# SSDN Formats Used in IRMs

*Field definitions*

July 16, 2001

Only a few structure definitions pertain to SSDNs used for Acnet device specification. This note describes each one in common use.

The general layout for an SSDN includes 4 sixteen-bit words with the following fields:

| *Word#* | *Mask* | *Meaning* |
|---|---|---|
| 1 | FF00 | Listype number (00–7F) |
| 1 | 00F0 | flags (0, 1, 2) |
| 1 | 000F | ident length (1 or 2) |
| 2 | FFFF | Node# of the front-end that sources this data |
| 3 | FFFF | Ident (channel# or other index#) |
| 4 | FFFF | Ident second word (or 00FF data item size) |

The listype number indicates the type of data being referenced. The ident number is usually an index number that indicates the item (of that type) being referenced. An ident is somewhat more general than this, but for Acnet SSDNs, only two ident lengths can be used, those having lengths of one or two words. The most typical case by far uses only one word; in that case, the 4$^{th}$ word can be used to specify a data item length in bytes.

The simplest example is a Reading property SSDN of an analog channel. Assume that the front end node is 0612 and the channel number is 0120. In that case, the ident length is one word (in addition to the node#). The listype number is 00 for analog readings. The flags are not used, and the item size need not be specified. Here is the resultant SSDN:

```
0001/0612/0120/0000
```

The item size is not needed, but the data size used in an Acnet RETDAT request should be 2 bytes. The offset word in the Acnet request should only be zero.

A similar example would be used for the Setting property SSDN in the case that the device is settable. If the device is a D/A-type settable device, the listype should be 01. The rest of the fields remain the same. (In the case of a motor-type device, the listype would be 07.)

If the same device has an analog alarm block, the listype should be 02. The request would specify a value from 2–20 bytes, usually 20 bytes.

If the device has a Basic Status property, the SSDN would probably appear much as it does for the analog reading case, since the same analog data table is used to house such 16-bit status values. An internal flag bit for such a channel will be set to assure that alarm scanning for such a device is done via digital logic (pattern/mask) rather than analog logic (nominal/tolerance).

The flags nibble influences how offset word specification is supported. This is most often used for cases in which one defines an Acnet device that can access any of a number of analog channels by using an appropriate offset as a kind of ident offset value. An application making such a request would need to be aware of the meaning of the channel numbers in a series. One might do this for a special application that accesses any item of an array of similar items. The `SSDN` would specify the base channel number ident, and the offset word would indicate the offset to the channel number actually sought. Most of the time this is used is for a generic device that can access any analog channel within a given node. In that case, the channel number in the `SSDN` would be `0000`, and the offset would then be the actual channel number sought. To specify such a case, the flags nibble would have the value 1.

But the flags nibble can be used in connection with two-word offsets as well. In that case, the specified offset word is added to the two-word ident value. If the two-word ident is a memory address, for example, this permits a single Acnet device to access 64 kilobytes of consecutive memory. This case would still specify the flags nibble as 1, but the ident length would be specified as 2. A variation is permitted to allow access to a larger region of memory by specifying the flags nibble value of 2. In this variation, the offset multiplied by 256 is added to the base address; thus, the range of memory accessible is 16 megabytes. Access to direct memory addresses in an `SSDN` is unusual, but it has been used on occasion. The usual listype used in such cases is `1D`, which specifies memory access by 16-bit words.

Most often, the flags nibble is `0`, implying that there is no support for specifying an offset word. But there is a special exception made to support access to waveform data for the case of fast digitizers. For a device using listype `00`, if the channel number ident value is related to a fast digitizer (via an entry in the internal `CINFO` table), then one may access the waveform by using a byte offset. (The flags nibble is still `0` for this case.)

When does the data item size byte get used? It is needed when accessing an array of channel readings from consecutive channels. For Booster HLRF, the structure containing logs of station trips is housed in consecutive analog channels. The special application that accesses this data knows what its structure is, and there is no need to define separate Acnet devices to access each 2-byte component of the trip log structure. The data item size should be specified as `02` in this case, so that only 2 bytes will be taken from each consecutive analog data table entry. If the trip log information is housed in 15 consecutive words, the request should ask for 30 bytes. Two bytes will be copied from each of 15 consecutive channels to fulfill the request. Again, this is not the usual case.

***Node numbers and EMX specifications***

A front-end node number is defined for each IRM. Front-end node `0612`, for example, is registered with the Domain Name Server as `node0612.fnal.gov`. But Acnet uses a different method for numbering front-end nodes. For this particular node, the Acnet node number is `0999`. Although the `DABBEL` user does not have to specify an Acnet node number in the `SSDN`, she/he does have to specify it in the `EMX` field specification. To specify an `EMX` for the simple analog channel device mentioned above, use the following:

```
4099/0009/0120/0000
```

Note that the `0999` value is split into two bytes. Acnet refers to the `09` byte as the trunk and to the `99` byte as the node. This document refers to `0999` as the node number. Note that the hi byte of the second word is `00`. This signifies an analog channel alarm, which is also used for Acnet digital alarms, since all such data comes from the analog data table. The only time that this byte is nonzero, it has the value `02`, which is used for "comment" alarms, called event alarms by Acnet. A comment alarm is most commonly used to declare that a front-end has reset.

Each Acnet device has an associated source node number that Acnet consoles use to target data requests. For most IRM devices, a server node is used for this source node number. The server node scans the second word of each `SSDN` in the data request to decide how to handle the request. If it finds that not all `SSDN`s specify its own node number, the server node supports the request as a server; it gathers the information from all the nodes specified in the `SSDN`s, and on receipt of appropriate answer fragments from each one, it returns the composite response to the Acnet console requester. To the requesting node, a simple request results in a simple response. The server node, however, must handle a more complex scenario that may involve forwarding of the original request via multicast addressing and arrangement of the resulting answer fragments so the requesting node cannot tell the difference. The reason for server node support is to accomplish delivery of the reply (to an Acnet request that includes data from several different nodes) in a single reply message. Acnet does not now support correlated data from different front-ends; this scheme increases the chance that an application program can actually collect correlated data.