HEWLETT® PACKARD

Using SI Tables to Create Electronic Program Guides.

Application Note.

Introduction.

The Electronic Program Guide (EPG) is a feature unique to the digital TV world. It's the electronic equivalent of the TV page in a newspaper. Created by the service provider, the EPG is used to entice consumers to purchase programming. Since pay per view is the primary mechanism, service providers will use to re-coup their investment: an effective, error-free EPG is a vital. Much time and effort is spent creating innovative EPG features. A combination of text, still images, and video clips; these features create a powerful advertising message for an event.

The protocol used to create the EPG is known as the Service Information (SI) tables. If something goes wrong with the SI, the user cannot view what's available, and is unable to make a purchasing decision. So whereas errors in the MPEG-2 video/audio data will cause annoyance, errors in the SI and hence EPG, will have an immediate effect on the service providers revenue stream.

In this article we will discuss the SI. Both from the point of view of how it's used to create an EPG, and what effect SI errors may have on the EPG

Service Information Tables (the SI): DVB & ATSC.

Most literature concentrates on the Program Specific Information (PSI). Defined in the MPEG-2 systems specification, the PSI Consists of the Program Association Table, the Program Map Table's, and the Conditional Access Table. The PSI tables provide the information required to decrypt and display a chosen event.

The SI provides information on available services; what frequencies carry them; it groups services into common interest categories; and provides information on the events in a service. The SI contains information used by network operators: for instance, which network originated the service. The SI allow purchasing decisions to be made, and help manage the flow of revenue to service providers.

SI table structure for DVB & ATSC systems.

We will describe the four main tables of the DVB-SI: the NIT, BAT, SDT & EIT. And the equivalent structure in the ATSC-SI. Some other tables exist, but these are the key ones.

Within the DVB system, each table is split into sections. Each section has a header, and either two (SDT & EIT), or three (NIT & BAT) loops containing descriptors specific to each table. A table can be made up of one or more sections, the section_number being incremented each time a new table section is sent. Since the section_number is an eight-bit field, the maximum number of sections per table is 256. The section length for all tables is 1021 bytes, apart from the EIT which is 4096. The table descriptors carry information relevant to its function. The descriptors are complex structures in themselves, and can be sources of subtle errors. Each section will contain the header, and either part or all the looped information pertaining to the table.

The PID for the transport stream packet containing the table, along with the table id (contained in the table section header), identifies the tables type & status. Note that the BAT and SDT can be carried on the same PID. In this case it’s the table_id which uniquely specifies the table type. Table information changes are notified by incrementing the section version number.

We will describe the tables in general terms, highlighting important components in their headers and loops; describing their function; and comparing them with their ATSC equivalents. Figure One shows the overall structure of these four DVB-SI tables.

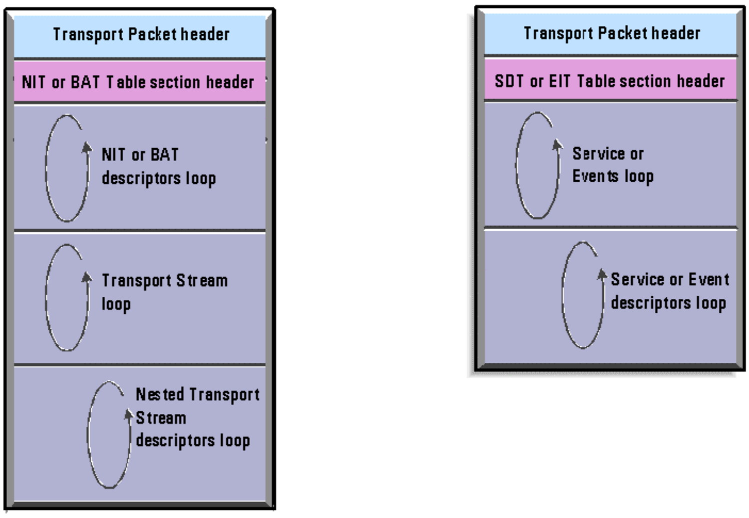
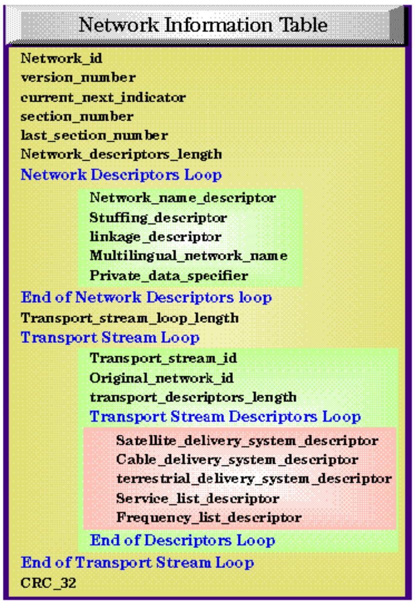
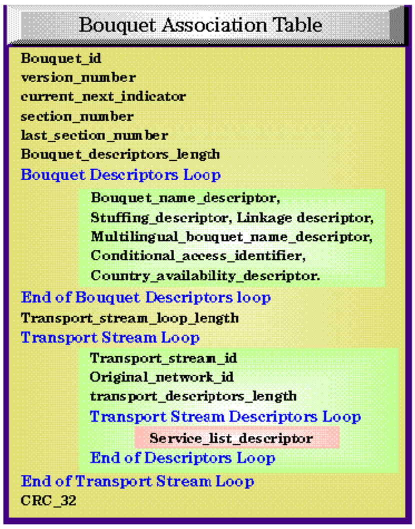

Transport Packet header
NIT or BAT Table section header
NIT or BAT descriptors loop
Transport Stream loop
Nested Transport Stream descriptors loop
Transport Packet header
SDT or EIT Table section header
Service or Events loop
Service or Event descriptors loop

Figure One: DVB Service Information Table Structure.


Network Information Table

Network_id
version_number
current_next_indicator
section_number
last_section_number
Network_descriptors_length
Network Descriptors Loop
Network_name_descriptor
Stuffing_descriptor
linkage_descriptor
Multilingual_network_name
Private_data_specifier
End of Network Descriptors loop
Transport_stream_loop_length
Transport Stream Loop
Transport_stream_id
Original_network_id
transport_descriptors_length
Transport Stream Descriptors Loop
Satellite_delivery_system_descriptor
Cable_delivery_system_descriptor
terrestrial_delivery_system_descriptor
Service_list_descriptor
Frequency_list_descriptor
End of Descriptors Loop
End of Transport Stream Loop
CRC_32


Bouquet Association Table

Bouquet_id
version_number
current_next_indicator
section_number
last_section_number
Bouquet_descriptors_length
Bouquet Descriptors Loop
Bouquet_name_descriptor,
Stuffing_descriptor, Linkage descriptor,
Multilingual_bouquet_name_descriptor,
Conditional_access_identifier,
Country_availability_descriptor.
End of Bouquet Descriptors loop
Transport_stream_loop_length
Transport Stream Loop
Transport_stream_id
Original_network_id
transport_descriptors_length
Transport Stream Descriptors Loop
Service_list_descriptor
End of Descriptors Loop
End of Transport Stream Loop
CRC_32

Figure Two: DVB-SI Network Information & Bouquet Association Tables.

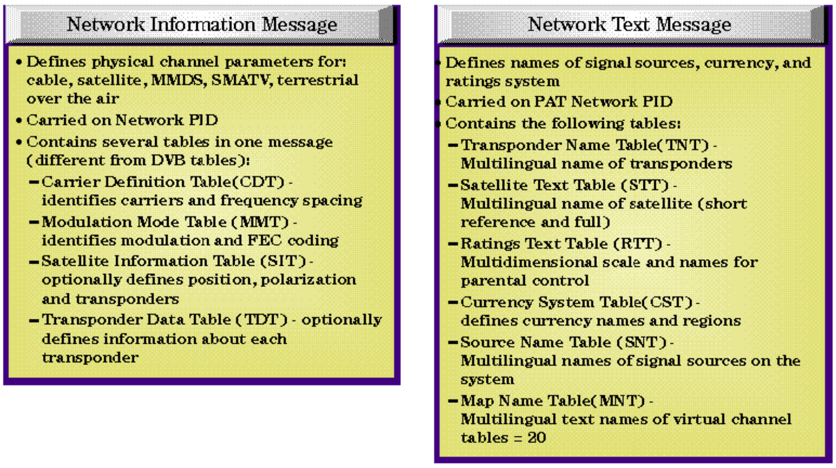
The ATSC-SI structure differs in several ways. It has a fixed PID to program number relationship. Since the program number is listed in the SI (service_id in DVB, source_id in ATSC), you can bypass the PAT/PMT and go direct to the content. Secondly, ATSC only requires one set of tables for all transport streams: DVB sends duplicates to describe other streams. These differences are meant to speed up event acquisition in ATSC. ATSC-SI syntax is almost identical to DVB-SI: with table sections, headers, and loops. I will only discuss DVB-SI syntax in detail: ATSC-SI syntax is practically identical.

DVB Network Information Table (NIT) & ATSC Network Information/Text Messages.

NIT sections always use PID 10hex, and carry network-wide information. Each network described is identified by its network_id. The NIT table_id has one of two values: 40hex for sections describing the current network, 41hex if sections describe other networks. This allows rapid translation of information at Network boundaries, e.g. when moving from a satellite to cable TV network. The current/next indicator allows a NIT to be inserted that isn't valid yet: again allowing rapid acquisition of new parameters, such as new rf channel frequencies.

Figure two illustrates the NIT structure. The Network descriptor loop carries network wide information. For instance, the linkage_descriptor provides links to other services describing the network. The transport stream loop identifies each stream carried, and its originating network (useful for billing). An inner transport stream descriptor loop contains network information on each transport stream. For example, if the stb is tuned to one rf channel, and the consumer requests a service on a different channel. The frequency for this channel is contained in this loop.

Figure three shows the equivalent ATSC-SI tables. The Network Information Message (carries physical channel parameters), and the Network Text Message (contains names of services, currency, and parental ratings info). Both are carried on the Network PID, and contain several sub-tables. Collectively, with the Virtual Channel Message, these tables are known as the Network Stream, and carry network-wide information. Note that unlike the DVB-NIT, the ATSC Network Stream cannot identify a services originating network.



Network Information Message

- Defines physical channel parameters for: cable, satellite, MMDS, SMATV, terrestrial over the air
- Carried on Network PID
- Contains several tables in one message (different from DVB tables):
  - Carrier Definition Table(CDT) - identifies carriers and frequency spacing
  - Modulation Mode Table (MMT) - identifies modulation and FEC coding
  - Satellite Information Table (SIT) - optionally defines position, polarization and transponders
  - Transponder Data Table (TDT) - optionally defines information about each transponder

Network Text Message

- Defines names of signal sources, currency, and ratings system
- Carried on PAT Network PID
- Contains the following tables:
  - Transponder Name Table(TNT) - Multilingual name of transponders
  - Satellite Text Table (STT) - Multilingual name of satellite (short reference and full)
  - Ratings Text Table (RTT) - Multidimensional scale and names for parental control
  - Currency System Table(CST) - defines currency names and regions
  - Source Name Table (SNT) - Multilingual names of signal sources on the system
  - Map Name Table(MNT) - Multilingual text names of virtual channel tables = 20

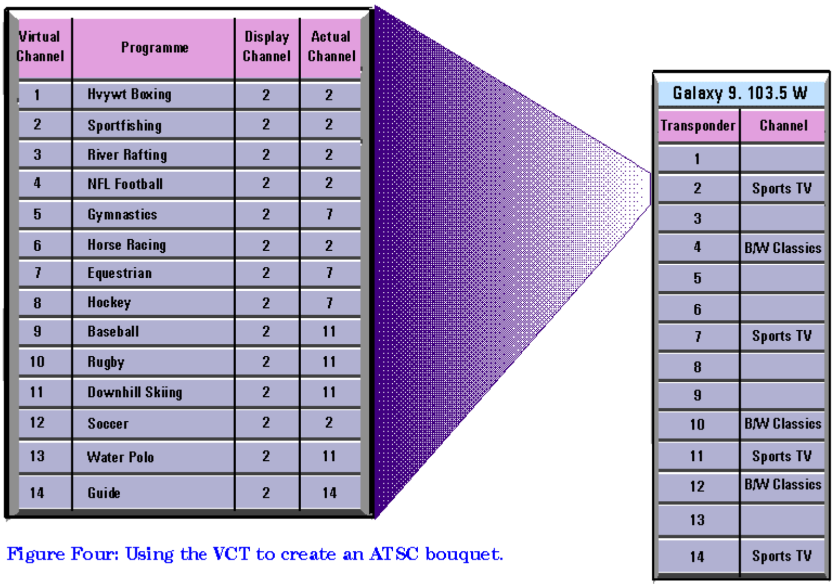
Figure Three: Network Information & Network Text Messages.

DVB Bouquet Association Table (BAT) & ATSC Virtual Channel Table.

Consider again figure two. Each BAT section describes all or part of a bouquet: sections identified by the BAT table_id (set to 4Ahex). The bouquet_id identifies which bouquet is described (e.g. sports services). The bouquet descriptor loop contains bouquet wide information, for instance its name and its country availability. The transport stream loop lists all the streams relevant to this bouquet. Its contents being identical to the NIT transport stream loop. The inner transport stream descriptor loop lists all services in that transport stream.

The BAT allows the grouping of common interest services in the EPG. For instance, a bouquet specifically for sports channels; with the transport stream loop listing all streams containing sports; the stream descriptor loop listing all sport services contained in each stream. The consumer can click on a sport services icon, get a listing, and then choose a service/event. The BAT is a powerful tool in the EPG designer's armoury.

In the ATSC-SI, the Virtual Channel Table is roughly analogous to the BAT. It's carried on the Network PID, and along with the Defined Channels Map (DCM), and Inverse Channels Map (ICM); is part of the Virtual Channel Message. The VCT contains the physical locations of all channels, transmission method, an application_ID and source_ID. It can be used to assign virtual channel numbers to each service. Figure 4 shows how this allows the creation of a seamless channel list from multiple sources: similar to the BAT.



Virtual Channel	Programme	Display Channel	Actual Channel
1	Hvywt Boxing	2	2
2	Sportfishing	2	2
3	River Rafting	2	2
4	NFL Football	2	2
5	Gymnastics	2	7
6	Horse Racing	2	2
7	Equestrian	2	7
8	Hockey	2	7
9	Baseball	2	11
10	Rugby	2	11
11	Downhill Skiing	2	11
12	Soccer	2	2
13	Water Polo	2	11
14	Guide	2	14

Galaxy 9. 103.5 W	
Transponder	Channel
1	
2	Sports TV
3	
4	B/W Classics
5	
6	
7	Sports TV
8	
9	
10	B/W Classics
11	Sports TV
12	B/W Classics
13	
14	Sports TV

Figure Four: Using the VCT to create an ATSC bouquet.

Comparing the Service & Event Information tables in the DVB and ATSC tables.

At this point the DVB & ATSC table structure start to diverge. Whereas the DVB-SI has neatly compartmentalized SDT's & EIT's, the ATSC-SI has a more complex system of tables.

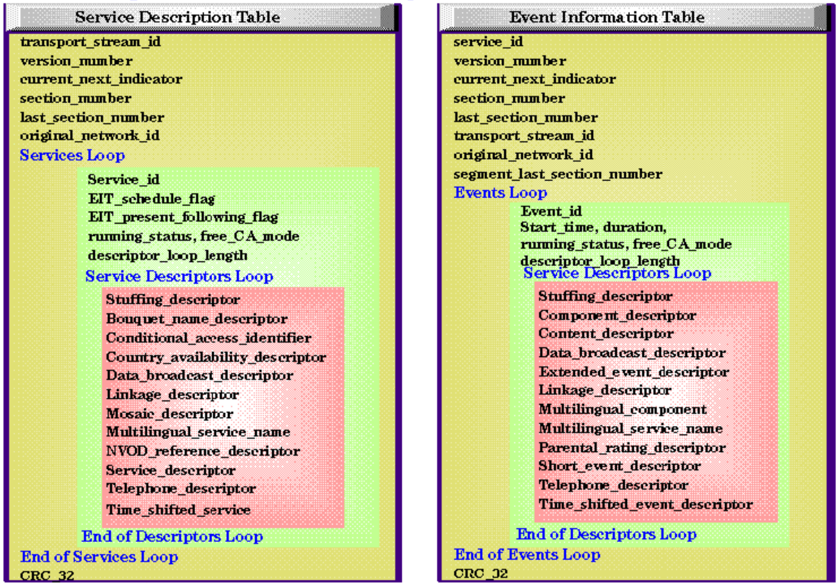


Figure Five: DVB Service Description & Event Information Tables.

The Service Description Table.

Figure 5 illustrates the DVB-SDT/EIT structure. SDT table sections describe the services carried by a transport stream. If the SDT table_id, is set to 42hex the section describes the current transport stream, if set to 46hex, it describes another transport stream. The header contains global information, such as transport stream and original network id's, section and version numbers. The service loop identifies each service (via the service_id), whether EIT information is available for each service, is the service-running etc. The inner service descriptor loop contains detailed information on each service listed in the outer loop. Such as what bouquet this service is on, a description of the service, telephone descriptors for interactivity etc. A unique descriptor in the DVB-SDT is the Mosaic function. This allows information on all services (or events) carried to be displayed on one screen. Including still images, MPEG-2 clips etc. EPG designers to sell services/events to the consumer can use the Mosaic.

The Event Information Table.

EIT's are carried on PID 12hex. The EIT has the same structure as the SDT, but four different types of EIT section can exist:

1. **Actual transport stream present/following event**. This contains information on the current and next event available: section table_id set to 4Ehex.
2. **Other transport stream present/following event.** Same as above, but for a different transport stream. Section table_id is set to 4Fhex.
3. **Actual transport stream event schedule**. Contains information on several events following the current event. The table_id is set to values between 50hex and 5Fhex.
4. **Other transport stream event schedule**. Table_id set to values between 60hex and 6Fhex.

The present/following EIT is distributed over two sections: with section_number 00hex reserved for present, and 01hex for following. Since the EIT schedule spans many days, it requires a more complex ordering of its sections.

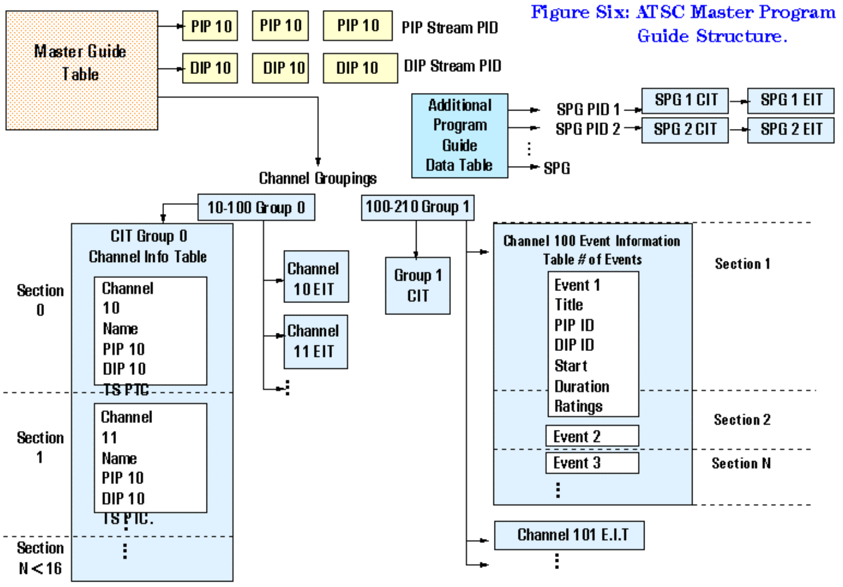


Figure Six: ATSC Master Program Guide Structure.

The 256 sections per table are divided into 32 segments of 8 sections each. Each segment contains event information for a three hour period, with segment0 of table_id=50hex covering events from 2400hrs to 02:59:59 etc. So that the 32 segments of table_id=50hex cover a four day period. With 16 table_id's available, a schedule/EIT covers 64 days of events. Table_id's and section_number's are ordered chronologically.

The service_id, transport_stream_id and original network_id identify the service, stream and network, the EIT section belongs to. Note the service_id is the same as the program_number in the corresponding program_map_section. The events loop lists all the events described by this section, their start time, length, encryption etc. The inner descriptor loop, associated with each event, carries detailed information on parental ratings, content description, telephone descriptors for interactivity etc.

ATSC Master Program Guide Tables.

Figure 6 illustrates the ATSC-MPG structure. The Master Guide Table (MGT) groups virtual channels in bundles of 64, with up to 16 bundles allowed. This is done to simplify data acquisition for large MPG's. The MGT defines the time for which the current MPG is valid. And contains the PID list, this includes PID's for the Descriptive Information Parcel (DIP) and Private Information Parcel (PIP): these carry detailed channel/event information text.

Once the channel group is identified, the Channel Information Table for that group is accessed. The CIT contains the information needed to determine the video, audio, and data PID's; physical transmission channel number; and virtual channel information on any available program in the MPG. The CIT information does not change during the lifetime of the MPG. The CIT along with the CIT DIP/PIP's is roughly equivalent to the DVB-SDT.

Finally, the Event Information Table is accessed. One EIT exists for each channel, so that each channel group will have up to 64 EIT's. It contains event information: programming event titles, start times etc. Along with the EIT-DIP/PIP's, the ATSC-EIT is equivalent to the DVB-EIT.

In addition to these tables, the MPG contains the Additional Guide Data Table. The AGDT carries time base information (pointers to PCR or other time base PID's), the Program Guide Map (defines optional special program guides), and the Default Override Record (allows service provider override of program ratings/themes).

Why the difference in DVB & ATSC structures?

The main reason for the ATSC-SI structure, is a desire to speed up access times to events, without increasing SI bandwidth. The single Master Program Guide, and fixed PID to program relationship, aims to do this. It remains to be seen whether the ATSC approach gives speed advantages over the DVB-SI method of carrying duplicate tables, that out-weighs the extra complexity of the ATSC-SI system.

How an EPG might Use the DVB-SI.

In this section we take the reader on a walk through the DVB-SI. From the point at which the user looks at the EPG, to the point at which the PSI are accessed.

The Death of Channel Hopping.

Why is it impossible to channel hop in a digital TV system? In an analog TV system, the complete TV program is modulated onto one of several frequencies. Tune into the frequency, and you see the picture. In the MPEG-2 multiplex, many program events are transmitted as a jumble of unconnected packets. RF channels are just a means of multiplexing transport streams, not individual programs. To reconstruct an event from its individual transport packets, the user

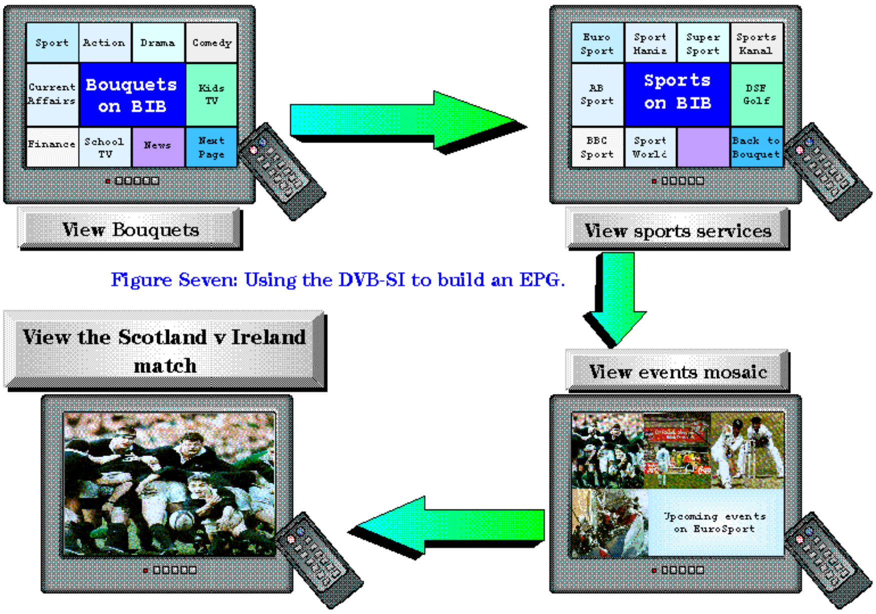


Figure Seven: Using the DVB-SI to build an EPG.

must choose the event, not the channel it’s on. This means it’s important the user can easily view information on all events.

Displaying a sporting event: an example.

Consider figure 7: We start from an EPG screen displaying event category icons (sports, movies etc.). Bouquet_name_descriptors extracted from the BAT are used to create these icons. If a particular BAT isn't available, for example an Adult Bouquet: this is reflected in the BAT descriptor loop country_availability_descriptor. We assume a sports bouquet is chosen.

Using the BAT to see if "Eurosport" is available.

Once a BAT is chosen, we switch to the second screen. This uses the service_list_descriptor, to display all the services available in the bouquet. The user wants to view Eurosport, and so the EPG needs to access the transport stream carrying this service. It takes the relevant transport_stream_id (contained in the BAT transport stream loop), and accesses the NIT. It cycles through the transport stream loop, until it gets a transport_stream_id match; cycles through the transport_stream_descriptor loop, extracts the frequency_descriptor, and tunes the stb to the correct rf channel.

Using the SDT to view what’s on "Eurosport".

Having tuned to the rf channel containing the right transport stream. Using the service_id (extracted from the BAT's Eurosport service_list_descriptor), the EPG looks for packets with PID=11hex & table_id=42hex: the current SDT. It cycles through the service loop, until it gets a service_id match. From the service loop, the EPG tells the user there’s an Event Schedule available and the service is running. From the services descriptor loop, it decodes the Mosaic descriptor, and creates a multi-cell screen showing a selection of events on Eurosport. It also tells the viewer they've got access to this service. The Mosaic is shown in the third screen in figure 7.

Using the EIT to choose an event on "Eurosport".

The users want to see the "Scotland v Ireland Rugby international". The EPG extracts the event_id from the mosaic, and looks for PID=12hex, table_id=4Ehex: present/following schedule for current transport stream. As the EPG has already recorded the service_id, it scans EIT section headers until it gets a service_id match. Knowing the event_id, it locks onto the correct event loop, and lists information about the event (start time, duration, a match profile, team lists etc.).

Viewing the event & paying for it!

Once the user accepts this event, the service_id (identical to the program number in the PAT) is used to find the relevant PMT for Eurosport. Decode & display can now commence.

Finally, using the original_network_id (retrieved from the BAT), the EPG can automatically dial back (telephone_descriptor in the event_descriptor loop), and provide payment to the original provider of this event.

Summary.

The scenario just played out is only one of many open to the designer of an EPG. I chose it, because it uses both the Bouquet and Mosaic concepts. It does not in any way reflect the philosophy of any particular service provider. A very similar scenario could be played out using the ATSC-SI, the route taken would be different, but the result would be the same. The thing to note is the dominance of the SI tables. It’s these that tell the decoder what to decode. The PSI very much take second place to the SI.

DVB & ATSC SI structure: faultfinding.

The scope for errors in the SI is very large, and the errors themselves may be very subtle. Unfortunately, even the subtlest of errors can have a dramatic effect on the EPG, and what the consumer sees. For each of the SI tables, we will give examples of what might happen, if certain parameters are incorrect. We will then discuss the type of equipment that’s required to identify and analyze these errors.

Possible NIT & NIM Errors.

The NIT contains three loops, two of which are nested. The innermost nested loop contains transport stream descriptors. One of which, the frequency_list_descriptor, contains a header and loop. So in effect, there are three nested loops. If a centre_frequency descriptor is incorrect, the stb will not tune to the channel containing that transport stream. The cable_delivery_system_descriptor is a loop describing the Forward Error Correction, modulation scheme, and symbol rate used in a cable system. If the Modulation descriptor is set to 02hex, rather than 03hex: equipment will think this is a 32 rather than 64-QAM transmission system. The ATSC-NIM contains similar information, and faults would cause similar errors.

Possible BAT/VCT Errors.

A typical BAT/VCT error, would be an incorrect service in a bouquet (errored service_id, or virtual channel) In the ATSC system VCT faults are really dangerous, since the VCT contains PID information. If faulty, an event cannot be decoded. If the BAT is at fault, the consumer may still scroll through each event (if the EIT and SDT are not corrupted). Note that the service_id is embedded in the service_list_descriptor, another 3rd level nested loop in the BAT.

Possible SDT Errors.

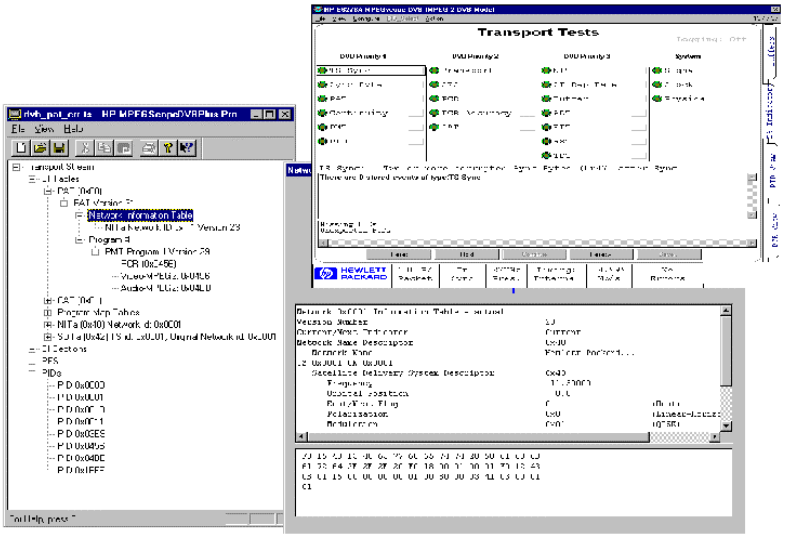
We will consider two errors, both subtle, but very serious if they occur. Firstly, what happens if the service_id (SDT) and program_number (PMT) do not match? An event is chosen, and the service carrying it, is identified via its service_id. But when the PAT is accessed, the program number is either totally wrong, or describes the wrong program. Secondly, the Mosaic descriptor is a sophisticated looped structure in itself. It can contain event_id's. If one of these is incorrect, then either the wrong event or non at all will be displayed.

Possible EIT Errors.

Taking aside possible numbering errors in the EIT/schedule, which would cause severe problems in reconstructing an event schedule. If we consider the EIT section structure itself, incorrect service to event_id linkage in the SDT header would make it impossible to decode events. An incorrect event_id would lead to the wrong event information being displayed. Whilst an incorrect parental_rating_descriptor, would give a false impression as to an events viewing suitability.

ATSC Master Program Guide Errors.

Very similar in nature to SDT/EIT errors, but with a complication. The separation of the SDT and EIT makes the compartmentalization of errors easier. The MPG structure is more complex: its sub-tables using information from the PIP's, DIP's, and MGT. This makes debugging more difficult. Secondly, the MPG doesn't need the PAT/PMT to identify event PID's. So a fault in the MPG, means an inability to decode. Whilst in the DVB-SI, there is still a chance that an event can be displayed.



```
Transport Stream
  SI Tables
    PAT (0x00)
      PAT Version 21
        Network Information Table
          NIT a Network ID ... Version 23
        Program 4
          PMT Program 1 Version 29
            PCR (0x045E)
            Video MPEG2: 0x04C6
            Audio MPEG2: 0x04L0
    CAT (0x01)
    Program Map Tables
    NIT a (0x40) Network id: 0x0C01
    SDT a (0x42) TS id: 0x0001, Original Network id: 0x0001
  SI Sections
  PES
  PIDs
    PID 0x0000
    PID 0x0001
    PID 0x0010
    PID 0x0011
    PID 0x03E8
    PID 0x0495
    PID 0x04DE
    PID 0x1FFF
For Help, press F1
```

```
TS Sync    Transport Error    Sync Byte Error
There are 0 stored events of type:TS Sync
```

```
Network 0x0C01 Information Table - Actual
Version Number                          23
Current/Next Indicator                  Current
Network Name Descriptor                 0x40
   Network Name                         ...
TS 0x0001 ON 0x0001
   Satellite Delivery System Descriptor 0x43
      Frequency                         11.70000
      Orbital Position                  0.0
      East/West Flag                    0          (West)
      Polarisation                      0x0        (Linear-Horiz
      Modulation                        0x01       (QPSK)
```

Figure Eight: Service Information Table decode & analysis using the HP E6277A.

Debugging DVB & ATSC SI.

What should test equipment used to debug the SI support? Firstly, due to the complexity of the SI table structure, being able to graphically display it, is a must. Secondly, the ability to compare sections of different SI tables would be invaluable: why? Because a service_id in the SDT, may not equate to the service_id in the EIT, or for that matter to the programme_number in the PAT. Thirdly, due to the impact table descriptor errors have, being able to decode them is very important.

Errors can either be of a random or systematic nature. Systematic errors can only be identified via comparison with what should and does exist. Since DVB and ATSC SI tables support Cyclic Redundancy Checking, random errors can be identified via SI monitoring. In the DVB system, this is done via the DVB health checks. It should be possible to do the same for the ATSC-SI. Testers should support monitoring of the DVB-SI or ATSC-SI CRC's.

The screen shots in figure 8, show Hewlett Packard’s HP E6277A displaying the SI structure: along with a NIT table decoded right down to the satellite_delivery_system_descriptor. Also shown is the HP E6277A CRC generated SI health checks screen: these can be viewed in real-time, and used for error event triggering.

Concluding Remarks.

We have compared and contrasted the DVB-SI and ATSC-SI. Differences are entirely due to the world-views of the DVB and ATSC communities: it remains to be seen, whether the ATSC approach does have a speed advantage over DVB. Both systems are complex, and compared to the analog world, represent a vastly different method of accessing a program event.

To show how an EPG could use the SI, we constructed a hypothetical event purchase scenario. This was done to highlight how crucial the SI is to the whole event purchasing process. EPG design is taking many man-years of design effort, the success or failure of a digital TV service may be down to its EPG.

Examples of errors that could occur where discussed. The different structure of the two SI's will impact on faultfinding. It is the author's view that the DVB-SI will be the easier to debug. Why? The DVB-SI tables are better compartmentalized. Specific EPG errors are closely linked to particular DVB-SI tables. Whereas in the ATSC-SI, there are more inter-table dependencies.

Finally, we indicated what features test equipment should have, if they are to be effective at SI debugging. Decode to the lowest level of descriptor really stands out here. Along with pointing out that though useful, CRC's can only indicate random errors, tables decode is required for the analysis of multiplexer equipment faults.

The whole area of EPG design, and its linkage to the construction and debugging of the DVB-SI is still evolving. The potential for errors is vast. And no other area can have as great a positive or negative effect on revenue generation than the EPG. It's an area that will give us all headaches for many years to come.

"This article was first published in DSP Multimedia magazine. Reprints are available from Hewlett Packard. Contact your local sales office for