PAF Language". This procedure includes the following specific statements:

- INSERT: storing information

- CALPUF: triggering updating by a distributor call,

and possibly:

- FETCHER: recovering errors signaled by PUF on-line in the PA80 file.

A DEPORTED CLIENT (REMOTE PUF)

While PUF is server, it can be used from a deported client: it is the Remote-PUF.

It is possible to use the middleware communications. The user must create his/her own communication monitor whose object will be to receive the message coming from the client and to transmit the information to the distributor. This information must be stored in the PAF workfile.

The dialog monitor has a communication field whose structure is the same as that of the screens generated by the Client/Server module. It is activated by the client application.

4.3. LIST OF STATEMENTS AND HOW THEY WORK

PUF CURSOR-INDEPENDENT STATEMENTS (ON-LINE ONLY)

To use PUF, refer to the specific INSERT, CALPUF and FETCHER statements, as well as the UPDATE clause and the SET statement described in Chapter "Implementation in User Programs", Subchapter "Syntax of the SQL-PAF Language".

On coming across the "UPDATE = pref" clause the following fields are generated:

```
01          pref-PUFCOM.
  05        pref-PUFID   PIC X(25)  VALUE SPACE.
  05        pref-PUFRET.
    10      pref-PUFPB   PIC X(05)  VALUE SPACE.
    10      pref-PUFRC   PIC X(02)  VALUE SPACE.
    10      pref-PUFNX   PIC X(08)  VALUE SPACE.
    10      pref-FILLER  PIC X(15)  VALUE SPACE.
  05        pref-PUFTRA  PIC X(04)  VALUE SPACE.
  05        pref-FILLER  PIC X(11)  VALUE SPACE.
```

PUFID is the identifier of the on-line conversation using PAF.

(similiar to the IDENT parameter of the CONNECT statement)

PUFRET contains the fields returned by PUF:

PUFPB : a problem was detected (ABEND or ERROR).
PUFRC : return code.
PUFNX : external name of the file responsible.
FILLER (15 chars. long) - not in use at present.

PUFTRA is the transaction code of the VA Pac Database

(similiar to the BASE parameter of the CONNECT statement)

```
01          pref-PAFCOM.
  05        pref-PAFID   PIC X(25)  VALUE SPACE.
  05        pref-PAFTRA  PIC X(04)  VALUE SPACE.
  05        pref-FILLER  PIC X(16)  VALUE SPACE.
  05        pref-PAFRC   PIC X(02)  VALUE SPACE.
  05        pref-ORDER   PIC X(01)  VALUE SPACE.
  05        pref-FI      PIC X(01)  VALUE SPACE.
  05        pref-FT      PIC X(01)  VALUE SPACE.
  05        pref-PUFERR.
    10      pref-PUFENR.
      15    pref-PUFMVT  PIC X(01)  VALUE SPACE.
      15    pref-PAFTAB  PIC X(10)  VALUE SPACE.
      15    pref-PAFEXT  PIC X(299) VALUE SPACE.
    10      pref-PUFGRE  PIC X(01)  VALUE SPACE.
    10      pref-PUFLIE  PIC X(66)  VALUE SPACE.
  05        pref-NUMERR  PIC 9(06)  VALUE ZERO.
  05        pref-NUMENR  PIC 9(06)  VALUE ZERO.
  05        pref-FILLER  PIC X(11)  VALUE SPACE.
```

PAFID is the identifier of the on-line conversation using PAF

(similiar to the IDENT parameter of the CONNECT statement)

PAFTRA is the transaction code of the VA Pac Database.

(similiar to the BASE parameter of the CONNECT statement)

FILLER (16 chars. long) - unused.

PAFRC is the return code of the PAF extractor

(70 on INSERT: the recording already exists)

ORDER is the statement code for the PAF extractor

("I" for INSERT; "E" for FETCHER)

FI is the end of reading errors:

(1 = read beyond the last recording; 0 = if not)

FT is the end of the process:

(1 = read beyond the last recording; 0 = if not)

And following the statement given to the PAF extractor ..

PUFERR will contain the recording read thanks to the FETCHER statement. Either the following five fields:

PUFMVT : code of erroneous transaction
PAFTAB : PAF table code of erroneous error
PAFEXT : erroneous transaction
PUFGRE : error gravity
PUFLIE : error label

Or ..

PUFENR will contain the error to write thanks to the INSERT statement. Either the following three fields:

PUFMVT : transaction code
PAFTAB : PAF table code of the transaction
PAFEXT : transaction

In all cases ..

NUMERR is the statement number of the "error" recording

NUMENR is the statement number of the recording to write; this field is to be input before INSERT statement and is supported by the user.

FILLER (11 chars. long) - unused.

5. PAF IMPLEMENTATION FOR VARIOUS ENVIRONMENTS

5.1. MVS/CICS VERSION

PAF IMPLEMENTATION FOR MVS/CICS

There are two variants for this system:

. VS COBOL variant (A-mode 24)
. COBOL II variant (A-mode 31)

The user has to choose the appropriate variant based on his/her usual development environment.

For BATCH processing, and only for the VS COBOL (24-bit) variant), LSR VSAM buffer management can be used to optimize access. In this case, the user has to include in the run JCL a one-line file (PACLSR) to parameterize physical access to the extractor.

The following is the recommended way to code the parameters:

```
PACLSR DD *
10,,15,30
```

If the user does not want to invoke the LSR routines, he/she just codes DUMMY on the PACLSR DD statement.

For ON-LINE processing, it is recommended to use the EIBTRMID field to identify the users in CONNECT statements (IDENT parameter).

Furthermore, the user can access several different databases from the same PAF program, in cases where several VA Pac databases coexist in the same region. The database calling code (4 characters), which is assigned to the BASE parameter in the CONNECT statement, makes it possible to select the database that the user wishes to query.

A typical CONNECT statement under CICS looks like this:

```
    EXEC PAF CONNECT C001 TO
         USER = ZC00-CUSR
         PASS = ZC00-CPSW
         LIB = ZC00-CLIB
         SESSION = ZC00-CSES
         NET = ZC00-TVIS
         SIZE = ZC00-LSCR
         IDENT = EIBTRMID
         BASE = ZC00-CBAS
```

5.2. CICS/DOS/VS VERSION

PAF IMPLEMENTATION FOR CICS/VSE

It should be noted that PAF implementation in batch mode uses a subprogram which must be link edited with user programs that are used only to make dynamic calls to the PAF access subprogram.

Furthermore, the SLI JPAFxxxx member contains the initialization step for the work file, as well as the DLBLs of the files in database xxxx.

A typical JCL jobstream looks like this:

```
     // DLBL L,'pacbase.modules.library'
     // LIBDEF PHASE,SEARCH=L.SUBA
     * --- STEP PROTOTYPE PAF ---
$ SLI MEM=JPAFPB80
     // DLBL APPLI,'application.file',,VSAM
     // EXEC PROTO,SIZE=AUTO
     /*
```

For ON-LINE processing, it is recommended to use the EIBTRMID field to identify the users in CONNECT statements (IDENT parameter).

Furthermore, you can access several different databases from the same PAF program, in cases where several VA Pac databases coexist in the same region. The database calling code (4 characters), which is assigned to the BASE parameter in the CONNECT statement, makes it possible to select the database that you wish to query.

A typical CONNECT statement under CICS looks like this:

```
    EXEC PAF CONNECT C001 TO
         USER = ZC00-CUSR
         PASS = ZC00-CPSW
         LIB = ZC00-CLIB
         SESSION = ZC00-CSES
         NET = ZC00-TVIS
         SIZE = ZC00-LSCR
         IDENT = EIBTRMID
         BASE = ZC00-CBAS
```

5.3. IMS/VS/ESA VERSION

PAF IMPLEMENTATION FOR IMS/VS/ESA

All IMS PAF programs, whether batch or on-line, must always call the on-line extractor. (The IMS version doesn't include a batch extractor. It's the on-line extractor that's used in both batch and on-line processing.)

- 'PBTPST' --> extractor for all entities except keywords.

- 'PBTPWS' --> extractor for keyword entities only (from version 8.0.1 on).

The user must routinely code the 'SET' statement in the WORKING-STORAGE SECTION of batch programs, as well as on-line ones if they've been developed using Batch Language. This statement should be coded as follows:

```
EXEC PAF SET TYPE = 01
END-EXEC
```

In addition, the user has to define for his/her application program a PSB that is to include the following:

. it may include the PCBs of the user databases,

. it must include the PCBs of the VA Pac system databases: AN, AR, AE and PA.

These PCBs can be listed in the PSB in any order, but they must be defined as follows:

```
                    .
                    .
PCB  TYPE=DB,DBDNAME=PACDAN$SUF,PROCOPT=GOT,KEYLEN=43
SENSEG  NAME=PAC7AN
PCB  TYPE=DB,DBDNAME=PACDAR$SUF,PROCOPT=GOT,KEYLEN=07
SENSEG  NAME=PAC7AR
PCB  TYPE=DB,DBDNAME=PACDAE$SUF,PROCOPT=GOT,KEYLEN=12
SENSEG  NAME=PAC7AE
PCB  TYPE=DB,DBDNAME=PACDPA$SUF,PROCOPT=A,KEYLEN=37
SENSEG  NAME=PAC7PA
                    .
                    .
                    .
```

where $SUF = the DBD suffix chosen when VisualAge Pacbase was installed.

Important note: Extractor call statements, which are generated by the preprocessor, have the following format:

```
CALL 'extractor' USING S-PCB-AN S-PCB-AR S-PCB-AE S-PCB-PA
                       cursor-name.
```

Therefore, the user has to give the same names to PCBs in the LINKAGE SECTION and the PROCEDURE DIVISION USING in batch programs (S-PCB-xx). For dialog screens, the definition screens of the PCBs called in the dialog PSB must be named PCB-xx. The dialog generator will add the prefix 'S-' in the source generated on the LINKAGE SECTION and PROCEDURE DIVISION USING level.

Doing Extractions under the Control of a Security System:

Entities can be extracted under the control of a security system (e.g., RACF). In this case, the extractor must be able to tell whether the program doing the extraction is batch or on-line. The method for controling the user code is actually different for a batch or an on-line program.

In batch processing, the user code, given in the CONNECT statement, is directly controlled, in relation to the security system, by means of an assembler program, PACSECB, which is transparent to the user.

In on-line processing, the user code is controlled, in relation to the one listed in the IO-PCB, by the security system.

In order to achieve this control, the extractor must know the type of the program calling it (batch or on-line). To do this, the MODE parameter must be coded in the SET order:

```
   - MODE = 'TP   ' <- the user program is an on-line
                       program
   - MODE = 'BATCH' <- the user program is a batch program
```

Important note: For an on-line program, extractor call statements, which are generated by the preprocessor, will appear as follows:

```
CALL 'extractor' USING S-PCB-AN S-PCB-AR S-PCB-AE S-PCB-PA
                       cursor-name S-IPCB.
```

For batch programs, call orders are generated the same way, with or without the control of a security system (without the S-IPCB parameter in the CALL).

5.4. DPS7 VERSION

PAF IMPLEMENTATION FOR DPS7

The Pre-processor is made up of the PAFP10 program, which is installed in the batch load module library.

Several different ways to process generated programs are available using PAF:

- using the GPRP procedure, which directly links the pre-processor to the generated stream (see the chapter devoted to GPR-, that is 'GPRT: Generation and Printing' Chapter in the Operations Manual 'VisualAge Pacbase 2.5 - Bull GCOS7 / TDS, Volume III'.

- using the PPAF procedure:

. by calling this procedure on optional before/after program control cards, combined with the compile-and-link JCL,

. by calling this procedure after executing the standard GPRT procedure, which produces the generated flow,

. by any other method that is best adapted to the site's specifications.

(See the Subchapter about PPAF in Chapter "Standard Utilities" in the Operations Manual, Part II, 'Batch Procedures'.)

Five PAF sub-programs are provided at installation time:

- Three sub-programs for batch, installed in the batch compil-units library:

. PBBTST for standard PAF requests,

. PBBTWS for PAF requests by keyword,

. PBBT98 for physical access to the VA Pac Database.

- Two sub-programs for on-line, installed in the on-line compil-units library:

. PBTPST for standard PAF requests,

. PBTPWS for PAF requests by keyword.

The work files necessary for PAF functioning are described in the Subchapter "Evolving Files" in the Chapter "VisualAge Pacbase Components".

An example of user batch program JCL that calls PAF can be found in member PAFJCL in library $NMLI.$LIBSRT.
This example contains all the required files for executing such a user program. The user can supply other batch work files besides the one provided at installation time since this file is allocated only for the duration of the JOB.

The name of the work file needed for on-line PAF functioning is dictated by TDS.

Since they can be used in writing programs that use the PAF module, the data element, data structure, and segment entities are provided as batch transactions in member PAFDICF in library $NMLI.$LIBSRT. The database administrator is responsible for putting this "PAF dictionary" into the database via the UPDT batch update procedure, after making sure that there's no conflict between the supplied entity codes and the entities that already exist in his/her network.

INTEGRATION OF PAF INTO A USER TDS

For PAF integration, a TDS source program (STDSPF) is provided in the library $NMTD.$LIBSL as well as in the JCL jobstreams used for generating and starting up this TDS in the library $NMLI.$LIBJCL.

To integrate PAF into a TDS, the user must perform the following functions:

1. Generate TDS.

   The user has to merge the existing STDS with the source code provided for PAF (STDSPF) by integrating the USE, SELECT, FILE-INTEGRITY, PROCESSING-MODE, and MESSAGE clauses, and the WORKING-STORAGE SECTION fields, into the required sections.

2. Allocate the PA file.

   The user has to run the PBINALPA procedure from the library $NMLI.$LIBJCL.

3. Start Up TDS.

   The user can run the PBEXTFPF procedure, but in this case, only the PAF files will be allocated and opened.

   To allocate and open all the files used by TDS applications, you have to merge the existing start-up JCL with the JCL provided with PAF (PBEXTDPF) by copying the 'ASSIGN' lines and the 'DEFINE' lines from the files used for PAF (AR, AN, AE and PA).

5.5. GCOS8 VERSION

PAF IMPLEMENTATION FOR GCOS8

EXAMPLES OF A PAF REQUEST PROVIDED ON THE INSTALLATION TAPE

A batch transaction file catalogued under '$UMCU/$MB.PAFT' is provided with this release. It contains examples of PAF requests for batch and on-line executions.

The first on-line example lists the Data Elements coming after a Data Element from a given sub-network. This request is executed from a menu screen and a list screen defining the context of the request: user code, password, library code, session number, number of Reads to store on the intermediary work file, view option, and data element code.

The second on-line example lists the selected entities for a given keyword. This request is executed from a menu screen and from a list screen identical to the WS screen in VisualAge Pacbase. The parameters are the same as those used in the first on-line example.

The Dialogue code is 'PA', and the screen codes are PA0000, PA0001, and PA0002. The external screen codes are PFP000, PFP001, and PFP002. The transaction and the sub-schema used to execute this dialogue are defined in the SYSGEN and the VA Pac WCL.

The first batch example lists all the information for a Data Element in a report with File Code-EQ. Program PAFEL1 executes this request. Sample JCL for the procedure used to execute this program is provided in the form of '9' cards in program TPAF.

This request is parameterized via a $ DATA temporary work file that contains a line defining the connection parameters.

```
COLUMN  1              :   *
COLUMN  2 -  9         :   User code
COLUMN 10 - 17         :   Password
COLUMN 18 - 20         :   Library code
COLUMN 21 - 25         :   Session number/type
COLUMN 26              :   Sub-network view
COLUMN 27 - 32         :   Number of intermediate Reads
```

```
The lines below are used to select the requested data ele-
ments:

   COLUMN 1               :   E
   COLUMN 2 - 7           :   Code of selected data elment
```

The second batch example prints all entities with references to keywords chosen in a report with File Code-EQ. Program PAFWS1 executes this request. Sample JCL for the procedure used to execute this program is provided in the form of '9' cards in program WPAF.

The PAF request is parameterized via a $ DATA intermediary work file that contains three lines defining the request:

--the first is an '*' type line that's identical to the one in the last request,

--the second indicates the keyword type as well as the entity you're searching for.

```
   COLUMN 1               :   W
   COLUMN 2               :   Keyword type
                          :    'M' = explicit keyword
                          :    'L' = implicit keyword
                          :    ' ' = both
   COLUMN 3 - 4           :   User Entity Occurence call
                          :   type if entity type is UEO
   COLUMN 5 - 7           :   Entity type
                          :    '   ' = all entities
                          :    'XXX' = for entity codes,
                          :            see PAF manual

--the third line formulates the request in the same way as
the WS screen in VisualAge Pacbase.

   COLUMN 1               :   X
   COLUMN 2 - 80          :   Keyword combinations
```

This whole set of transactions must be inserted into a Library whose TYPE OF COBOL TO GENERATE option = 'N'; the programs must be generated from a sub-library whose TYPE OF COBOL TO GENERATE option = '5'.

Since all of the entities required to execute the test deck are provided, the network chosen for installation can be completely independent of the other networks.

All of the PAF entities are provided with this Release in the form of a batch transaction file catalogued under '$UMCU/$MB.PAFD'.

>>> Refer to Operations Manual 'VisualAge Pacbase 2.5 - Bull GCOS8 / DMIV', Subchapter 'Installation Process'.

5.6. UNISYS 2200 VERSION

PAF IMPLEMENTATION FOR UNISYS 2200

The PPAF procedure is used to process the generated programs which use the PAF function (refer to the corresponding Subchapter in Chapter "STANDARD UTILITIES" in the Operations Manual Part II "BATCH PROCEDURES").

The program obtained via the PPAF procedure must be compiled with the UCOB compiler.
The user program uses specific PAF access sub-programs, and the VisualAge Pacbase Database access sub-programs. So you must specify to the linker, either for a static link (RESOLVE USING clause of link) or for a dynamic link (SEARCH clause of SSDP), the files which contain these sub-programs, i.e.:

```
    For PAF sub-programs:
        PBBTST and/or PBBTWS in $QUAL*$LIBRELB


    For accesses to the VA Pac Database (DMS or SFS)
        PUINDX              in $QUAL*$LIBBASE

    For accesses to the Database (all  supports)
        PUACCESS            in $QUAL*$PACSSCH
```

PACBASE DICTIONARY

The Data Element, Data Structure, and Segment entities, which can be used to write programs which use the PAF function, are provided as batch transactions.

IMPORTANT:

The introduction of this "PAF Dictionary" in the VA Pac Database via the UPDT batch procedure is under the responsability of the Database manager.

5.7. VISUALAGE PACBASE OS2 AND WINDOWS/NT VERSION

COMPLEMENT: PAF ENVIRONMENT INSTALLATION

The PAF Function processes SQL requests, written in user programs, for access to the VA Pac Database, by the generation of data and sub-programs in the COBOL source code generated from these programs.

The pre-processor processes the generated programs to perform this transformation. The pre-processor is made up of the PAFP10.EXE program which is installed in the batch program directory, ¥BATCH¥PGM, of the VA Pac servers.

The PPAF procedure processes the user's generated programs that use PAF (refer to the corresponding Subchapter in Chapter STANDARD UTILITIES of the Batch Procedures: User's Guide.

THE EXTRACTION SUB-PROGRAMS

The PAF user programs (batch or on-line), generated with a '3' variant (to comply with COBOL Micro Focus) use the same extraction sub-programs. There are three extraction sub-programs:

. PBBTST (for standard extractions)

. PBBTWS (for keyword extractions)

Both dynamically called by the PAF user programs, and

. PBBT98, which is dynamically called by the extractors (PBBTST and PBBTWS) to access the VA Pac Database and to the PAF workfile.

The three sub-programs which are dynamically called are supplied compiled and linked (.DLL files) and are in the format of a COBOL source (.CBL files). The .DLL files are installed in the batch programs' directory, ¥BATCH¥PGM of the VA Pac servers.

Example of compilation: CBLLINK -D PAFRUB.CBL

The COBOL source files of the extractors are supplied in the PAFCBL directory. The extractors which are dynamically called must be compiled and linked on the site when the release of the site's Micro Focus compiler is not compatible with that used for the VA Pac release.

The COBOL.DIR file contains the compilation instructions used to compile extraction sub-programs.

The READ.ME file contains documentation on the compiler used for VA Pac. It should be read carefully.

PAF DICTIONARY

PAF Dictionary is described with occurrences of Data Structures, Segments and Data Elements that will be used to write programs calling the PAF Function. These occurrences are supplied as batch update transactions in the MBUPDT.PAF file, copied in the METHOD sub-directory during the installation.

The introduction of this "PAF dictionary" in the VisualAge Pacbase Database is the responsibility of the VisualAge Pacbase Database Administrator, who must perform the following beforehand:

1. Check that these occurrences do not conflict with

   occurrences existing in the network.

2. Copy the file ¥METHOD¥MBUPDT.PAF into MBUPDT in the ¥INPUT¥'db_name' directory.

3. Modify the '*'-type line in the file MBUPDT in ¥INPUT¥'db_name'.

In order to avoid dictionary compatibility problems with the entities supplied for the PAF function, it is advised to create an independant sub-network of libraries in which the PAF utilities will be written.

EXAMPLE OF COMPILATION AND LINK OF A PAF PROGRAM

The purpose is to compile the batch program PAFRUB.
The compiler used for VA Pac is Microfocus 4.0.nn. It is advised to use the same compiler for the PAF programs to avoid conflicts between Micro Focus libraries.

Compilation instructions (COBOL.DIR file):

```
ASSIGN "EXTERNAL"
SEQUENTIAL "LINE"
```

Compilation and link command:

```
CBLLINK -E PAFRUB.CBL
```

EXAMPLE OF AN EXECUTION PROCEDURE USING PAF

The user wants to execute the batch program PAFDEL. The following installation parameters have been chosen:

. Release = VAPAC
. Database name = TEST
. The programs and the ¥ASSIGN directory are installed on C:.

```
ECHO OFF
CLS

ECHO *** Delete previous PAF files ***
ECHO DEL C:VAPAC PAF WPAF.*

ECHO *** Assignment of PACLAN and PAF files ***
CALL C:¥VAPAC¥ASSIGN¥TEST¥PAC7AE.CMD
CALL C:¥VAPAC¥ASSIGN¥TEST¥PAC7AN.CMD
CALL C:¥VAPAC¥ASSIGN¥TEST¥PAC7AR.CMD
SET SYSPAF=C:¥VAPAC¥PAF¥WPAF

ECHO *** Assignment of user files ***
REM * Add user program's specific files *

ECHO Execution of PAFRUB
PAFDEL
IF ERRORLEVEL 1 GOTO ERROR
ECHO End of extraction
GOTO END

:ERROR
ECHO PAFDEL Execution error
:END
ECHO ON
```

5.8. VISUALAGE PACBASE FOR UNIX VERSION

PAF IMPLEMENTATION FOR VISUALAGE PACBASE FOR UNIX

EXTRACTION SUB-PROGRAMS

For user programs generated with variant '3' of TYPE OF COBOL TO GENERATE (to comply with PC/MICROFOCUS COBOL), the same extraction sub-programs are used for both batch and on-line processing. These extraction sub-programs are supplied compiled and linked (.gnt files) and in the format of COBOL sources (.cbl files).

There are three extraction sub-programs:

. PBBTST (for standard extractions) and PBBTWS (for keyword extractions) are dynamically called by the PAF user programs.

. PBBT98 is dynamically called by the extractors (PBBTST or PBBTWS) for access to the VisualAge Pacbase for UNIX Database and to the PAF workfile.

The compiled files of the extractors are supplied in the batch programs' directory of the VisualAge Pacbase for UNIX servers ($PACDIR/batch/gnt).

The COBOL source files of the extractors are supplied in the $PACDIR/pafcgi directory. These sub-programs must be compiled on the site when the release of the site Micro Focus compiler is different from the one used for VisualAge Pacbase for UNIX.

The release of the Micro Focus compiler used for VisualAge Pacbase for UNIX is:

. "3.1" for TANDEM, DPX/20 and DEC/OSF1 platforms

. "3.1" or "3.2" for the HP9000 platform

. "4.0" for OSF and SUN platforms

. "3.2" or "4.0" for the RS6 platform

. "8.15" or "8.75" for the HP platform.

THE PAF DICTIONARY

Occurrences of Data Structures, Segments, and Data Elements used with the PAF Function are supplied as batch update transactions.

The introduction of this "PAF Dictionary" in the PACBASE Database via the UPDT batch procedure is under the responsability of the Database Administrator.

COMPILATION AND EXECUTION OF PAF PROGRAMS

The $PACDIR/pafcgi directory includes a sample script for the compilation and another one for the execution of a PAF program. They are called "pafcomp" and "pafrun". It is advised to copy this directory into a work directory, such as "pafuser" via the command:

```
cp -r $PACDIR/pafcgi/pafuser
```

and to copy the generated PAF programs into this work directory.
Then, from this directory, edit and run the compilation and execution scripts.

You may use the compilation script to compile the PAF system sub-programs.

EXECUTION OF A PAF EXTRACTOR

Before executing the PAF extractor, you should perform the following file assignments:

```
. Permanent input files:
  - VisualAge Pacbase data file   : PAC7AR
  - VisualAge Pacbase index file  : PAC7AN
  - Error message file            : PAC7AE

. PAF workfile                    : SYSPAF
. User files (when necessary).
```

6. ERROR MESSAGES

6.1. THE PAF TRANSLATOR

THE PAF TRANSLATOR

The PAF Translator can detect a number of syntax errors in SQL-PAF statements. Each error, including the corresponding error message and the line number which identifies the beginning of the PAF sequence in the translated program, is printed in an output report.

The possible error messages are listed below, including explanatory comments, in some cases.

UNKNOWN COLUMN CODE : <column-code>

The <column-code> does not identify a table column specified in the FROM clause (in the selected language).

TOO MANY ELEMENTARY CONDITIONS IN SELECT CLAUSE

There are more than 50 elementary conditions in this SQL-PAF query.

CURSOR CODE IS TOO LONG : <cursor-code>

The cursor code must contain four characters.

CURSOR CODE ALREADY DECLARED : <cursor-code>

TOO MANY CURSORS DECLARED

There are more than 100 cursors declared in this SQL-PAF query.

UNKNOWN CURSOR CODE : <cursor-code>

There is a cursor management statement for a cursor which has not been declared in the PAF user program.

NO CONNECT STATEMENT FOR CURSOR : <cursor-code>

NO OPEN STATEMENT FOR CURSOR : <cursor-code>

NO FETCH STATEMENT FOR CURSOR : <cursor-code>

NO CLOSE STATEMENT FOR CURSOR : <cursor-code>

NO INIT STATEMENT FOR CURSOR : <cursor-code>

THE PAF SEQUENCE IS TOO LONG

A PAF sequence is a series of lines grouped between EXEC PAF and END-EXEC. The maximum number of these lines is 50.

END OF PROGRAM DURING A PAF SEQUENCE

OPERAND CANNOT BE NUMERIC : <operand>

OPERAND CANNOT BE ALPHANUMERIC : <operand>

INVALID OPERAND LENGTH : <operand>

Alphanumeric constant operands have a maximum length of 120 characters.

INVALID COBOL OPERAND : <operand>

COLUMN TYPES ARE DIFFERENT : <col1-code> <col2-code>

An elementary condition applies in the comparison of a numeric column with an alphanumeric column.

LEFT PARENTHESIS MISSING
RIGHT PARENTHESIS MISSING

Elementary conditions which follow the keyword WHERE must be enclosed between balanced parentheses, i.e., the number of LEFT parentheses must equal the number of RIGHT parentheses.

NO QUIT STATEMENT IN PAF-USER PROGRAM

SYNTAX ERROR : <erroneous-syntax>

Incorrect syntax for the SQL-PAF language.

INVALID LITERAL LENGTH : <literal>

The maximum length of a literal is 120 characters.

TOO MANY LITERALS ON A SINGLE LINE

The number of literals on a PAF sequence line must not exceed 40.

INCORRECT ENDING OF LITERAL : <literal>

UNKNOWN TABLE CODE : <table-code>

The <table-code> does not identify a PAF table (in the selected language).

UNKNOWN UEO TYPE CODE (WRONG TABLE CODE): <UEO-type-code>

The User Entity Occurrence Table code is incorrect because the UEO Type code, used to build the generic Table code, does not exist (in the selected sub-network).
For more details, refer to the description of the User Entity Occurrence Tables.

UNKNOWN UEO DESCRIPTION (WRONG TABLE CODE) : <DSn>

The code of the User Entity Occurrence Table is incorrect since the specified User Entity Description Number does not exist.
For more details, refer to the description of the User Entity Occurrence Tables.

INVALID PACBASE CONNECTION PARAMETERS

INVALID STRING DELIMITER : <delimiter>

The delimiter specified in the SET statement must have a value of either SINGLE (single (') quotes) or DOUBLE (double (") quotes), respectively.

INVALID EXECUTION MODE : <execution-mode>

The execution mode specified in the SET statement must have a value of either BATCH or TP.

INVALID GENERATION VARIANT(S) : <generation-variant(s)>

The generation variant(s) specified in the SET statement must be VisualAge Pacbase variant(s).

THERE SHOULD NOT BE A CONDITION ON COL.: <column-number>

There can't be any conditions on columns 05 (VA Pac entity code), 06 (VA Pac entity label), and 07 (explicit entity keywords) of the KEYWORD table.

THERE SHOULD NOT BE SEVERAL CONDITIONS ON COL. 01

There can't be more than one condition on column 01 (entity type) of the KEYWORD table.

THERE SHOULD NOT BE SEVERAL CONDITIONS ON COL. 02

There can't be more than one condition on column 02 (UE type code) of the KEYWORD table.

THERE SHOULD NOT BE SEVERAL CONDITIONS ON COL. 03

There can't be more than one condition on column 03 (keyword type) of the KEYWORD table.

THERE SHOULD NOT BE SEVERAL CONDITIONS ON COL. 04

There must be one and only one condition on column 04 (WS search argument).

THERE SHOULD BE A CONDITION ON COLUMN 01

If there is an elementary condition on column 02 (UE type code), then there must be a condition on column 01 (entity type).

INCORRECT COMPARATOR FOR COLUMN: <column-number>

Only the '=' operator can be used in the elementary conditions of a KEYWORD table query.

INCORRECT OPERAND FOR COLUMN: <column-number>

The operand in the elementary condition of a KEYWORD table query mustn't be another column of the table.

ACCESS TO PAF IS NOT ALLOWED

Check the access key.

6.2. THE PAF EXTRACTOR

THE PAF EXTRACTOR

ERROR CODES RETURNED BY THE EXTRACTOR SUB-PROGRAM

The <cursor-code>-RETCOD field, generated by the PAF Translator, contains the error code returned by the PAF Extractor Sub-Program.

The '00' value in this field indicates that no error was detected.

There are three types of errors:

. SQL-PAF statement sequence errors,

. File access errors,

. Errors in extracted data.

SQL-PAF STATEMENT SEQUENCE ERRORS

Return Codes : 01 to 10

The following chart summarizes the errors which can occur in the sequence of SQL-PAF statements.

Statements on lines under (1) precede statements in columns under (2).
"NULL" means that no prior statement has been issued.

When there is a sequence error, the corresponding box contains the return code value.

EXAMPLE: The first line indicates that no statement can be issued before the INIT statement.

(1) \ (2)	INIT	CONNECT	OPEN	FETCH	CLOSE	QUIT
NULL		01	01	01	01	01
INIT	02		03	04	05	
CONNECT	02			06	07	
OPEN	02		08			
FETCH	02		08			
CLOSE	02			09	10	
QUIT		01	01	01	01	01

RETURN CODE VALUES AND MEANING

01	Initializations	not performed.
02	Initializations	already performed.
03	OPEN	of an unconnected cursor.
04	FETCH	of an unconnected cursor.
05	CLOSE	of an unconnected cursor.
06	FETCH	of an unopened cursor.
07	CLOSE	of an unopened cursor.
08	OPEN	of an unclosed cursor.
09	FETCH	of a closed cursor.
10	CLOSE	of a closed cursor.

FILE ACCESS ERRORS

File access errors occur in relation to the VisualAge Pacbase Database files (Index, Data, and Error Messages) and the Temporary Work File.

RETURN CODE VALUES AND MEANING

21 Open error on Index File,

22 Open error on Data File,

23 Open error on Error Message File,

24 Open error on Temporary Work File,

31 Read/Write error on Temporary Work File,

32 Read error on VisualAge Pacbase File,

40 VisualAge Pacbase Database Connection error.

41 Unauthorized use of PAF (access key).

ERRORS IN EXTRACTED DATA

Return Code: 50

Errors in extracted data occur with numeric columns in User Entity Occurrence Tables. This happens when the internal format of a Data Element which describes a User Entity can be modified even though Occurrences of that User Entity have already been defined and described. Thus, the content of the column associated with this Data Element can be alphanumeric instead of numeric.

In such cases, the query is still valid. It is when the FETCH statement is issued that the Extractor returns error code '50'.

7. PRESENTATION OF THE PAF-PDM FUNCTIONS

7.1. FOREWORD

FOREWORD

The PAF Function and the PDM Extension support functions which may be used jointly.

They do not replace the initial PAF and PDM functions but enhance them as they co-operate.

Hereafter, they will be referred to as PAF-PDM functions.

NOTE: The PAF-PDM functions may also be used independently of each other.

These functions are therefore sub-divided into PAF+ and PDM+.

PAF+ is documented in the PAF Reference Manual.
PDM+ is documented in the Personalized Documentation Manager Reference Manual.
The following page lists all the manuals and documents which may be necessary when using the PAF-PDM functions.

Using PAF-PDM requires an in-depth knowledge of the VisualAge Pacbase metamodel and (if installed) the metamodels of the WorkStation's Pacdesign or Pacbench modules (specific to the methodology in use).

DOCUMENTATION

Listed below is the exhaustive list of manuals and documents which may be necessary when using PAF-PDM:

1. PAF Reference Manual, with an appendix containing two examples of Extraction Master Paths including the execution reports printed by the XPAF Validation procedure (Ref: DD PAF).

2. PAF Tables / Host (Ref: DD PAG).

3. PAF Tables for WorkStation User Entities:

   DESCRIPTION OF USER ENTITIES DEDICATED TO THE WORKSTATION

   See sub-chapter 'Introduction to the PAF function', section 'Description of tables'.

4. Personalized Documentation Manager Reference Manual (Ref: DD PDM).

5. PDM+ Examples (Ref: DD PDX):

. Volume calling a Master Outline,

. Master Outline Validation report printed by the XPDM procedure,

. Volume Print report printed by the GPRT procedure,

. Print and Specific Layouts used by the Volume.

6. VisualAge Pacbase Operations Manuals.

7.2. OBJECTIVES OF PAF-PDM FUNCTIONS

OBJECTIVES OF THE PAF-PDM FUNCTIONS

AUTOMATIC STRUCTURING AND MAINTENANCE OF DOCUMENTS

The initial purpose of PAF-PDM is to add new functionalities to the PDM extension.
The underlying principle of these functionalities is to make the most of the cross-references between entities in the metamodel.

EXAMPLE:
Metamodel of VisualAge Pacbase platform, PACBENCH module:

You want to document an on-line application and, therefore, print the documentation attached to all occurrences involved in this application's development.

To simplify, we will only consider the part of the Volume which documents one Screen of the application. Documentation may be found:

. On the General Documentation (-G) of this Screen,

. On the General Documentation (-G) of its Segments,

. On the Description (-D) of the Data Elements called by these Segments.

With PDM, you have to write each individual Segment call into the Volume Description (-D).
As a result, when a new Segment is called in the Screen (-CS), it must also be added in the Volume Description.

With PAF-PDM, you specify the information to be printed in the Volume by defining -- once and for all -- an Extraction Master Path, also called PTEx.

In the example shown above, the extraction path will start with the Screen entity, find the called Segments, and finally work its way down to the Data Elements, its course being guided by the metamodel's cross-references.

PAF-PDM is therefore a tool not only for the automatic documentation of applications, but also for the automatic maintenance of this documentation. When the documented application is modified, you only have to re-generate the relevant Volumes without having to change their Description.

DOCUMENTATION STANDARDIZATION

PDM+ allows you to write Master Outlines (PTEds), i.e. skeletons which may be used for various purposes:

. With PDM, print options assigned to a Volume apply to this particular Volume only. As a result, options must be specified in each Volume which makes standardization not an easy task.

PDM+ allows you to specify all relevant print options in one Master Outline. This PTEd is then called by as many Volumes as needed.

. Coding standardized calls is another PDM+ facility.

```
For instance, in a Master Outline, the following call
TGEN___D_
```
will cause all Text occurrences whose code starts with the letters 'GEN' to be printed in ALL Volumes where this Master Outline is invoked.

. Furthermore and most important, standardization in documentation structuring is achieved with PDM+, in co-operation with PAF+.

It is in this framework that the expression Master Outline takes on its full meaning as the PTEd becomes a structural skeleton. The generation of different Volumes documenting one or several applications may be based on only one Master Outline (PTEd) managing data extracted by one PTEx.

NOTE: A Volume may call several Master Outlines.

CONCLUSION

With PAF-PDM, automatic documentation structuring and standardization are not synonyms of strictness and rigidity since data extraction and printing are completely user-defined.
However, this definition should come from one authoritative entity, failing which standardization may prove a vain word.

7.3. OPERATING MODE OF PAF-PDM FUNCTIONS

PAF-PDM OPERATING MODE

The PAF+/Extraction and the PDM+/Outline can be used separately or together.

PAF+ allows for the writing of the Extraction Master Path (PTEx) and for its execution when it is a User Extractor.

PDM+ allows for the writing and execution of the Master Outline (PTEd).

When both functionalities are used jointly, the Master Outline calls an Extraction Master Path of the Macro-Command type.

. When the PAF+/Extraction function is used alone, it allows for the generation of User Extractor programs with the possibility of formatting the data extracted.

. When the PDM+/Outline function is used alone, it allows for standardization skeletons (standard Print Options, Text occurrences always called, standardized calls).

. When both functions are used together, PAF+ extracts data from the Database. This data is processed by PDM+ and finally printed in a Volume.

PAF+: THE EXTRACTION MASTER PATH

PAF+ allows you to write an Extraction Master Path (PTEx), i.e. an exploration course throughout the Specifications Database, from which a data extraction program is automatically generated.

The writing of a PTEx means defining and describing an occurrence of a dedicated User Entity coded '.PPTEX' and whose type code is 7E (CH: $7E.....).

There are two types of PTEx, therefore occurrences of the '.PPTEX' User Entity may be of either one of the following types:

. Type 'M' allows you to generate a Macro-Command, i.e. a sub-program which will have to be called in a Master Outline (PTEd). See the Personalized Documentation Manager Reference Manual for a complete documentation on the PDM+ Functionality.

. Type 'E' allows you to generate a User Extractor program executed independently.

The input of an occurrence of an Extraction Master Path User Entity is documented in the PAF Reference Manual, Chapter "Extraction Master Path: Definition / Description".

VALIDATION

The Extraction Master Path must then be validated by the XPAF batch procedure which generates the User Extraction Program or the Macro-Command Sub-Program.

The XPAF procedure is documented in the Operations Manual, Part II - "Batch Procedures", Chapter "Utilities".

When no error is detected, the validation produces a COBOL source program which must be compiled and linked to be executed.

OPERATIONS

. Execution of a User Extractor (E-type PTEx):

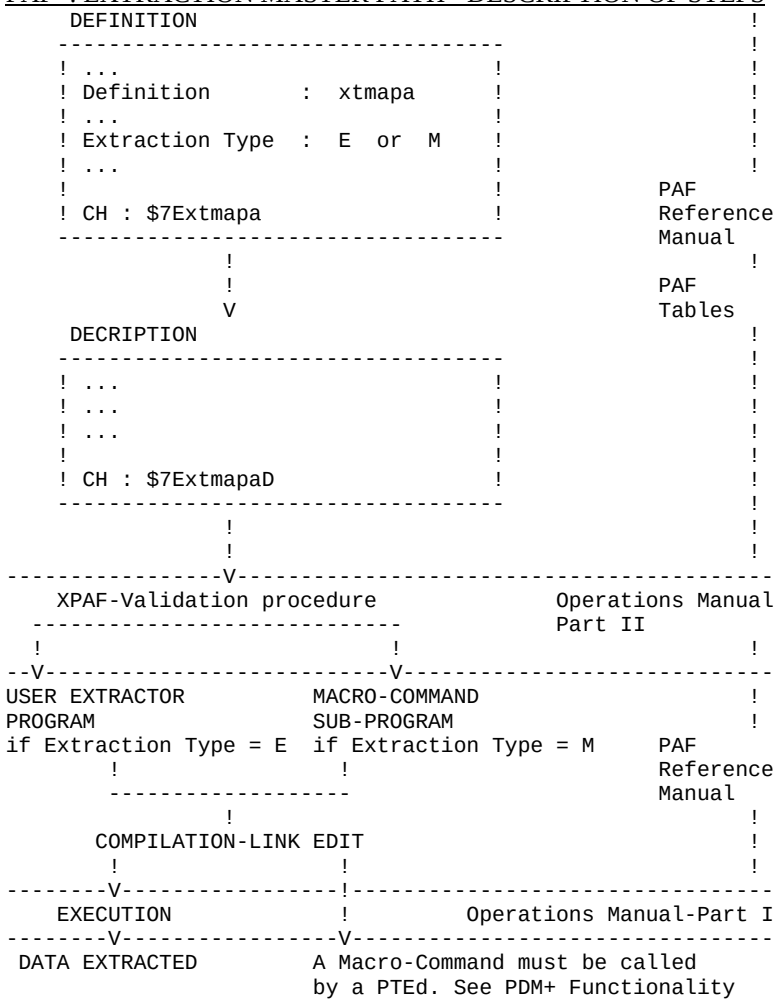
Once validated, compiled, and linked, a User Extractor is ready for execution.
User Input is described in the PAF Reference Manual, Chapter "EXECUTION OF A USER EXTRACTOR / E-Type PTEx".
For technical information on this execution, please refer to the Operations Manual, Part I, Chapter "INSTALLATION", Subchapter "COMPLEMENT: EXECUTION OF A PAF+ USER EXTRACTOR".

. Execution of a Macro-Command (M-type PTEx):

Once validated, compiled, and linked, a Macro-Command is not ready for execution. It must be called in a Master Outline.
See the Personalized Documentation Manager Reference Manual for a complete documentation on the PDM+ Functionality.

NOTE: An Extraction Master Path is independent of the Database in which it is defined and described as long as the root is the same.

PAF+: EXTRACTION MASTER PATH - DESCRIPTION OF STEPS



PDM+: THE MASTER OUTLINE

PDM+ allows to write Master Outlines (PTEd) supported by occurrences of the P-type Volume entity.

A PTEd organizes the printing of data extracted by PAF+ if its Description calls a PTEx on an M-type line.

A Master Outline must be validated by the XPDM batch procedure.

> The XPDM procedure is documented in the Operations Manual, Part II - "Batch Procedures", Chapter "Utilities".

A Master Outline cannot be printed.
Another Volume (other than P-type) must call the PTEd on a P-type Description line.

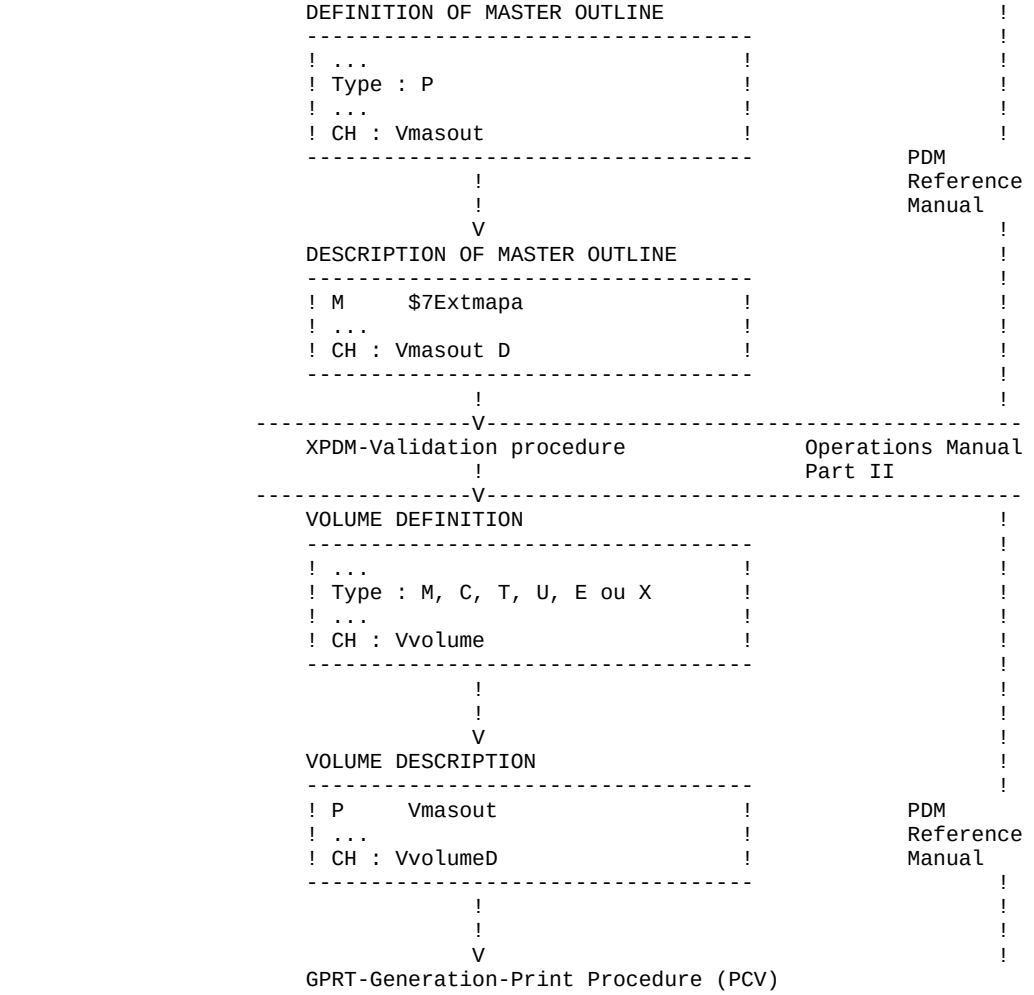
All that is necessary then, is to generate the Volume calling the Master Outline with the GPRT PCV command.

NOTE: A Master Outline is independent of the Database in which it is defined and described as long as the root is the same.

If an Extraction Master Path is modified and then re-validated (XPAF), all Master Outlines calling this Extraction Master Path may have to be modified and must always be re-validated (XPDM).

PDM+: THE MASTER OUTLINE - DESCRIPTION OF STEPS

The chart below presents PAF-PDM used jointly: the Macro-Command Sub-Program is called in a Master Outline.



8. EXTRACTION MASTER PATH: DEFINITION / DESCRIPTION

EXTRACTION MASTER PATH DEFINITION / DESCRIPTION

An Extraction Master Path is defined and described on an occurrence of a User Entity dedicated to PAF+/Extraction. The code of this User Entity is '.PPTEX' and its call type is '7E'. It is supplied with the product and must not be modified.

The following pages explain the Definition screens (CH: $7E.....) and Description screens (CH: $7E.....D) of the User Entity Occurrences. A brief description of the input fields is also included.

The values given in the definition and the description of the User Entity occurrences are not checked upon user input. For example, if a field that is listed as required in the following pages is not filled in, the user will not be given an error message.

It is advised to document the Extraction Master Path by drawing an extraction tree on the General Documentation screen (-G) of the occurrence and by writing comments on the Description screen. Comments will automatically be printed when the validation procedure of the Extraction Master is submitted.

EXTRACTION MASTER PATH: DEFINITION / DESCRIPTION 8

DEFINITION

CODE D'APPEL : 7E ENTITE UTILISATEUR : .PPTEX

DEFINITION _____

NOM DE L'OCCURRENCE: _____

PROGRAM CODE : 1_____
EXTRACTION TYPE : 2
OPTIONS : 3_____
MAX. RECORD SIZE : 4__
SORT OPTION : 5____

N L	CLASS VALEU	DESCRIPTION OF FIELDS AND FILLING MODE
		PROGRAM CODE Required. The value input in this field indicates: . Where the COBOL source of the Extraction Master Master Path will be stored before its compilation, . What the PROGRAM-ID clause contains. Furthermore if the Extraction Master Path is a user extraction program, this value is the external name of the executed program.
		EXTRACTION TYPE Required.
	M	Macro-command (sub-program)
	E	User extraction program (program)
5		OPTIONS
	STATIC	Default value: Static CALL of the PAF extractor
	DYNAM	Dynamic CALL of the PAF extractor
		MAX. RECORD SIZE (NUMERIC) Optional. maximum size of the extracted records formatted by presentation lines. This field only applies to a User Extraction ('E' extraction type).
		SORT OPTION Optional. There are two sort options regarding the Extractor's output results:
	CURS	Default value: Primary sort on the extracted occurrence (entity type and occurrence code) Secondary sort on the identification criterion. (extraction pathway which leads to an occurrence)
	IDENT	Primary sort on the identification criterion Secondary sort on the extracted occurrence (entity type and occurrence code) For a macro-command, the CURS option is always implemented.

EXTRACTION MASTER PATH: DEFINITION / DESCRIPTION **8**

DESCRIPTION 1

DESCRIPTION 7E _____ 1 .....

A	NLG	L	T	REF.	SELECTION-COMMENT	VIR.PRES	OPDV
		1	2	3	4	5	6

N L	CLASS VALEU	DESCRIPTION OF FIELDS AND FILLING MODE
		EMBEDDED LEVEL Optional. Its value must be numeric and included between 1 and 9. Value '1' must be unique in an Extraction Master Path because it corresponds to the entry point. Input in this field is relevant in S-, A-, and V-type lines.
	   S A I O P V * blanc	LINE TYPE Required. Sequencing Access Input conditions Output conditions Presentation Virtual cursor Comments (printed in the validation report) Ignored by the validation procedure For more information, refer to Chapter 'Extraction Master Path Definition/Description', in this manual.
		CODE OF REFERENCE CURSOR Optional. Required on S-, A-, and V-type lines. On S- and A-type lines, it specifies the cursor code associated with the PAF table. It allows for the extraction sequencing and identifies the occurrence extracted when the identification criterion is created. For more information, refer to Chapter 'The Extraction Master Path', Subchapters 'The Extraction Sequence (S-type lines) and (A-type lines)'. On V-type lines, it defines a virtual cursor code for the selective extraction of an occurrence. For more information, refer to Chapter 'The Extraction Master Path', Subchapter 'Selective Extraction (V-type lines)'.
4		SELECTION OR COMMENT Required on S-, A-, P-, I-, and O-type lines. For more information, refer to Chapter 'The Extraction Master Path'.
		VIRTUAL & PRESENTATION CURSOR

N L	CLASS VALEU	DESCRIPTION OF FIELDS AND FILLING MODE
		Optional. The first four characters are a virtual cursor and the last four are a presentation cursor. For more information, refer to Chapter 'The Extraction Master Path', Subchapters 'Selective Extraction (V-type line)' for the virtual cursor and 'Presentation (P-type line)' for the presentation cursor.
		PROCESSING OPTIONS Optional. 'O' column: Print option (for S-type lines)
	blank or	Storing of the occurrence in the extraction result file; this data can be edited.
	N	Storing of the occurrence in the temporary Work File; this data cannot be edited.
		'P' column: Definition of en entry point
	P	Definition of an occurrence as an entry point.
		'D' column: Delimiter value
	blank	Default value in the description syntax of the extraction on S- and A-type lines.
	,	Default value on P-type lines. If these default values are not convenient, the user may specify other values in this field. Warning: The "-" and "'" values are not authorized.
		'V' column: Value of the delimiter which bounds a constant on a P-type line (see Chapter 'The Extraction Master Path', Subchapter 'Presentation (P-type line).
	}	Default value If this value is not convenient, the user may modify it in this field. Warning: The "-" and "'" values are not authorized.

9. EXTRACTION MASTER PATH

9.1. EXTRACTION SEQUENCE (S-type lines)

EXTRACTION SEQUENCING

Data is extracted from the Database according to a path -- a sequencing -- based on the cross-references existing between entities in the metamodel.
This is why a thorough knowledge of the metamodel being used is essential.

The metamodel can be the VisualAge Pacbase classic metamodel or the Work Station metamodel for the Pacdesign or Pacbench Rews modules (which varies with the methodology in use).

An Extraction Master Path can be thought of as a tree whose branches sub-divide and explore the Specifications Dictionary in finer and finer detail.

The syntax used to describe the sequencing of an extraction is similar to the language used for on-line navigation within the Database.

In other words, you should ask yourself the following questions before anything else:

1. What data do I need to extract ?

2. What on-line input is necessary to access the screens which correspond to the extracted data?

Extraction sequencing is entered on S-type lines in the SELECTION-COMMENT field of the $7E.....D screen.

An extraction sequence is composed of three elements; the entity type, the occurrence, and the type of line.

S-type lines examples are given at the end of this subchapter and two complete PTEx examples are presented at the end of this manual.

EXTRACTION MASTER PATH 9
EXTRACTION SEQUENCE (S-type lines) 1

1. ENTITY TYPE:

The entity type to be entered corresponds to the entity type that is entered in the On-Line CHOICE field.

NOTE: If the entity is a WorkStation entity, its type must be coded according to the following format:

//ü__CCC

where 'ü' is the Methodology code ('M' for Merise, 'D' for YSM...),
and 'CCC' is the entity local code.
You can refer to sub-chapter 'Introduction to PAF Function', section 'Description of Tables' for the list of method codes.

2. OCCURRENCE:

The occurrence must be separated from the entity type (default value: SPACE, modifiable in column 'D'). It is identified by the following two elements separated by a dash (-):

1) Cursor code identifying the hierarchically higher PAF Table. The extraction sequence depends exclusively on the cursor code which must be unique in an Extraction Master Path.

2) Contents of the occurrence code: PAF Column code.

The PAF Column codes are given in a specific manual, Description of PAF Tables (Ref: DD PAG).

NOTE: If the occurrence is an occurrence of a WorkStation entity, the code of the Column code is the code of the Data Element called by the User Entity which supports this entity in the Database.

In this case the cursor code and the Data Element/Column code are separated by two dashes (--). See Example e) at the end of this subchapter.
The codes of these Columns are given in the manuals referenced in Paragraph 1.

EXTRACTION MASTER PATH 9
EXTRACTION SEQUENCE (S-type lines) 1

3. TYPE OF LINE - TABLE TO EXTRACT:

This type of line corresponds to the coding in the second part of the On-Line CHOICE field (ex: //M DOM DOM-COEU X1MCD, where X1MCD is the type of line). It completes the entity type to specify which table is to be extracted.
This coding must be separated from the occurrence identifier (default value: SPACE, modifiable in column 'D').

NOTE: Coding for WorkStation entities is given in the manuals referenced in Paragraph 1.

EXAMPLES

a)

```
  LIN : L T REF. SELECTION-COMMENT

        010 : 1 S PGM  P
        020 : 2 S LDST P PGM-CPGM CD
```

On line 010, the L column (which corresponds to the embedded level) does not need to be filled in since this line correponds to the initial entry.

On line 020, the list of Data Structures for each occurrence of the Program entity will be requested.

b)

```
  LIN : L T REF. SELECTION-COMMENT

        010 : 1 S SEG  S
        020 : 2 S LDEL S SEG-CSEG CE
```

The Data Elements (LDEL cursor) belonging to each occurrence of the Segment (SEG cursor) entity will be listed.

c)

```
  LIN : L T REF. SELECTION-COMMENT

        010 : 1 S PGM  P
        020 : 2 S PGMP P PGM-CPGM P
```

The '-P' lines of each occurrence of the Program entity will be listed.

d)

```
  LIN : L T REF. SELECTION-COMMENT

        010 : 1 S DOM  //M DOM
        020 : 2 S LMCD //M DOM-CUEO X1MCD
```

The Conceptual Data Models for each occurrence of the Domain entity will be listed.

```
e)
  LIN : L T REF. SELECTION-COMMENT

       010 : 1 S PHA  //M PHA
       020 : 2 S FLC  //M CHA PHA--PHACH
```

The definition of the Flowchart associated with each occurrence of the Phase entity will be listed.

Explanation of this coding:

On the Definition of a Phase appears the code of its associated Flowchart. The Data Element which contains the occurrence code of the associated Flowchart must then be extracted.

In the manual, DESCRIPTION OF THE USER ENTITIES DEDICATED TO THE WORKSTATION, the first chapter on the Flowchart entity contains the description of the Data Elements used in the Definition of the Flowchart. For the fourth field - FLOWCHART - the Data Element code (.PHACH) corresponding to the VALUE column will be read. It will be coded -PHACH in the Extraction Master Path, separated from the cursor code by the second dash (-).

More generally, all Data Elements which describe a User Entity AND whose code is prefixed by a period, must correspond to a Column code prefixed by a dash in place of the period.

NOTE: At the end of this manual, an appendix presents two examples of Extraction Master Paths. One uses the WorkStation metamodel, and the other one uses the classic ViasualAge Pacbase metamodel.

9.2. EXTRACTION SEQUENCE (PARTICULAR CASES)

THE EXTRACTION SEQUENCE (PARTICULAR CASES)

Ambiguities may appear when selecting certain tables for extraction, particularly cross-reference tables. In this case the occurrence identifier and the table must be added.

EXAMPLE: The objective is to list the uses of a Segment in Programs.

These uses may be viewed on the 'S....XP.....CP..' and the 'S....XP.....W.....' screens.
The first screen lists the Segments called by the Data Structures that are called in the Program (P.....CD). The second screen lists the Segments called in the Working Storage Section of the Program (P.....W).
This is why the following may be ambiguous:

```
LIN : L T REF. SELECTION-COMMENT
020 : 2 S SEGP S SEG-CSEG XP
```

Therefore, it should be written as:

```
          S SEG-CSEG XP PGM-CPGM CD
and/or    S SEG-CSEG XP PGM-CPGM W
```

At this step in the pathway, if the PGM cursor has not been defined, i.e. if no extraction has been requested on the Program entity, the following lines must be entered:

```
          S SEG-CSEG XP * CD
and/or    S SEG-CSEG XP * W
```

The result will be the uses of the Segment in the -CD and/or -W of all the Programs of the queried Library.

9.3. EXTRACTION SEQUENCE (A-type lines)

SEQUENCE FOR A-TYPE LINES

Like an S-type line, an A-type line expresses an extraction selection. To condition an extraction by testing the value of a Data Element (which belongs to the cursor, BUT which is not an identifier) the user must write an A-type line and enter the hierarchially superior cursor in the 'VIR.' field.

EXAMPLE:

The objective is to find and list the uses of the Data Elements in the '-CD' of Programs.

```
LIN : L T REF. SELECTION-COMMENT              VIR.PRES
010 : 1 S DEL  E
020 : 2 A PCDE P * CD                         DEL
030 :   O      CDEL = DEL-CDEL
040 :   O      OR CRES = DEL-CDEL
050 :   S PGM  P PCDE-CPGM
```

NOTE: In this example the identifiers of the PCDE cursor are P (for Program) and CD (for Data Structure); CDEL is an additional Data Element of the table.

First, extract all the Data Elements, then select the -CD lines of Programs calling the Data Elements (the Programs not having been called in a preceding extraction).

The pathway does not logically follow. The link between the Data Elements and the Programs is assured by an A-type line which contains the cursor of the Data Element Table in the first four characters of the VIR.PRES field. This type of line includes a selection expressed with an asterisk.

Once the objects of the extraction are identified, the extraction conditions (the "filter") are entered on one or several O-type lines which must refer to the cursor specified in the VIR.PRES field of the A-type line.

NOTE: The filter is discussed in the next Subchapter.

9.4. CONDITIONS AND FILTERS (I and O-type lines)

INTRODUCTION

The condition line authorizes the extraction from a Table (Input).
The filter selects the occurrences resulting from the extraction (Output).

CONDITIONS : I-TYPE LINES

Conditioning an extraction is particularly useful when the extraction must follow a search path starting from the data which fulfills the condition(s). Data occurrences that are excluded by the condition will not be extracted.

The condition is expressed on an I-type line, in COBOL, in the form of a Boolean operator + the expression. The condition references a hierarchically greater Table already extracted.

EXAMPLE: The objective is to list the occurrences used by a Program.

For the Report entity, occurrences are to be extracted only if the call of their Data Structures in the -CD of the Program is an I- or J-type line.

```
LIN : L T REF. SELECTION-COMMENT
010 : 1 S PGM  P
020 : 2 S LDST P PGM-CPGM CD
030 : 3 S LRPT D LDST-CDST LR
040 :   I      LDST-ODSTUS = 'I' OR 'J'
```

FILTERS: O-TYPE LINES

Filters make it possible to screen out unwanted occurrences of an extraction. A filter is expressed on an O-type line in SQL-PAF language using the WHERE clause of the EXEC PAF DECLARE command.

The syntax of this language is described in Chapter "Implementation in User Programs", Subchapter "Syntax of the SQL-PAF Language", Paragraph "Cursor Declaration".

The Filter uses a column of the current Table as a filter criterion.

NOTE: At least one filter line is mandatory after an

A-type selection line.

EXAMPLE: The objective is to list the texts called in a Volume whose occurrence code is RAP001 and to list the Data Elements linked to these texts.

```
LIN : L T REF. SELECTION-COMMENT
010 : 1 S TXT  T
020 : 2 S XVOL T TXT-CTXT XV
030 :   O      CVOL = 'VOL001'
040 : 3 S TXTD T XVOL-CTXT D
050 :   O      CDEL <> SPACE
```

9.5. SELECTIVE EXTRACTION (V-type line)

SELECTIVE EXTRACTION: V-TYPE LINES

The processing loops of an extraction may cause the same occurrence to be extracted several times. Such might be the case if a Data Element is used several times in a Program. The repeated extraction of these occurrences may or may not be relevant. If it is not, the principle of Selective Extraction is used.

A Selective Extraction associates each occurrence with an identification criterion which differs from the one associated with the extraction level in the PTEx.

EXAMPLE: The objective is to list the uses of the Data Elements in Programs. Each occurrence will be identified by the following criteria, according to the number of levels in the extraction:

```
1  Program         (PPPPPP) => PGM PPPPPP
2  Data Structures(DS)      => PGM PPPPPP LDST DS
3  Segment         (DSSS)   => PGM PPPPPP LDST DS LSEG DSSS
4  Data Element    (EEEEEE) => PGM PPPPPP LDST SD LSEG DSSS
                               LDEL EEEEEE
```

To link the Data Elements directly to the Program which uses them, a virtual or selection cursor code (DELP) must be defined. This cursor will have an embedded level of 2 since the path passes directly from the Program to the Data Elements:

```
=> PGM PPPPPP DELP EEEEEE
```

VARIANT: To link the Data Elements to their Data Structures, the selection cursor code is DELD. This cursor will have an embedded level of 3 (Program, Data Structures, Data Elements).

```
=> PGM PPPPPP LDST DS DELD EEEEEE
```

The selection cursor code is positioned in the Selection- Comment field (on S, I, or O-type line) of the entity to be selected, in the first four characters of the VIR.PRES field. Then, a V-type line is added at the end of the PTEx. On this line, the embedded level of the selection cursor is entered in the L column and the selection cursor code itself is entered in the REF. column.

```
EXAMPLE:
        LIN : L T REF. SELECTION-COMMENT            VIR.PRES
        010 : 1 S PGM  P
        020 : 2 S LDST P PGM-CPGM CD
        030 : 3 S LSEG D LDST-CDST LS
        040 :   I      (LDST-ODSTUS NOT = 'I' AND 'J'
        050 :   I      AND LDSC-ODSTOR = 'V' OR 'S')
        060 : 4 S LDEL S LSEG-CSEG CE               DELP
        070 :   O      CDEL <> FILLER
        900 : 2 V DELP


                VARIANT :

                LIN : L T REF. SELECTION-COMMENT            VIR.PRES
                ... : . . ...  ......................
                030 : 3 S LSEG D LDST-CDST LS
                ... : . . ..........................
                060 : 4 S LDEL S LSEG-CSEG CE               DELD
                070 : . . ..........................
                900 : 2 V DELD
```

A selection cursor must be entered as many times as branches of the extraction tree lead to the same entity starting from the same reference entity.

For example, different extraction paths may finally lead to the Data Elements used in a Program. In the preceding example, the DELP cursor should be used several times, each time referenced to the same V-type line.

REMARK: The SELECTION-COMMENT field of a V-type line can contain a title or a label for documentation purposes.

9.6. PRESENTATION (P-type line)

PRESENTATION: P-TYPE LINES

Using presentation lines (or format lines), the user may:

. Select columns which describe occurrences resulting from an extraction,

. Specify a particular presentation for these selections.

These lines may be used to describe User Extractors (E-type PTEx) and Macro-Commands (M-type PTEx).

P-type lines are not required. Whether P-type lines are entered or not, a PTEx always creates a 'raw' result file which contains all the Tables extracted (except those withheld by the user; see the 'Print Option', in the 'Processing Options' of the 'Extraction Master Path: Definition / Description' Chapter).

USES OF P-TYPE LINES

In User Extraction Programs, Presentation lines may be used to:

. 'Draw' personalized list layouts,
. Automatically format user input for batch procedures such as GPRT or EXTR (but this is useless for UPDP).

In Macro-Commands, Presentation lines allow the user to specify a format which will be taken into account in a PDM+ Volume. Such a format is called by a G-type line in a Master Outline.

>>> For more details, refer to the Personalized Documentation Manager Reference Manual.

SYNTAX OF P-TYPE LINES (SELECTION-COMMENT field):

P-type lines are written in positional language, based on the juxtaposition of the following parameters:

PPP,LLL,content

PPP Positioning:

Numeric, 3 characters maximum, indicates the position of the beginning of the transfer into the reception field.

LLL Length:

Numeric, 3 characters maximum, indicates the length of the field to transfer into the reception field.

content

Field to transfer:

. Constant (bounded by the value present in column 'V' of the OPDV field, '}' by default).
. Table column (column code) extracted by the current cursor (generally a column which contains an occurrence code).
. Table column (cursor-column) extracted by another cursor.
A column from another table may be called in 'content'; so columns coming from different tables may be formatted the same way.

'PPP,LLL,content' can be repeated as many times as necessary; knowing that the expanded length of the reception field must not exceed the value specified by the MAX. RECORD SIZE field in the PTEx Definition.

NOTE: A presentation may be written on several consecutive P-type lines provided that the presentation cursor is entered on the first line only. See below Paragraph 'Positioning Presentation Cursors' for more details.

Several presentations for an extracted occurrence may be defined, and one or several of them may be selected. For example, an E-type occurrence corresponds both to a Data Element and to a Property. The user may then define a specific presentation for a Data Element, and another presentation for a Property.

A P-type line may be conditionned if it is directly followed by an I-type line.

```
Ex:  P  PRE1  1,8,'DATA ELEMENT',10,6,CDEL
     I         DEL-TDEL = 'R'
     P  PRE2  1,9,'PROPERTY',11,6,CDEL
     I         DEL-TDEL = 'I'
```

If a presentation is associated with several cursors, the cursor code must be replaced by an asterisk '*'.

```
Ex:  P  PRE1  1,8,'DATA ELEMENT',10,6,CDEL
     I         *-TDEL = 'R'
```

The comma is the default value for delimiting the parameters of P-type lines. Another character can be used as a delimiter if it is specified in column 'D' of the OPDV field of each P-type line concerned.

SPECIFIC SYNTAX OF THE MACRO-COMMAND P-TYPE LINES

The SELECTION-COMMENT field in the description of a User Entity occurrence recognizes the '$VF' PDM presentation parameter which corresponds to the character used for vertical separators. Presentation is limited to 132 characters (to avoid truncations at the time of printing).

All or part of a cursor presentation may be taken into account in a PDM+ Volume. For more information refer to the description of a Master Outline in the 'Personalized Documentation Manager' Reference Manual.

EXTRACTION MASTER PATH 9
PRESENTATION (P-type line) 6

POSITIONING PRESENTATION CURSORS:

A presentation cursor is positioned inside the selection (S, A, I, or O-type lines) of the table to be formatted in the last four characters of the VIR.PRES field.

NOTE: If a V-type line exists for the cursor that is to be formatted, it is recommended that the cursor be positioned directly on this line (see line 900 of variant 2 shown below).

The P-type line is entered, either:

. directly in the selection of the Table to be formatted, in which case entering the presentation cursor in the REF. field is not necessary (see variant 2), or
. at the end of the Extraction Master Path as in variant 1.

The REF. field must contain the presentation cursor entered previously, except in the case of variant 2.

The user may enter as many P-type lines as there are presentations to be extracted. A presentation may also be described on several P-type lines (see the NOTE above).
Conversely, a specific presentation may be associated with the same cursor.

EXAMPLE:

```
LIN : L T REF. SELECTION-COMMENT            VIR.PRES
010 : 1 S PGM  P
020 : 2 S LDST P PGM-CPGM CD
030 : 3 S LSEG D LDST-CDST LS
040 :   I      (LDST-ODSTUS NOT = 'I' AND 'J'
050 :   I      AND LDST-ODSTOR = 'V' OR 'S')
060 : 4 S LDEL S LSEG-CSEG CE
070 :   O      CDEL <> FILLER
080 :   P      2,7,}W1EX UE},9,6,CDEL
```

VARIANT 1:

```
LIN : L T REF. SELECTION-COMMENT            VIR.PRES
... : . . .... ..............................
060 : 4 S LDEL S LSEG-CSEG CE               PDEL
070 :   O        CDEL <> FILLER
... : . . .... ..............................
999 :   P PDEL 2,7,}W1EX UE},9,6,CDEL
```

EXTRACTION MASTER PATH 9
PRESENTATION (P-type line) 6

VARIANT 2:

```
LIN : L T REF. SELECTION-COMMENT             VIR.PRES
... : . . ...  ..............................
060 : 4 S LDEL S LSEG-CSEG CE                DELP
... : . . ....................................
900 : 2 V DELP SELECTION & PRES. SUPPORT          PDEL
910 :   P      2,7,}W1EX UE},9,6,CDEL
```

Above, the extraction finds the Data Element lines which are then ready to be processed by EXTR, the batch extraction procedure.

10. EXECUTION OF A USER EXTRACTOR / E-Type PTEx

EXECUTION OF A USER EXTRACTOR: USER INPUT

. Several *-type lines, at least one is required.

POS.	LEN.	VALUE	MEANING
2	1	*	Line type
3	8	uuuuuuuu	User code
11	8	pppppppp	Password
19	3	bbb	Library code
22	4	ssss	Session number
			Default value: current session
26	1		Session status if frozen:
		' '	Frozen
		T	Test
27	1		Library Network Option:
		C	Default value:
			Selected and upper libraries
			See User's Reference Manual for
			other possible values

. A required 'X'-type line. One of its purposes is to qualify the extraction's scope.

Several X-type lines may be entered so as to parameterize different executions of one User Extractor program.

POS.	LEN.	VALUE	MEANING
2	1	X	Line type
3	4		Cursor code
			Default value: 1rst cursor
7	8		Occurrence code
			Default value: All occurrences
			NOTE: The '*' generic character
			is allowed.
15	8		Start limit
23	8		End limit
			---> User Input Chart-Con'd next p.

USER INPUT - Cont'd

POS.	LEN.	VALUE	MEANING
31	1	1	Printing of a debug report default value : no debug report
32	6		Number of records in SYSPAF file Equivalent to PAF/SIZE parameter Default value : 10

For technical information, please refer to the Operations Manual, Part I, Chapter "INSTALLATION", Subchapter "COMPLEMENT: EXECUTION OF A PAF+ USER EXTRACTOR".

11. EXAMPLES OF EXTRACTION MASTER PATHS & XPAF REPORTS

EXTRACTION MASTER PATH VALIDATION REPORT EXAMPLES

The following pages contain two examples of Extraction Master Path validation reports (XPAF Procedure).

An XPAF execution report is composed of:

1. The "COMMENT OR ERROR" page;
2. The user input, the list of the Master Outlines which call the Extraction Master Path (they will also have to be validated), the '-G' lines associated with the Extraction Master Path;
3. The EXTRACTION MASTER PATH DESCRIPTION LINES;
4. The extraction simulation.

The first Extraction Master Path example explores the Flowchart entity cross-references specific to the WorkStation's metamodel.

The second example obtains the same results but uses the Specifications Database metamodel.