FROM: Peter Johansson
TO: X3T10 SCSI-3 Working Group
DATE: September 9, 1996
RE: Enhanced Configuration ROM for SBP-2 Devices

If SBP2 is to be able to support devices that implement a diversity of command sets, provision must be made within configuration ROM to describe the software interfaces to these command sets.

The attached document is a proposed replacement for clause 7.3, Unit directory, in Revision 1b of the SBP-2 working draft. The proposal adds new entry types to the unit directory that permit the specification of an organization responsible for both the definition of the command set and the command set version used by the unit. In addition a new entry is defined to permit the specification of logical unit characteristics.

The pages that follow reproduce the current clause 7.3 and show change bars for the proposed modifications.
7.3 Unit directory

Configuration ROM for SBP-2 units shall contain a unit directory in the format specified by this standard. The unit directory shall contain Unit_Spec_ID and Unit_SW_Version entries, as specified by ISO/IEC 13213:1994.

The SBP-2 unit directory shall additionally contain a Command_Set_Spec_ID, Command_Set_Version, Management_Agent and Unit_Characteristics entry and one or more Logical_Unit entries, as defined by this standard.

7.3.1 Unit_Spec_ID entry

The Unit_Spec_ID entry is an immediate entry in the SBP-2 unit directory that specifies the organization responsible for the architectural definition of the unit. Figure 45 shows the format of this entry.

\[\begin{array}{c|c}
\text{most significant} & \text{least significant} \\
12_{16} & 00 \ 609E_{16} \\
\end{array}\]

Figure 45 Unit_Spec_ID entry format

12_{16} is the concatenation of key_type and key_value for the Unit_Spec_ID entry.

00 609E_{16} is the unit_spec_ID obtained from the IEEE/RAC. The value indicates that the X3 Secretariat is responsible for the software interface definition.

7.3.2 Unit_SW_Version entry

The Unit_SW_Version entry is an immediate entry in the SBP-2 unit directory that, in combination with the unit_sw_version, specifies the software interface of the unit. Figure 46 shows the format of this entry.

\[\begin{array}{c|c}
\text{most significant} & \text{least significant} \\
13_{16} & 01 \ 0483_{16} \\
\end{array}\]

Figure 46 Unit_SW_Version entry format

13_{16} is the concatenation of key_type and key_value for the Unit_SW_Version entry.

01 0483_{16} is the unit_sw_version value that indicates that the unit conforms to this standard.

7.3.3 Command_Set_Spec_ID entry

The Command_Set_Spec_ID entry is an immediate entry in the SBP-2 unit directory that specifies the organization responsible for the command set definition for the unit. Figure 47 shows the format of this entry.

\[\begin{array}{c|c}
\text{most significant} & \text{least significant} \\
38_{16} & \text{command_set_spec_ID} \\
\end{array}\]

Figure 47 Command_Set_Spec_ID entry format
**38_16 is the concatenation of key_type and key_value for the Command_Set_Spec_ID entry.**

The command_set_spec_ID is an organizationally unique identifier obtained from the IEEE/RAC. The organization to which this 24-bit identifier has been granted is responsible for the definition of the command set implemented by the unit.

7.3.4 Command_Set_Version entry

The Command_Set_Version entry is an immediate entry in the SBP-2 unit directory that, in combination with the command_set_spec_ID, specifies the command set implemented by the unit. Figure 48 shows the format of this entry.

```
+----------------------------------+
<table>
<thead>
<tr>
<th>most significant</th>
<th>least significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>39_16</td>
<td>command_set_version</td>
</tr>
</tbody>
</table>
+----------------------------------+
```

Figure 48 Command_Set_Version entry format

**39_16 is the concatenation of key_type and key_value for the Command_Set_Version entry.**

The meaning of command_set_version shall be specified by the owner of command_set_spec_ID.

7.3.5 Management_Agent entry

The Management_Agent entry is an immediate entry in the SBP-2 unit directory that specifies the base address of the Unit’s management agent CSR. Figure 49 shows the format of this entry.

```
+----------------------------------+
<table>
<thead>
<tr>
<th>most significant</th>
<th>least significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>7A_16</td>
<td>csr_offset</td>
</tr>
</tbody>
</table>
+----------------------------------+
```

Figure 49 Management_Agent entry format

**787A_16 is the concatenation of key_type and key_value for the Management_Agent entry.**

The csr_offset field shall contain the quadlet offset, from the base address of initial register space, FFFF F000 0000_16, to the base address of the MANAGEMENT_AGENT register for the unit. All SBP-2 unit CSR’s shall be located at or above address FFFF F001 0000_16; therefore the value of csr_offset shall not be less than 4000_16.

7.3.6 Unit_Characteristics entry

The Unit_Characteristics entry is an immediate entry in the SBP-2 unit directory that specifies characteristics of the unit implementation. Figure 50 shows the format of this entry.

```
+----------------------------------+
<table>
<thead>
<tr>
<th>most significant</th>
<th>least significant</th>
</tr>
</thead>
<tbody>
<tr>
<td>39_16</td>
<td>q</td>
</tr>
</tbody>
</table>
+----------------------------------+
```

Figure 50 Unit_Characteristics entry format

**393B_16 is the concatenation of key_type and key_value for the Unit_Characteristics entry.**
The \( q \) bit shall specify the task management (queuing) model implemented by the unit. If \( q \) is clear, the unit implements the basic task management model defined by this standard in 10.2. Otherwise the unit implements the full task management model.

The \textit{ordered} bit (abbreviated as \( o \) in the figure above) specifies the implicit task attribute implemented by the SBP-2 unit. If the unit implements SIMPLE tasks only, the \textit{ordered} bit shall be clear. Otherwise, if the unit implements ORDERED tasks only, the \textit{ordered} bit shall be set. When \textit{ordered} is set, the unit shall both execute tasks in order and report their completion status in the same order.

The \textit{isochronous} bit (abbreviated as \( i \) in the figure above) specifies whether or not the SBP-2 unit supports isochronous operations. When \textit{isochronous} is set, isochronous login requests, stream CDB requests and stream control requests shall all be supported. If the \textit{isochronous} bit is set, the \textit{irmc}, \textit{cmc} and \textit{isc} bits in the bus information block shall also be set, as described in 7.1.

The \textit{ORB size} field shall specify, in quadlets, the fetch size used by the SBP-2 unit to obtain ORB's from initiator memory. The initiator shall allocate, on a quadlet aligned boundary, at least this much memory for each ORB signaled to the target.

The \textit{extended request} bit (abbreviated as \( x \) in the figure above) specifies the ORB fetch size implemented by the unit. When \textit{extended request} is clear, the unit fetches 32 bytes when an ORB is referenced. If \textit{extended request} is set, the unit fetches 64 bytes, regardless of the actual size of the ORB referenced.

The \textit{ejectable} bit (abbreviated as \( e \) in the figure above) shall specify whether or not the unit medium is removable. When \textit{ejectable} is set it indicates that the medium is removable.

The \textit{multiple lu} bit (abbreviated as \( m \) in the figure above) encodes information about the number of logical units implemented by the SBP-2 unit. If the unit implements one logical unit with a logical unit number of zero, \textit{multiple lu} shall be clear.

The \textit{device type} field indicates the peripheral device type implemented by the SBP-2 unit. The \textit{device type} field shall be valid only if the \textit{multiple lu} bit is clear. When \textit{device type} is valid, it shall contain a value as specified by SPC for the peripheral device type returned in INQUIRY data. When the \textit{multiple lu} bit is set, \textit{device type} shall be 1F_{16}.

### 7.3.7 Logical_Unit entry

The Logical Unit entry is an immediate entry in the SBP-2 unit directory that specifies characteristics of logical units implemented by the SBP-2 device. There shall be at least one Logical Unit entry in the unit directory for logical unit zero. Additional Logical Unit entries are optional and may be used by SBP-2 devices that implement multiple logical units. Figure 51 shows the format of this entry.

<table>
<thead>
<tr>
<th>most significant</th>
<th>3C_{16}</th>
<th>e</th>
<th>r</th>
<th>device_type</th>
<th>lun</th>
<th>least significant</th>
</tr>
</thead>
</table>

\textbf{Figure 51 \ Logical Unit entry format}

\( 3C_{16} \) is the concatenation of \textit{key type} and \textit{key value} for the Logical Unit entry.

The \textit{ejectable} bit (abbreviated as \( e \) in the figure above) shall specify whether or not the unit medium is removable. When \textit{ejectable} is set it indicates that the medium is removable.
The `device_type` field indicates the peripheral device type implemented by the logical unit. This field shall contain a value as specified by SPC for Peripheral Device Type.

The `lun` field shall identify the logical unit to which the information in the Logical Unit entry applies.