Search Buffers

The search and value buffers are used together to define:

- the search criteria used to select a set of records using a FIND command (S1, S2, S4); and

- the range of values to be traversed by logical sequential read commands (L3/6, L9).

If a search buffer is provided, a value buffer is also expected. If it is not provided, Adabas will create a dummy value buffer to pair with the search buffer. For complete information about the relationships between the different types of ABD or buffer specifications, read *Understanding the Different Buffer Types*.

Only one search and value buffer pair should be specified in a single Adabas direct call.

The user provides the search expressions in the search buffer and the values which correspond to the search expressions in the value buffer.

The syntax used for the search buffer depends on the type of search criteria to be employed:

1. Single file search. The search criteria consists of one or more fields contained in a single file;

2. Multiple file search using physically coupled files. The search criteria consists of fields contained in two or more files which have been physically coupled using the ADAINV utility;

3. Search using the soft coupling feature. This feature provides for a combination of search, read, and internal list matching.

A search criteria may also contain one or more fields which are not defined as descriptors. If nondescriptors are used, Adabas performs a read operation to determine which records are to be returned to the user.

Notes:

1. On files containing a large number of records, performing a search using nondescriptors can result in excessive response times.
2. The behavior of nondescriptor searches in Adabas databases differs between mainframe and open systems in regards to null suppression in the fields. In open systems, nondescriptor searches do not return records with null values in a field if the field is null-suppressed (NU); on mainframe systems, the null-suppression (NU) of fields is ignored during nondescriptor searches. At this time, to resolve this problem, we recommend that you remove the null suppression option (NU) for open systems fields, if the fields must be used for a nondescriptor search.

- Search Buffer Syntax

Search Buffer Syntax

This section outlines the syntax statement options for the search buffer. Delimiters (commas, slashes, parentheses, semicolons) must separate all search buffer entries as indicated. One or more spaces may be present between entries. The search statement must end with a period.

This section covers the following topics:

- Search Expression

- Connecting Search Expressions

- Searching One File

- Searching Multiple, Physically Coupled Files

- Searching One or More Files Using Soft Coupling

- Physically Coupled Files

- Soft Coupling

Search Expression

The search expression syntax is common to all types of searches:

Each of the elements in this syntax is now described:

field-name

The search expression can name a field (descriptor or nondescriptor), subdescriptor, superdescriptor, hyperdescriptor, collation descriptor, or phonetic descriptor. When using nondescriptors, multiple-value fields are permitted, but sub-/superfields are not.

If a nondescriptor is used, Adabas reads the entire file in order to determine which records satisfy the search criteria. If only descriptors are used, the inverted lists are used and no reading of records is necessary. Search criteria containing nondescriptors and descriptors may be combined.

If a descriptor field is not initialized and logically falls past the end of the physical record, the inverted list entry for that record is not generated for performance reasons and therefore, the record will not be returned in a search. To generate the inverted list entry in this case, it is necessary to unload short, decompress, and reload the file; or use an application program to initialize the field for each record of the file.

If the descriptor is defined with the NU option (null-value suppression), null values are not stored in the inverted lists; therefore, a search for all the records which have the null value will always result in no records found (even if there are records in Data Storage which contain a null value for the descriptor). This rule also applies to subdescriptors. A superdescriptor value is not stored if any field from which it is derived is defined with the NU option *and* the value of that field is actually null.

Note:
Large object (LB) fields cannot be specified in a search buffer, nor can they be specified in format selection criteria.

command-id

The command ID value is enclosed in parentheses and identifies a list of ISNs resulting from a previous S*x* command that specified the save-ISN-list option.

i (Occurrence Index)

The occurrence index (*i*) identifies a particular occurrence of a descriptor or nondescriptor within a periodic group and is used to limit the search to only the values located in the specified occurrence. If no index is provided, the values in all occurrences are searched.

- The index comprises one to five digits; leading zeros are permitted.

- An index is *not* permitted for a descriptor that is a multiple-value field, or a subdescriptor or superdescriptor derived from a multiple-value field. However, if the multiple value field is within a periodic group, the index is allowed but identifies the occurrence within the PE group and not within the multiple-value field.

D Daylight Savings Time Indicator

For fields with the time zone option TZ, a selection for daylight savings time can also be made using a D indicator element similar to that described in the section *Daylight Savings Indicator (D)*.

In search buffers, the following rules apply:

- A daylight savings indicator cannot be specified without a corresponding date-time field. The date-time field must be defined with one of the following date-time edit masks: DATETIME, TIMESTAMP, or NATTIME).

- The daylight savings indicator must precede the corresponding date-time field. For example:

  ```
  AAD,AA,14,U,E(DATETIME),GE.
  ```

Be sure to read *Daylight Savings Indicator Rules* for more information about daylight savings indicator usage in format and search buffers.

S (Significance) and Null Indicators

For fields with the SQL null value compression option NC, a selection for "null" or "not null" can also be made using an "S" null indicator element similar to that described in the section *The SQL Significance Indicator and Field Series Notation*.

Note:
The NC option cannot be applied to fields with the NU (null-value suppression), FI (fixed storage), MU (multiple-value), or PE (periodic group) options, or to group fields.

The SQL significance ("S") indicator must be added to the field name ("field-nameS") and the corresponding SQL null indicator must be specified in the value buffer.

The following hexadecimal null indicators are allowed as search argument values:

FFFF   select null values

0000   select non-null values

Any other null indicator value causes an Adabas response code 52 (ADARSP052).

The null indicator (hexadecimal FFFF or 0000) has a standard length of two bytes and fixed-point format; this length and format cannot be overridden.

The "S" indicator can only be used with the equals (=) value-operator; using S with any other value operators causes an Adabas response code 61 (ADARSP061).

Examples:

The S significance operator is part of the search argument for the field AA.

```
AAS.
```

Select records with the FN field value of packed +1 and the AA field value of null (undefined):

Search Buffer	FN , 2 , P , D , AAS.	Search argument
Value Buffer	001F FFFF	Field value specification

Note:
Insignificant null values are not stored in the index. This can cause a search-for-null operation to be quite costly for an application program's performance.

Select records with the FN field value of packed +1 and the AA field value of non-null:

Search Buffer	FN , 2 , P , D , AAS.	Search argument
Value Buffer	001F 0000	Field value specification

length

The length of the field/descriptor value as provided in the value buffer. If the length is omitted, the value in the value buffer must comply with the standard length of the field/descriptor, as shown in *Length and Data Format*.

format

The format of the field/descriptor value as provided in the value buffer. If the format is omitted, the value in the value buffer must comply with the standard format of the field/descriptor, as shown in *Length and Data Format*.

value-operator

A value-operator indicates the logical operation to be performed between the preceding descriptor and its corresponding value in the value buffer.

The following operators may be specified:

Operator	Description
EQ (or) =	equals
GE	greater than or equal to
GT (or) >	greater than
LE	less than or equal to
LT (or) <	less than
NE	not equal to

If no value-operator is specified, an equals (EQ) operation is assumed.

Examples:

The following search buffer examples show the use of a value-operator:

Example	Description
AA.	AA equals the value specified in the value buffer (the default)
AA,LT.	AA is less than the value specified in the value buffer
AA,GE.	AA is greater than or equal to the value specified in the value buffer.

The following search buffer using the NE (not equal to) operator selects all records with the FN field not equal to "MIKE":

Search Buffer	`FN,4,A,NE.`	Search argument
Value Buffer	MIKE	Field value specification

Replacing the NE operator in this example with EQ (equal to) would select only records with FN field with values of "MIKE".

Connecting Search Expressions

A *connecting operator* may be used to connect search expressions. The permissible connecting operators are as follows:

Operator	Description
D	The results of two search expressions are to be combined using a logical AND operation. For example: `AA,D,AB.`

O	The results of two search expressions are to be combined using a logical OR operation. The OR operator may only be used to connect search expressions which use the same descriptor. Here is a valid and an invalid example:
	
	Valid:
	
	`AA,O,AA.`
	
	Invalid (two different descriptors are used):
	
	`AA,O,AB.`

R	Fields or command IDs that point to ISN lists derived from different descriptors are to be combined using a logical OR operation. For example: `AA,5,A,R,AB,LT.`

S	A FROM-TO range (inclusive) which involves two search expressions. The same descriptor must be used in both expressions. Here is a valid and an invalid example:
	
	Valid:
	
	`AA,S,AA.`
	
	Invalid (two different descriptors are used):
	
	`AA,S,AB.`

Operator	Description
N	Excludes a single value or a range of values from the immediately preceding FROM-TO range. This operator can only be specified in conjunction with the S operator, and must apply to the same field specified in the FROM-TO range. Phonetic descriptors cannot be specified. Here are some valid and invalid examples: ● Valid: <pre>AA,S,AA,N,AA.</pre> Invalid (two different descriptors are used): <pre>AA,S,AA,N,AB.</pre> ● Valid: <pre>AA,S,AA,N,AA,S,AA.</pre> Invalid (two different descriptors are used): <pre>AA,S,AA,N,AA,S,AB.</pre> <pre>AA,S,AA,N,AA,N,AB.</pre>
Y	The results of any number of D, O, R, S, and N search operations can be combined using a logical AND operation. For example: <pre>AA,D,AB,Y,AA,O,AA,Y,AA,S,AA,N,AA,S,AA.</pre> The Y connecting operator functions like parentheses: only one level is allowed; that is, nested parentheses are not supported. All search expressions connected with the Y operator must apply to the same file.

If different operators are used within a single search buffer argument, the operators are processed in the following sequence:

1. Evaluate all S operations, as described in this documentation.

2. Evaluate all N operations, as described in this documentation.

3. Evaluate all O operations, as described in this documentation.

4. Evaluate all D operations, if needed.

5. Evaluate all R operations, if needed.

6. Evaluate all Y operations, if needed.

Example:

The following search buffer:

```
AA,S,AA,O,AA,D,AB,R,AC,D,AD.
```

is processed in this sequence:

```
( ( (AA,S,AA),O,AA),D,AB),R,(AC,D,AD)
```

The following search buffer:

```
AA,D,AB,Y,AA,O,AA,Y,AA,S,AA,N,AA,S,AA.
```

is processed in this sequence:

```
(AA,D,AB),Y,(AA,O,AA),Y,((AA,S,AA),N,(AA,S,AA))
```

Searching One File

The following syntax statement is relevant when searching fields in a single file:

search-expression [{,*connecting-operator*,*search-expression*}...] .

For the syntax of the *search-expression*, read *Search Expression*. For information about the *connecting-operator*, read *Connecting Search Expressions*.

Searching Multiple, Physically Coupled Files

The following syntax statement is relevant for multiple-file searches in which fields from two or more physically coupled files are to be used:

/file-x/search-expression[{,*connecting-operator*,*search-expression*}...]
{,D,*/file-y search-expression*/[{,*connecting-operator*,*search-expression*}...]}... .

where *file-x* and *file-y* are the file numbers of the physically coupled files. For information about search buffer syntax using physically coupled files, see *Physically Coupled Files*. For the syntax of the *search-expression*, read *Search Expression*. For information about the *connecting-operator*, read *Connecting Search Expressions*.

Searching One or More Files Using Soft Coupling

The following syntax statement is relevant for searching one or more files using soft coupling:

```
(m-file,m-field,s-file,s-field[{;m-file,m-field,s-file,s-field }...])/s-file-x/search-expression[{,connecting-operator,search-expression}... ]
[{,D,/s-file-y/search-expression[{,connecting-operator,search-expression}...]}...] .
```

where m-file and s-file are...., m-field*s-file-x* and *s-file-y* are the file numbers of the softly coupled files.

For information about search buffer syntax using soft coupling, see *Soft Coupling*. For the syntax of the *search-expression*, read *Search Expression*. For information about the *connecting-operator*, read *Connecting Search Expressions*.

Physically Coupled Files

The syntax of the search buffer for a multiple-file search in which fields from two or more physically coupled files are to be used is:

```
/file-x/search-expression[{,connecting-operator,search-expression}...]
{,D,/file-y search-expression/[{,connecting-operator,search-expression}...]}... .
```

The search criteria of the physically coupled files can be specified in any order. The ISN values actually returned are from the coupled file specified by the Adabas control block's file number field; this file is called the "primary" file.

The elements in this syntax are now described:

file-x **and** *file-y*

> The file numbers of the physically coupled files. All files specified must have been previously coupled using the COUPLE function of the ADAINV utility. A file number can appear only once for a given file. The file number must immediately precede its search criteria (consisting of one or more search expressions and appropriate connecting operators). A maximum of five (5) files may be specified in a single search buffer for physically coupled files.

D

> The only connecting operator allowed between the search criteria of the physically coupled files is the AND (D) symbol.

search-expression

> A search expression for the associated physically coupled file. For the syntax of the *search-expression*, read *Search Expression*.

connecting-operator

A connecting operator to connect the search expressions of the search criteria for an individual physically coupled file. While the connecting operator between search criteria for the physically coupled files must be "D" (AND), the connecting operators between the search expressions that comprise the search criteria for an individual file can be any of the operators described in *Connecting Search Expressions*.

Example:

Find the ISNs of all the records in file 1 that contain the three-byte (length override) unpacked decimal (format override) value "+20" in their AB fields, and that are coupled to records in file 2 containing the value "ABCDE" for field RB, which has a standard length of ten bytes and an alphanumeric format.

Search Buffer	/1/AB,3,U,D,/2/RB.	Search argument
Value Buffer	character-notation 020ABCDEbbbbb hex-notation F0F2F0C1C2C3C4C54040404040	Field value specification

Soft Coupling

The syntax of search buffer for a search in which soft coupling is to be used is:

```
(m-file,m-field,s-file,s-field[{;m-file,m-field,s-file,s-field }...])/s-file-x/search-expression[{,connecting-operator,search-expression}... ]
[{,D,/s-file-y/search-expression[{,connecting-operator,search-expression}...]}...] .
```

The elements in this syntax are now described:

m-file, *m-field*

For *m-file*, specify the number of the main file. This file must also be specified in the file number field of the Adabas control block. The final resulting ISN list will include ISNs contained in the main file only.

For *m-field*, specify the field in the main file that is to be used as the soft-coupling link field. This field must be a descriptor, subdescriptor, superdescriptor, or hyperdescriptor. It may not be a long alphanumeric field or be contained within a periodic group.

The combination of *m-file*, *m-field*, *s-file*, and *s-field* specifications comprise a single soft coupling. A maximum of 42 soft-coupling criteria may be specified. All of the soft coupling must be specified in one set of parentheses.

s-file, *s-field*

For *s-file*, specify the file number of the search file; for *s-field*, specify a field within the search file. For each ISN selected from this search file (according to the search criteria), the field specified as *s-field* will be read. The value of the field will then be used to determine which ISNs in the main file have a matching value.

The field may be a descriptor or nondescriptor; it can be a subdescriptor, superdescriptor, hyperdescriptor, or a long alphanumeric field. It must have the same format as the corresponding *m-field*. The standard length may be different. The field may not be contained within a periodic group.

The combination of *m-file*, *m-field*, *s-file*, and *s-field* specifications comprise a single soft coupling. A maximum of 42 soft-coupling criteria may be specified. All of the soft coupling must be specified in one set of parentheses.

s-file-x and *s-file-y*

The file number of the coupled files for which you want to specify search criteria. A file number can appear only once for a given file. The file number must immediately precede its search criteria (consisting of one or more search expressions and appropriate connecting operators). A maximum of five (5) files may be specified in a single search buffer.

D

The only connecting operator allowed between the search criteria of the coupled files is the AND (D) symbol.

search-expression

A search expression for the associated coupled file. For the syntax of the *search-expression*, read *Search Expression*.

connecting-operator

A connecting operator to connect the search expressions of the search criteria for an individual coupled file. While the connecting operator between search criteria for the physically coupled files must be "D" (AND), the connecting operators between the search expressions that comprise the search criteria for an individual file can be any of the operators described in *Connecting Search Expressions*.