

# Natural for VSAM - Operation

This section provides information on various operational aspects of Natural for VSAM:

- Invoking Natural for VSAM
  - OPEN/CLOSE Processing
  - Natural File Access
  - Buffers for Memory Management
  - Application Programming Interfaces
- 

## Invoking Natural for VSAM

If the Natural interface to VSAM is available, it is initialized when you start a Natural session. It can be switched off by setting the VSIZE parameter to 0 (see also the relevant description in the section Parameters).

## OPEN/CLOSE Processing

In this section, VSAM files means both VSAM user files and VSAM Natural system files.

Database OPEN/CLOSE processing is controlled by the Natural parameter OPRB, which is described in Profile Parameters in the Natural Parameter Reference documentation.

Instead of using the OPRB parameter, you can also use the NTOPRB macro of the Natural parameter module, which is described in Parameter Modules in the Natural Parameter Reference documentation.

An OPEN/CLOSE error must be followed by the NAT3539 error message. In a TP environment, the NAT3516 error message can also occur during an active Natural session if the file is closed.

### **Note:**

For dynamic OPEN handling within a session, you can use the application programming interface USR2008N.

The section below covers the following topic:

- OPRB Parameter for VSAM Databases

## OPRB Parameter for VSAM Databases

The OPRB parameter is not applicable under CICS or Com-plete, because in these environments, the TP monitor controls the OPEN/CLOSE processing of VSAM files.

By default, that is, without the OPRB parameter being specified, VSAM files are opened for input/output so that they can be read and/or updated.

If you want all used VSAM files to be opened for input only, you specify the OPRB parameter using the following syntax:

```
OPRB = (.ALL)
```

With this syntax, you specify an OPEN request for *all* VSAM files to be addressed. All files are opened for input only; individual files, however, are only opened when they are actually addressed by a given program.

**Note:**

If you want all VSAM system files to be opened for input, you have to set the Natural profile parameter ROSY=ON; see also the relevant section in the Natural Parameter Reference documentation.

If you want to open VSAM files for input (I) or output (O) per DBID, use the following syntax:

```
OPRB = (DBID = nnn, { MODE = { I
                             O
                             string ; ...
                           } [ , string ; ... ] } [ , ... ] )
```

With MODE, you specify a global default handling for DBID *nnn*.

If you do not want to specify a default handling per DBID or if, for some VSAM files, you want an input/output handling other than the default one, you specify the *string* parameter in the appropriate way.

The DBID must be defined with the NTDB macro as a VSAM DBID, and *string* varies depending on the operating system (see below).

**Important:** If several strings are to be defined, a semicolon (;) must be specified as delimiter character. If not, the semicolon must be omitted.

## Under z/OS

Under z/OS, you specify the *string* as follows:

```
{ FNR = nnn
  { DD = dd-name , TYP = { K
                        E
                        R
                        P
                        O
                        I
                        B
                        A
                        [ ,R ]
                      }
  { ,
  { ,
  [ ,R ]
```

The specified VSAM files must be defined as DDMs. However, instead of specifying the file number of the Natural DDM that corresponds to the VSAM file to be addressed, the *dd-name* and type ( **K**SDS, **E**SDS, **R**RDS, or **P**ATH) of this file can be specified directly, which saves you from having to look into the DDM first.



## Natural File Access

The Natural interface to VSAM supports VSAM entry-sequenced datasets (ESDS), key-sequenced datasets (KSDS), relative record datasets (RRDS), variable relative record datasets (VRDS), and paths for alternate indexes.

To enable Natural to access VSAM files, a Natural DDM is required for each VSAM file that is to be made accessible to Natural programs.

The section below covers the following topics:

- Natural Data Definition Modules - DDMs
- SYSDDM Main Menu
- Catalog DDM
- Edit DDM
- Restrictions with DDM Generation as Compared to Adabas

### Natural Data Definition Modules - DDMs

A DDM (data definition module) must be set up for each file. DDMs are created and maintained with Predict (see the Predict documentation for details) or with the Natural utility SYSDDM; they are stored in the Natural dictionary system file (FDIC).

With VSAM, in addition to logical Natural DDMs, also VSAM user DDMs can be created from one physical DDM.

If you do not have Predict installed, use the SYSDDM utility to generate DDMs from VSAM files. The SYSDDM utility is described in the Natural Editors documentation; the parts of it relevant to VSAM are described in the following sections.

All DDMs used within a session are located in the Natural buffer pool. This increases performance and enables synchronization of DDM usage across multiple sessions.

### SYSDDM Main Menu

The following functions on the main menu of the SYSDDM utility are relevant to Natural for VSAM:

Function	Explanation
Catalog DDM	<p>The DDM currently in the work area is cataloged, making it available for use within Natural applications. The DDM must have previously been placed in the work area by a READ command, or have been entered by using the Edit DDM function described below.</p> <p>Below are further details about Catalog DDM.</p>
Edit DDM	<p>Reads a DDM from the system file FDIC and into the SYSDDM work area, where it can be edited.</p>
List DDMs	<p>Displays a single DDM source (DDM editor <i>not</i> invoked) or a list of DDMs. The display format and options are identical to those of the LIST DDM command.</p>
Copy DDM to Another FDIC File	<p>One or all DDMs can be copied to a different Natural system file (FDIC) and/or to a different database. This is, for example, necessary during conversion of a Natural application from test to production status.</p> <p>In addition to the DDM name, DBID and FNR, with Natural for VSAM the file type <b>V</b> must be specified, as well as the DD/FCT name of the Natural system file FDIC, if the FDIC file is a VSAM file.</p>
List DDMs with Additional Information	<p>Displays a list of the DDMs stored in the specified FDIC system file. From the list, you can select individual DDMs for further processing. This function differs from the List DDMs function in that it displays additional items of information on the individual DDMs.</p> <p>The information displayed includes file name, DBID, file number, DDM length, security type (with Natural Security only), file type (that is, LOG.DDM, PHY.FILE, LOG.FILE or USERDDM for VSAM DDMs) and remarks as, for example, the VSAM file organization (KSDS, VRDS, RRDS, ESDS); see the section SYSDDM Utility in the Natural Editors documentation for details.</p>
Delete DDM	<p>Deletes a previously cataloged DDM from the Natural system file FDIC. The DDM remains in the work area.</p> <p><b>Important:</b> If a DDM is deleted with SYSDDM, the corresponding Natural Security file profile is automatically deleted.</p>

The following parameters relevant to Natural for VSAM can be specified for the various functions:

Parameter	Explanation
DDM Name	The name of the DDM to be processed.
FNR	The file number of the DDM to be processed.
DBID	The database which contains the DDM to be processed.
Replace	If <b>Y</b> is entered, DDMs which are being copied or cataloged and which are already existent are replaced. If <b>N</b> is entered, such DDMs are not replaced.
FDIC Type	The type of the system file FDIC.
DDM Type	The type of the DDM. For VSAM, the type must be <b>V</b> .
DBID Type	The type of the DDM. For VSAM, the type is <b>V</b> .

## Catalog DDM

A DDM can be cataloged by either using Function Code **C** in the SYSDDM main menu or entering the CATALOG command in the Command line of the DDM maintenance editor.

File name and file number are required for this function. With Natural for VSAM, a DBID assigned to VSAM must be specified. If no DBID is entered, it is assumed to be 0 and is generated dynamically at execution time based on the DBID of the Natural user system file (FUSER) in use (see also the description of the UDB parameter in the section Profile Parameters in the Natural Parameter Reference documentation).

If a DBID assigned to VSAM is specified (and **V** for VSAM in the field "Type of this DDM"), SYSDDM prompts you for additional information.

### Note:

The actual DBID assignments for VSAM is made with NTDB macros when assembling NATPARM; see Installation Procedure for z/OS and z/VSE for details.

## Additional Options for VSAM Files

If the DDM is to access a VSAM file, an additional screen, requiring the entry of additional VSAM options, is displayed:

```
11:24:04          ***** NATURAL SYSDDM UTILITY *****          2006-05-25
                   - Catalog a VSAM file/DDM -

      DBID 254   FNR  12   DDM AUTOMOBILES-VS          Def seq
-----
VSAM file information

VSAM file name ..... AUTO
VSAM View .....(Y/N) N
Logical related to FNR ....
User defined prefix .....

VSAM file organization

KSDS, ESDS, RRDS, VRDS (K,E,R,V) .. K

Compress file .....(Y/N) N
Zones X'0C' / X'0F' (C/F) F
```

The additional options for VSAM files consist of two parts: VSAM File Information and VSAM File Organization.

### VSAM File Information Options

Option	Explanation
VSAM File Name	<p>The DDNAME/FCT entry as defined to the TP monitor or when using batch mode, for example:</p> <pre data-bbox="427 310 639 338">//PERSON DD . . .</pre> <p>where PERSON would be entered under VSAM File Name.</p>
VSAM DDM	<p>Indicates whether this DDM represents a logical user DDM or a physical DDM.</p> <p><b>Y</b> Indicates that the DDM represents a logical DDM, which means that it does not necessarily correspond to the physical layout of the VSAM file. A logical DDM must use the same file number as the physical DDM from which it is derived, and the corresponding physical DDM must exist at the time the user DDM is invoked during execution. The short names of the logical DDM must be identical to those defined in the physical DDM. The sequence of fields within the DDM can be different from the physical sequence. The primary-key field must not be deleted from the DDM.</p> <p>Since the logical DDM is a subset of the physical DDM, the corresponding subsets of the underlying VSAM file appear to the user as independent files with different record layouts. When processing a logical DDM, the user obtains records only from the corresponding subset and not from any other subset contained in the same physical VSAM file.</p> <p><b>N</b> Indicates that the DDM represents a physical DDM. Only one DDM with a given file number can be used as the physical DDM for a VSAM file. This physical DDM is used internally by Natural to calculate field offsets.</p>

Logical DDMs are used to define different record types in a physical VSAM file. At DDM generation, these record types are identified by specifying a prefix for the primary key.

If a logical DDM is read, only records whose key begins with the specified prefix are returned from the VSAM file. Records beginning with any other prefix are ignored. If not specified otherwise, the prefix corresponds to the logical file number.

A different prefix must be assigned to each logical DDM. Natural automatically links the prefix with the logical key. The field layout in the logical DDM need not be the same as in the physical DDM.

The following two options are used only if the DDM represents a logical file which is to be derived from a physical VSAM file.

Option	Explanation
Logical Related to FNR	<p>This option is used to enter the file number of the physical DDM from which the logical file or DDM is derived.</p> <p>A logical DDM corresponds to a record type which is controlled by a prefix. Several logical record types can be contained in a physical VSAM file. The record types are distinguished by a prefix which determines which records are to be processed. See the example below.</p>
User-Defined Prefix	<p>The prefix value which is to be assigned for the logical file.</p> <p>The default prefix value is the logical file number (length 3).</p>

### Example of Logical Related to FNR

Physical Dataset		
Key		
{	X1234 X2345 X3456	} DDM1    PREFIX = X    FNR = 10
{	Z1234 Z1209 Z9000	} DDM2    PREFIX = Z    FNR = 11
Read DDM1 with key		
Display key		
results in:		
	Key	
	1234	
	2345	
	3456	

### VSAM File Organization Options

Option	Explanation
KSDS ESDS RRDS VRDS	<p>The type of VSAM file:</p> <p>K            KSDS file (default)</p> <p>E            ESDS file</p> <p>R            RRDS file</p> <p>V            VRDS file</p>
Compress File	<p>Specifies whether the file is to be compressed or not. The default is .</p> <p>N    Indicates that the file is not to be compressed. The file is written in the maximum length (that is, the length of all fields within this file) as defined in SYSDDM or Predict.</p> <p>Y    Indicates that the file is to be written in variable record length. During compression, the record is scanned backwards for default values, which are blank for alphanumeric fields, low values for binary fields, low values with a zone for packed fields and X'F0' for numeric fields. Compression stops as soon as the first non-default value is detected or the first descriptor is found. The new computed length is used to write the record to the file; this applies to KSDS and ESDS files only.</p> <p>Compression of trailing null values in VSAM records minimizes the space required for VSAM records. The application programming interface USR0100N in the library SYSEXT is provided to be able to maintain the logical record length by a Natural program.</p>
ZONES X'0C' / X'0F'	<p>In Adabas all positive packed values have X'0F' as a zone. This value could be different in VSAM.</p> <p>F    Indicates that all packed data are written to the VSAM file with the zone X'0F'. This is the default.</p> <p>C    Indicates that all packed values are written to the VSAM file with the zone X'0C'.</p>

## Edit DDM

To edit the DDM currently loaded in the work area, you can use the DDM editor of the SYSDDM utility. If no DDM has been read into the work area, an empty screen is displayed, allowing the manual entry of a DDM definition.

Instead of entering a complete DDM manually, you can read an existing DDM definition into the work area, by entering "EDIT *DDMname*" in the DDM editor Command line. This DDM can be modified and cataloged under a different name.

**Note:**

When you modify a DDM, all objects which reference this DDM have to be cataloged again.

**DDM Editor**

```

11:26:09          ***** EDIT DDM (VSAM) *****          2007-02-25
DDM Name EMPLOYEEES-VS          Def.Seq.          DBID 254 FNR 1
Command
I T L DB Name          F Leng S D Remark
----- top -----
 1 AA PERSONNEL-ID          A 8.0 P
 * CNNNNNNN
 * C=COUNTRY
G 1 AB FULL-NAME
 2 AC FIRST-NAME          A 20.0 N
 2 AD MIDDLE-NAME          A 20.0 N
 2 AE NAME          A 20.0 A
 1 AF MAR-STAT          A 1.0 F
 * M=MARRIED
 * S=SINGLE
 * D=DIVORCED
 * W=WIDOWED
 1 AG SEX          A 1.0 F
 1 AH BIRTH          N 6.0
G 1 A1 FULL-ADDRESS
M 2 AI ADDRESS-LINE          A 20.0 N
 2 AJ CITY          A 20.0 N
    
```

If you enter the HELP command or a question mark in the Command line, the editor help information is displayed.

The header information of the DDM editor is:

DDM name	The name used to reference the DDM in a Natural program. The name must be unique within the specified Natural system file.
Def. Seq.	The default sequence by which the file is read when it is accessed with a READ LOGICAL statement in a Natural program.
DBID	<p>The database in which the file to be accessed with the DDM is contained.</p> <p>With Natural for VSAM, a DBID assigned to VSAM must be specified. If <b>0</b> is specified, the default DBID for the Natural user system file (FUSER) as defined in the Natural parameter module is used.</p> <p><b>Note:</b> The actual DBID assignments for VSAM are made with NTDB macros when assembling NATPARAM; see Installation Procedure for z/OS and z/VSE in the section Installation for details.</p>
FNR	<p>The number of the file being referenced.</p> <p>The specified file number is used internally by Natural for VSAM.</p>

The DDM itself comprises the following field definition attributes which can be entered or modified:

Attribute	Explanation
I	Line indicator. This field is used by the DDM editor to mark lines.  E Lines containing an error detected during execution of the CHECK command. S Lines containing a scanned value. X/Y Lines selected for copy/move operation.
T	Field Type:  G Group header. M Multiple-value field. P Periodic group header. * Comment line. <i>blank</i> Elementary field.
L	Level number assigned to the field. Valid level numbers are 1 - 7. Level numbers have to be specified in consecutive ascending order.
DB	For VSAM files, the two-character code which is used in VSAM.
Name	A 3- to 32-character external field name. This is the field name used in Natural programs to reference the field.
F	Field format. For valid formats, refer to User-Defined Variables, Format and Length of User-Defined Variables (in the Natural Programming Guide).
Leng	Standard field length. This length can be overridden in a Natural program. For numeric fields (format N), the length is specified as <i>nn.m</i> , where <i>nn</i> represents the number of digits before the decimal point and <i>m</i> represents the number of digits after the decimal point.
S	This attribute is not applicable to Natural for VSAM.
D	Descriptor Option.  A Indicates that the field is an alternate index for a VSAM file. P Indicates that the field is a primary key. S Indicates that the field is a primary subdescriptor or superdescriptor; that is, a primary key for a VSAM file. X Indicates that the field is an alternate subdescriptor or superdescriptor; that is, an alternate index for a VSAM file.
Remark	A comment which applies to a field and/or the DDM.

Most of the editor and line commands available with the Natural program editor also apply to the Natural DDM editor. Special commands, such as PROFILE, RENUMBER, SET, SHIFT etc. and some line commands are not available. Refer to the section SYSDDM Utility of the Natural Editors documentation and to the section Program Editor in the Natural Editors documentation for more details on editor commands.

### Extended Editing at Field Level

The DDM editor can also be used to enter or modify DDM definitions at field level.

Extended editing mode is used to specify field headers and edit masks to be applied when the field is used in a DISPLAY or INPUT statement, as well as further specifications for VSAM DDM definitions. All the other information specific to the field (field type, length, name, format, remarks) can also be modified at this point.

The extended editing mode is invoked by entering the line command **.E** in the first positions of the line containing the field.

A range of field definitions can be selected for editing by entering **.Ennn** where *nnn* is the number of fields to be selected.

The field level editing mode is terminated when you press ENTER with or without having made any modifications.

The Extended Field Editing screen displays special attributes of the field definition if the edited DDM is a VSAM DDM:

```

11:25:26          ***** EDIT DDM (VSAM) *****          2007-02-25
                    - Extended Field Editing -
DDM Name AUTOMOBILES-VS          Def.Seq.          DBID   254 FNR   12
I T L DB Name          F Leng  S D Remark
----- top -----
  1 GA OWNER-PERSONNEL-NUMBER          N 8.0    A SECONDARY KEY
-----
Field Header ..... OWNER/NUMBER_____
Field Edit Mask ..... _____
Alternate Index Name .. AUTOY____
Maximum Occurrence .... 1
Upgrade Flag ..... _ (X)
Unique Key Flag ..... _ (X)
Null Flag ..... _ (X)
Field GA redefines field __ with offset 0
    
```

The following attributes can be specified:

Attribute	Explanation
Alternate Index Name	If the field references a VSAM alternate index or a path (denoted by an <b>A</b> in column <b>D</b> ), the index or path name must be entered here.
Maximum Occurrence	The number of occurrences for a multiple-value field or a periodic group (denoted by an <b>M</b> or <b>P</b> in column <b>T</b> ).
<b>The following flags only apply to alternate indexes and not to paths.</b>	
Upgrade Flag	<p>Since Natural does not use VSAM paths, upgrading can be performed either by Natural or by VSAM when using a KSDS or ESDS file with alternate indices defined.</p> <p>A blank value indicates that upgrading the alternate index is to be done by VSAM, which is the default. If VSAM is to perform the upgrading, define the VSAM file using IDCAMS with UPGRADE.</p> <p>If you enter an <b>X</b>, upgrading of the alternate index is performed by Natural. If so, the AIX must be defined with the NONUPGRADE option.</p> <p><b>Note:</b> For LSR handling, it is recommended that you specify this option. Under CICS, the FCT entry must also contain the VARIABLE option.</p>
Sort Flag	<p>If this option is marked with an <b>X</b>, the alternate index is to be read in ascending or descending value order.</p> <p>This option only takes effect if the Upgrade Flag option is specified, too.</p>
Unique Key Flag	<p>If this option is marked with an <b>X</b>, Natural ensures that the values of the alternate index field are unique. An attempt to update with a non-unique value results in an error message. The default value is a blank.</p> <p>This option only takes effect if the Upgrade Flag option is also specified.</p>
Null Flag	<p>A value of <b>S</b> indicates that null values for the alternate index field are suppressed. The default value is a blank.</p> <p>This option only takes effect if the Upgrade Flag option is also specified.</p>

**Note:**

For all DDMs cataloged with Natural which contain alternate indexes and any specifications for the above flags, all flags are nullified during runtime as soon as path processing is activated for these DDMs.

The last two fields on the screen are used to define sub-/superdescriptors for a VSAM file. For example, to define the field S1 as superdescriptor beginning in field BA and ending in field BB, the following would be entered:

```
S1 redefines BA with offset 0
```

The field S1 must have been defined to VSAM as a primary or secondary key.

VSAM superdescriptors can only be constructed from fields which are contiguous. To define the field S2 as a superdescriptor which begins in the 11th position of field BA and ends with the first two positions of field BB, the following would be entered:

S2 redefines BA with offset 10

In addition, the length of S2 would have to be set to 7. As mentioned above, S2 must have been defined as a primary or alternate index to VSAM.

## Restrictions with DDM Generation as Compared to Adabas

- No keys can be defined within periodic groups.
- Descriptors that contain multiple-value fields are not allowed with VSAM.
- Natural DDMs for VSAM cannot contain multiple-value fields or periodic groups **within** periodic groups.
- The same field cannot be defined more than once in the same DDM. A data definition as in the following example would lead to unpredictable results when used with VSAM:

### Example:

```

...
G 01 AB FULL-NAME
      02 AC FIRST-NAMEA 20.0 N
      02 AD MIDDLE-I           A 1.0 N /* duplicate short name
      02 AE NAME               A 20.0
      01 AD MIDDLE-NAME       A 20.0 N /* duplicate short name
...

```

Natural would treat the field MIDDLE-I not as a redefinition of MIDDLE-NAME but as a separate field.

## Buffers for Memory Management

The VSIZE parameter is suballocated into ten different areas whose sizes are determined by the assembly of NVSPARM. The different VSAM areas are split into fixed and variable buffer types. If there is insufficient space in the VSIZE buffer for all NVSPARM areas, you receive error message NAT3592 during initialization. At runtime, error message NAT3513 can occur for fixed buffer types. In this case, you must adapt the corresponding NVSPARM value. Variable buffers are increased during runtime, NAT3513 does not occur. Some buffer sizes depend on the use of VSAM system files. The relevant buffers are FCT, FWA, TSA and UPD.

The VSIZE parameter is suballocated as follows:

- FCT - File Control Table
- FWA - File Work Area
- OPV - Open Table
- SFT - System File Table
- SWT - Switch Table
- TAF - Table of Accessed Files

- ROLL - Table of Session Status Information
- DFB - Table of Decoded Format Buffers
- TSA - Table of Sequential Access
- UPD - Table of Update Records
- VCA - Natural Control Area for VSAM

## FCT - File Control Table

FCT contains file-specific information and is a fixed buffer type.

FCT also contains the complete VSAM access control block (ACB), information on existing user exits, and information on the application programming interface USR0100N.

The size of the table is determined by using the following formula and then rounding up to a double-word boundary:

$$(72 + ACB-length) (TAFE * 2) + 80$$

Without VSAM system files, the default setting is:

$$(72 + 76) (10 * 2) + 80 = 3040$$

With VSAM system files, the default setting is:

$$(72 + 76) (26 * 2) + 80 = 7776$$

FCT and SWT (see below) share a common buffer area.

## FWA - File Work Area

FWA contains information on a VSAM request and is a fixed buffer type.

FWA also contains information on the VSAM request parameter list (RPL).

The size of the table is determined by using the following formula and then rounding up to a double-word boundary.

$$(40 + RPL-length) (TAFE * 2) + 80$$

Without VSAM system files the default setting is:

$$(40 + 76) (10 * 2) + 80 = 2400$$

With VSAM system files the default setting is:

$$(40 + (76*4)) (26 * 2) + 80 = 17968$$

FWA and OPV (see below) share a common buffer area.

## OPV - Open Table

OPV contains information on an OPRB string and is a fixed buffer type.

The size of the table is determined by using the following formula and then rounding up to a double-word boundary:

$$24 * (TAFE * 2) + 48$$

The default setting is :

$$24 * (10 * 2) + 48 = 528$$

OPV and FWA (see above) share a common buffer area.

## SFT - System File Table

This table is only active if VSAM system files are defined. The buffer type is fixed.

This area contains the description of the VSAM system files FNAT, FUSER, FDIC, FSEC and FSPOOL as well as the system file used by Natural ISPF, if available.

The size of the area is 8192 for SFILE=ON. The default setting is 0.

## SWT - Switch Table

SWT contains information necessary for the application programming interface USR1047N for dynamic DD/DLBL modification. SWT is allocated only if the value specified for the parameter DDSWITE in NVSPARM is greater than 0.

The SWT buffer type is fixed.

The size of the table is determined by using the following formula and then rounding up to a double-word boundary:

$$24 * DDSWITE + 48$$

The default setting is 0.

SWT and FCT (see above) share a common buffer area.

## TAF - Table of Accessed Files

This area describes the record layout for each file accessed by Natural; it is created by reading the physical or logical DDM for the file. Each TAF entry consists of a header entry and an entry for each field in the DDM. The header entry describes the type of file, file name, primary key, etc. The field entries describe the format, offset, and length of every field in the file. The layouts for the header and field entries are described in the macros NVMTAF and NVMFLD respectively.

The TAF buffer type is fixed.

The size of the table is determined by using the following formula and then rounding up to a double-word boundary:

$$(((32 * TAFN) + 112) * TAFE) + 80$$

The default setting is:

$$(((32 * 50) + 112) * 10) + 80 = 17200$$

## ROLL - Table of Session Status Information

This table is used to keep track of the position within a file for every active FIND or READ statement; it is identified by the CID. This allows Natural to release all VSAM resources during a ROLLOUT operation and then reposition itself correctly after a ROLLIN operation.

The ROLL buffer type is fixed.

The size of this area is determined by the parameter ROLLSIZ in the NVSPARM module, rounded up to a double-word boundary:

$$TAXSIZE + 80$$

The default setting is:

$$550 + 80 = 632$$

## DFB - Table of Decoded Format Buffers

The table is suballocated into two areas, one for global format IDs (GFIDs) and one for command IDs (CIDs).

For any given I/O request, this area describes which fields from the VSAM record area are returned to the Natural record buffer. Each DFB (decoded format buffer) entry consists of one header, identified by the CID or the GFID of the I/O request, plus an entry for each field to be returned to Natural. Each field entry in the DFB contains the format, offset, and length of the field as derived from the associated TAF entry for the file. The layouts of the header and field entries are described in the macros NVMDFB and NVMDFF respectively.

The DFB buffer type is fixed. If the no-space-condition occurs for GFID-oriented entries, the oldest entries are deleted.

The size of the TDFB area is determined by using the following formula and then rounding up to a double-word boundary:

$$(16 * DFBN * 2 + 36) * DFBE * 2 + 128$$

The default setting is:

$$(16 * 50 * 2 + 36) * 10 * 2 + 128 = 32848$$

## TSA - Table of Sequential Access

The TSA is used to keep important pointers and information on each READ/FIND statement. There is one TSA entry for each active READ and FIND statement, and each entry is identified by the CID. The layout of the TSA is described in the macro NVMTSA.

The TSA buffer type is variable.

The size of the area is determined by using the following formula and then rounding up to a double-word boundary:

$$(104 + \text{KEYLGH}) * \text{TSAE} + 80$$

The default setting is:

$$(104 + 32) * 10 + 80 = 1440$$

## UPD - Table of Update Records

This area contains an entry for every READ or FIND loop that contains an UPDATE or DELETE statement. These entries are released when an END TRANSACTION or BACKOUT TRANSACTION statement is executed. Each entry contains control information about the record and the values of all the fields that might be updated within the loop. The layout of each UPD entry is described in the macro NVMUPD.

The UPD buffer type is variable.

The size of the UPD area is determined by the parameter UPDL in the NVSPARM module, rounded up to a double-word boundary.

The default setting is 8272 without VSAM system files and 32848 with VSAM system files.

## VCA - Natural Control Area for VSAM

VCA is a fixed length area which contains pointers, addresses, flags, and work areas that are important to a Natural environment for VSAM. The layout for this area is described in the macro NVMCA. Within a Natural environment for VSAM, R3 always points to this area.

The size of this area is 6600 bytes.

## Application Programming Interfaces

Natural for VSAM provides the following application programming interfaces (APIs) in the Natural system library SYSEXT:

API	Function
USR0100N	Controls the VSAM variable record length (LRECL).
USR1047N	Supports dynamic switching of DD/DLBL names defined in a DDM.
USR2008N	Supports dynamic OPEN calls for VSAM datasets.

A short description of the APIs is provided in the following section; for more detailed information, log on to the library SYSEXT and display the API(s) desired.

The section below contains information on the following APIs:

- USR0100N
- USR1047N
- USR2008N

## USR0100N

The API USR0100N controls the record length of variable VSAM files.

The API is invoked as follows (a sample program called USR0100P is provided in the library SYSEXT):

```
CALLLNAT 'USR0100N' parm1 parm2 parm3 parm4 parm5
```

The parameters are described in the following table:

Parameter	Format/Length	Explanation
<i>parm1</i>	A1	Specifies either of the following function codes:  G For retrieval statements; the current record length is determined for <i>parm5</i> .  P For update/store statements; the length specified in <i>parm5</i> becomes the current record length.
<i>parm2</i>	A8	Specifies the DD/DLBL name for the current file (optional); if specified, <i>parm5</i> is only valid for this file.
<i>parm3</i>	N5	Specifies the DBID taken from the DDM (optional); is used instead of the DD/DLBL name and only in conjunction with <i>parm4</i> .
<i>parm4</i>	N5	Specifies the FNR taken from the DDM (optional).
<i>parm5</i>	N5	Specifies or returns the record length depending on the setting of <i>parm1</i> .

**Note:**

If neither *parm2* nor *parm3* and *parm4* are specified, *parm5* is valid for all files.

## USR1047N

The API USR1047N enables dynamic modification of DD/DLBL names within a Natural program if the DDSWITE parameter is specified in NVSPARM. It can be used if data are spread over several VSAM files which have different DD/DLBL names, but the same record structure.

The API is invoked as follows (a sample program called USR1047P is provided in the library SYSEXT):

```
CALLNAT 'USR1047N' parm1 parm2 parm3 parm4
```

The various parameters are described in the following table:

Parameter	Format/Length	Explanation
<i>parm1</i>	A1	Specifies either of the following function codes:  S For switching of DD names with the next following database calls. R For resetting of DD names; the switch table entry of function S has been deleted (see SWT - Switch Table).
<i>parm2</i>	A8	Specifies the old DD name taken from the DDM.
<i>parm3</i>	A8	Specifies the new DD name for the next database calls.
<i>parm4</i>	P4	Return code of Natural for VSAM.

The parameter *parm4* can contain the following response codes:

Code	Explanation
0	Normal return.
4	The switch table (SWT) is too small; increase the DDSWITE parameter.
8	The switch table entry has not been found; program error.
12	Invalid function code.
16	The switch table is not allocated; that is, the DDSWITE parameter is set to 0.

## USR2008N

This API is not applicable under Com-plete and CICS.

USR2008N supports dynamic OPEN calls during a Natural session if OPSUPP=ON is specified in NVSPARM (see also OPSUPP in the section Parameters).

The API is invoked as follows (a sample program called USR2008P is provided in the library SYSEXT):

```
CALLNAT 'USR2008N' parm1 parm2 parm3 parm4 parm5 parm6
```

The parameters are described in the following table:

<b>Parameter</b>	<b>Format/Length</b>	<b>Explanation</b>
<i>parm1</i>	N5	Specifies the DBID taken from the NTDB macro definition; see NTDB Macro in the section Parameters.
<i>parm2</i>	A1	Specifies the global OPEN MODE; see OPEN/CLOSE Processing.
<i>parm3</i>	A4	Specifies the data management type, for example, VSAM.
<i>parm4</i>	A40/16	Specifies the valid OPRB syntax and/or DDM long name instead of the DD= or FNR= definitions.
<i>parm5</i>	P4	Returns the Natural for VSAM error number.
<i>parm6</i>	A50	Returns the Natural for VSAM error text.