

Using extreme files

This section details specific programming consideration when dealing with extreme partitioned files.

- General Restrictions
 - Extreme file capacity
 - Well-formed application logic
 - Application logic that is not well-formed
 - Field style extreme files
 - ISN style extreme files
-

General Restrictions

- The CONVISN API is not supported against extreme partitioned files.
- Where ISN lists are returned these will not be in ascending ISN sequence across partition because the partition identity is no longer part of the 4-byte ISN value for extreme files

Extreme file capacity

The mathematical maximum records that can be held for an extreme file (across all the partitions) is determined by the maximum number of partitions multiplied by the maximum number of ISNs per partition.

1. The maximum number of partitions is 65535.
2. The maximum number of ISNs per partition is 4,294,967,295.

Well-formed application logic

Good programming practices generally result in well-formed application logic. For the purposes of Vista, here are examples of well-formed application logic.

A typical nested access connecting records of two files:

```

READ file-one BY abc FROM '1001'
    Some processing on the *ISN(file-one)
    FIND file-two WITH def = def(file-one)
        Some processing on the *ISN(file-two) and perhaps also *ISN(file-one)
    END-FIND
END-READ

```

Add a modification into this (no END TRANSACTION is used to simplify the illustration):

```

READ file-one BY abc FROM '1001'
    Some processing on the *ISN(file-one)
    FIND file-two WITH def = def(file-one)
        Some processing on the *ISN(file-two) and perhaps also *ISN(file-one)
        UPDATE (or DELETE) *ISN(file-two)
    END-FIND
    UPDATE (or DELETE) *ISN(file-one)
END-READ

```

For the purposes of Vista extreme files, this logic is well-formed because the iterations deal with one record at a time, the current record. In Natural programming this is typified by *ISN. Most application logic is programmed in this way, for no other reason than it makes sense because sensible application logic is cost-efficient to maintain. Vista supports this logic transparently without regard for the style of extreme partition in use. This is possible because Vista remembers in memory the partition details for the current record, and doesn't therefore have to use the special field (or 8-byte ISN) in these cases.

See Source Type in the Adabas Vista Parameters documentation for information on how to select the appropriate source type for using extreme files with well-formed applications.

Application logic that is not well-formed

In modern applications it is quite difficult to produce poorly-formed logic simply because the language syntax used today are themselves well-formed. Natural is an excellent example of that. However there are special cases where this may be necessary or there may be very old applications (usually 3GL) that are called where the logic is not so well-formed. These situations give rise to exceptions, which means you may have to adjust the logic to accommodate extreme files (and perhaps even standard partitioned files).

One classic example is where an application makes use of USERISN for a file. In this situation the application can use an ISN on a command that it has previously stored, in a file or in memory, meaning the command bears no relationship to any concept of current record. USERISN is not the only situation this arises by it is a classic case; and it is not recommended – a clear indication of not being well-formed. There are many other situations where an ISN-based command is used that otherwise conflicts with what would otherwise be considered *current* record.

All is not lost, Vista supports these situations, but in order to do so you need to make small changes to the application in these usually relatively few areas. You can modify the application logic to instruct Vista about the partition information of the command that is about to be issued using an API called PARTID. See the section API function overview for more information.

Field style extreme files

Field style extreme partitioned files require that the FDT is adjusted to have a new field with a) length 4 b) format packed and c) null suppressed. The 2-character name of this field can be any valid Adabas short name.

Note:

For Natural: the new field must be defined as "P 7" in the DDM(s).

The application [or Natural] must be sure to name this field at the front of all Format Buffers (DDMs) when anything other than well-formed application logic (or the PARTID API) is not used. See the section API function overview for more information.

See Source Type in the Adabas Vista Parameters documentation for information on how to select the appropriate source type for using field style extreme files.

Notes:

1. ET/BT options M/P not currently supported.
2. Multi-fetch requires all format buffers to use the extreme-field.
3. S2 commands are only supported with ISN buffer length less than or equal to 4.
4. S9 commands sorting a supplied ISN list in the ISN buffer **MUST** be preceded by a PARTID NEXT API call to set the partition-ids of the ISNs in the list. An error will occur if a partition-id list is detected from a previous PARTID NEXT call, but the current command does not require one.

ISN style extreme files

These files require use of ACBX commands only.

Use of the PARTID API is also supported.

Where ISN lists are returned (Sx commands, L9 option I), the buffer to contain the ISNs should be doubled in size. This is because the contents will go from a list of 4 byte ISNs to 8 byte. Similarly, multi-fetch will return an enhanced multi-fetch buffer with increased ISN related fields:

ISN or Multifetch Buffer: *RDE* count{*RDE1* }...

A record descriptor element (RDE) has the structure shown in the following table.

Format	Length	Content
All fields unsigned integer, right aligned	4 bytes	Length of this record in record buffer. Records may have different lengths.
	4 bytes	Adabas response for this record. If a nonzero response is given, no record is stored in the record buffer.
	8 bytes	Partition ID(4 bytes)+ ISN for this record (4 bytes)
	8 bytes	Rightmost 4 bytes: (L9 only) ISN quantity: value count for this descriptor.

See Source Type in the Adabas Vista Parameters documentation for information on how to select the appropriate source type for using ISN style extreme files.

Notes:

1. Command must be ACBX format.
2. ET/BT options M/P not currently supported.
3. L9 option I only supported for single segment calls.