REMOTE CONTROL SYSTEMS FOR BROADCASTING PRODUCTION EQUIPMENT
Routing switcher type–specific messages

Tech. 3245–E – Supplement 5 October 1992

CONTENTS

European Broadcasting Union
Case Postale 67, CH–1218 Grand–Saconnex (Geneva) Switzerland

Introduction

Document Tech. 3245 describes the specifications of a digital remote–control system for broadcasting production equipment. It defines completely the electrical/mechanical level (level 1), and the supervisory level (level 2), of the interface. The two remaining levels – the system service level (level 3), and the virtual machine level (level 4) – are defined only in terms of function and control message syntax.

Supplement 1 to Tech. 3245 completes the definition of the system service level by detailing the system service messages and, in addition, defines the virtual machine messages which are common to all types of virtual machine – the common messages.

The present Supplement defines the type–specific virtual machine messages which are applicable to routing switchers. Type–specific messages applicable to other categories of equipment are defined in other supplements to Tech. 3245.

In order to implement a complete network, the system designer therefore requires:

Document Tech. 3245	– the general specification
Supplement 1	– system service and common messages

and one or more other supplements appropriate to the category or categories of equipment to be used.

The specification described in this Supplement has been developed jointly with the SMPTE, and is functionally identical to that to be published as an SMPTE Recommended Practice.

Chapter 1
General concepts

This *Chapter* contains a general explanation of some of the concepts used in the formulation of the Routing Switcher Type–Specific message set. It constitutes tutorial information, and is intended to assist in the understanding of the specifications in *Chapter 2* of this document. A working knowledge of the following ESbus topics is assumed:

ESbus system overview
Control Message architecture
Supervisory protocol
Tributary interconnection
Electrical and mechanical characteristics
System Service and Common Messages

Conventions

Acronyms and abbreviations are shown in upper–case characters.

e.g.	Information Field – I/F

Message Keywords and names of Information Fields are shown in upper–case characters

e.g.	CONNECT CROSSPOINT SOURCE POINTER

These command Keywords and Information Field names are used within the text of this document to imply requested action, Information Field identity, and in turn the Information Field contents of the Virtual Machine. To assist in readability of this document, these terms are used in the context of the presentation material.

e.g.	“If this I/F is PRESET, ...”

(“PRESET” in this context refers to a command contained within the Type–specific Message set.)

Terms with special meaning to this or related documents are shown with leading upper–case characters:

e.g.	Virtual Machine Common Messages.

1. Scope of this Dialect

This dialect is intended for a remote–control system to be used as a link between a routing switcher control system (as the controlling device) and its associated routing switcher matrices (as controlled device(s)).

It is not intended for controlling the routing switcher control system itself from other places.

2. Multidimensional Information Fields

The controlled elements in a routing switcher are the crosspoints. The crosspoints are arranged in a multidimensional way, i.e. to identify an individual crosspoint, it is usually necessary to specify the following characteristics of its location:

- its row (1st dimension);
- its column (2nd dimension);
- its level (3rd dimension);
- its matrix (4th dimension) – applies only to a switcher consisting of several matrices.

In ESbus dialects all kinds of status data are maintained in Information Fields (I/Fs). Each type of information has its corresponding I/F associated with a unique I/F Name.

In routing switcher systems nearly all status data are related to crosspoints. Due to the fact that crosspoints are arranged in a multidimensional way, and that each crosspoint is the carrier of status data, the I/Fs describing routing switcher data must also be multidimensional. The particular item of information belonging to one crosspoint is just an element of the whole Information Field of a certain type.

This requires additional descriptors which point to the "location" of this element within the field, i.e. to the row , column, level, and matrix.

When such an element of an I/F is tallied, these descriptors are simply carried as parameters in the format.

When such an element is accessed, however, a different mechanism is required due to the fact that the Common commands which access I/Fs (e.g. READ, UPDATE, CYCLE) allow only for specifying the I/F name; no additional descriptor information is permitted in the format.

Such additional information must therefore be transmitted in advance by presetting one or more "pointers", which predefine the parameters necessary for multidimensional access.

The pointers themselves also reside in Information Fields of their own and thus can be PRESET as any other presettable I/F. This gives the advantage that the pointer information need only be transmitted when it really changes, not in advance of every I/F access.

The names of the relevant pointers are:

- MATRIX POINTER,
- LEVEL POINTER,
- SOURCE POINTER,
- DESTINATION POINTER.

Note: In order to PRESET a multidimensional I/F no pointers are required, since the whole format of an I/F appears within the PRESET command, including the descriptors.

Due to special requirements of routing switchers, one of these dimensions may not be specified, e.g. the I/F DESTINATIONS–TO–SOURCE does not use the DESTINATION POINTER, because it shows all connections between a specified source and any destination in the form of a list.

3. Wildcard Characters

In order to facilitate access to a whole array (row, column, level, matrix) of one type of a multidimensional status information, a "wildcard" character is introduced (FFh or FFFFh).

A pointer, preset to the wildcard, indicates (when an I/F access requiring this pointer is made), that the information corresponding to the full available range of the pointer is desired.

For example, if the LEVEL POINTER is preset to FFh, a subsequent READ of the I/F CROSSPOINT STATUS will result in a multiple I/F RESPONSE message (either in many single messages or in one message using the BEGIN/END construct or in any combination) tallying the crosspoint status of all existing levels.

A second application of the wildcard concerns certain commands (e.g. CONNECT CROSSPOINT), where it can be used as a normal parameter with the same effect.

The description of the commands and information fields in *Chapter 2* contains detailed instructions as to when and how wildcards can be used.

4. Procedures and Events

As with all Dialects the complete Routing Switcher Dialect consists, by definition, of both Common Messages and the Routing Switcher Type–specific Dialect described in this document.

The elements of the Common Messages prove very useful for switcher applications, and are therefore recommended.

The concept of Procedures, provided by the Common Messages, can be used to predefine a lengthy set of commands (by the DEFINE PROCEDURE command) pending the arrival of the EXECUTE PROCEDURE command, which causes the entire Procedure to be performed.

The concept of Events, provided by the Common Messages, can be used to predefine a command (by the DEFINE EVENT command) to be executed at a certain point in time. As the time scale, usually the Timeline, is used, an individual software clock running in each controlled Virtual Machine is preset by the bus controller (by the System Service command REQUEST TIME TRANSMISSION).

Both concepts can easily be combined by first defining a procedure, then defining an Event with EXECUTE PROCEDURE as the command to be carried out on the Timeline.

Details about these facilities are described in the documents about System Service and Common Messages mentioned above.

5. Sample Command Sequences

The following message sequences show the application of the dialect (including the Common Messages). The commands are also shown encoded into their hex codes.

The PRESET commands signed with a "*" sign may be omitted if the corresponding I/F already has been preset by a preceding PRESET command.

5.1. Disconnecting all crosspoints in level #10 of matrix #2

<DISCONNECT CROSSPOINT>	<43>
<MATRIX = 2>	<02>
<LEVEL = 10>	<0A>
<SOURCE = wildcard>	<FFFF>
<DESTINATION = wildcard>	<FFFF>

5.2. Requesting the status of all crosspoints in level #2 of matrix #1

<PRESET>	*	<60>
<MATRIX POINTER>		<41>
<MATRIX = 1>		<01>
<PRESET>	*	<60>
<LEVEL POINTER>		<42>
<LEVEL = 2>		<02>
<PRESET>	*	<60>
<SOURCE POINTER>		<43>
<SOURCE = wildcard>		<FFFF>
<READ>		<22>
<DESTINATIONS–TO–SOURCE>		<46>

The response from the controlled Virtual Machine may be:

```
<I/F ITEM RESPONSE>                                                     <23>
        <BEGIN>                                                         <01>
                <DESTINATIONS–TO–SOURCE>                                <46>
                        <MATRIX = 1>                                    <01>
                        <LEVEL = 2>                                     <02>
                        <SOURCE = 0>                                    <0000>
                                <PARAMETER COUNT = 1> (one connection)  <0001>
                                <DESTINATION = 3>                       <0003>
                <DESTINATIONS–TO–SOURCE>                                <46>
                        <MATRIX = 1>                                    <01>
                        <LEVEL = 2>                                     <02>
                        <SOURCE = 1>                                    <0001>
                                <PARAMETER COUNT = 0> (no connection)   <0000>
                <DESTINATIONS–TO–SOURCE>                                <46>
                        <MATRIX = 1>                                    <01>
                        <LEVEL = 2>                                     <02>
                        <SOURCE = 2>                                    <0002>
                                <PARAMETER COUNT = 4> (multiple connections) <0004>
                                <DESTINATION = 2>                       <0002>
                                <DESTINATION = 4>                       <0004>
                                <DESTINATION = 9>                       <0009>
                                <DESTINATION = 17>                      <0011>
                <DESTINATIONS–TO–SOURCE>                                <46>
                        <MATRIX = 1>                                    <01>
                        <LEVEL = 2>                                     <02>
                        <SOURCE = 31>                                   <001F>
                                <PARAMETER COUNT = 1>                   <0001>
                                <DESTINATION = 1>                       <0001>
        <END>                                                            <02>
```

5.3. Instructing the controlled Virtual Machine to tally any changes in the crosspoint status of matrix #3

```
<PRESET>                                        *                       <60>
        <MATRIX POINTER>                                                <41>
                <MATRIX = 3>                                            <03>
PRESET>                                         *                       <60>
        <LEVEL POINTER>                                                 <42>
                <LEVEL = wildcard>                                      <FF>
<PRESET>                                        *                       <60>
        <DESTINATION POINTER>                                           <44>
                <DESTINATION = wildcard>                                <FFFF>
<UPDATE>                                                                <3F07>
        <SOURCES–TO–DESTINATION>                                        <47>
```

On a change, the response from the controlled Virtual machine may be:

```
<I/F ITEM RESPONSE>                                                     <23>
        <SOURCES–TO–DESTINATION>                                        <47>
                <MATRIX = 3>                                            <03>
                <LEVEL = 2>                                             <02>
                <DESTINATION = 9>                                       <0009>
                        <PARAMETER COUNT = 1>                           <0001>
                        <SOURCE = 11>                                   <000B>
```

5.4. Defining a procedure that establishes a default configuration of connections

```
<DEFINE PROCEDURE>                                       <3F0B>
        <NAME = 1>                                       <01>
        <BYTE COUNT = XX>                                <00XX>
                <CONNECT CROSSPOINT>                     <42>
                <MATRIX = 1>                             <01>
                <LEVEL = 1>                              <01>
                <SOURCE = 0>                             <0000>
                <DESTINATION = 12>                       <000C>
                <CONNECT CROSSPOINT>                     <42>
                <MATRIX = 1>                             <01>
                <LEVEL = 1>                              <01>
                <SOURCE = 1>                             <0001>
                <DESTINATION = 10>                       <000A>
                <CONNECT CROSSPOINT>                     <42>
                <MATRIX = 1>                             <01>
                <LEVEL = 1>                              <01>
                <SOURCE = 2>                             <0002>
                <DESTINATION = 2>                        <0002>
                <CONNECT CROSSPOINT>                     <42>
                <MATRIX = 1>                             <01>
                <LEVEL = 1>                              <01>
                <SOURCE = 3>                             <0003>
                <DESTINATION = 7>                        <0007>
                 ...
```

Once defined, the procedure may be carried out as often as desirable simply by commanding:

```
<EXECUTE PROCEDURE>                                      <26>
        <NAME = 1>                                       <01>
```

5.5. Setting a crosspoint at 09:00 on the timeline

```
<DEFINE EVENT>                                           <27>
        <NAME = 3>                                       <03>
        <I/F NAME OF TRIGGER SOURCE = TIMELINE>          <24>
        <TRIGGER VALUE = 09:00:00:00>                    <09000000>
                <CONNECT CROSSPOINT>                     <42>
                <MATRIX = 1>                             <01>
                <LEVEL = 1>                              <01>
                <SOURCE = 3>                             <0003>
                <DESTINATION = 7>                        <0007>
```

Chapter 2
Routing Switcher Type–specific Messages
(virtual machine type is 05h)

General notes

1. *All parameters described below as “1–byte number” or “2–byte number” are binary coded unsigned numbers.*
2. *Parameters which can be used with Wildcard Characters are indicated by “FFh = all” or “FFFFh = all”.*
3. *In all cases, the temporal order of EVENTS must be preserved. Mutually exclusive commands actuated by the EVENT construct, that are placed on the EVENT cue at the same trigger point, will cause both events to cancel.*

1. Numerical Index of Keywords, Information Field Names, and Mnemonics

Hex	Message Keyword	(mnemonic)	Hex	Information Field Name	(mnemonic)
40h	not used		40h	not used	
41h	not used		41h	MATRIX POINTER	MPOI
42h	CONNECT CROSSPOINT	CONC	42h	LEVEL POINTER	LPOI
43h	DISCONNECT CROSSPOINT	DISC	43h	SOURCE POINTER	SPOI
44h	not used		44h	DESTINATION POINTER	DPOI
45h	not used		45h	not used	
46h	SPECIFIC MUTE		46h	DESTINATIONS–TO–SOURCE	DTOS
47h	not used		47h	SOURCES–TO–DESTINATION	STOD
48h	TEST CROSSPOINTS	TESC	48h	CROSSPOINT STATUS	CSTA
49h	not used		49h	SOURCE SIGNAL STATUS	SSTA
4Ah	not used		4Ah	LEVEL CONFIGURATION	LECO
4Bh	not used		4Bh	LEVEL BLOCK STATUS	LEBS
4Ch	not used		4Ch	not used	
4Dh	not used		4Dh	not used	
4Eh	not used		4Eh	SOURCE NAME	SNAM
4Fh	not used		4Fh	DESTINATION NAME	DNAM
. . .			. . .		
60h	PRESET	PRST	60h	not used	

2. Keywords

40h not used

41h not used

42h CONNECT CROSSPOINT

causes the crosspoint between the specified source (row) and the specified destination (column) in the specified level of the specified matrix to be connected.

Format:	<CONNECT CROSSPOINT>		
	<MATRIX>	1–byte number	(FFh = all)
	<LEVEL>	1–byte number	(FFh = all)
	<SOURCE>	2–byte number	
	<DESTINATION>	2–byte number	(FFFFh = all)

Notes: 1. If the matrix is addressed with the wildcard FFh, the crosspoints between specified source and destination in the specified level of all available matrices are connected.

2. If the level is addressed with the wildcard FFh, the crosspoints between specified source and destination in all available levels of the specified matrix are connected.

3. If the destination is addressed with the wildcard FFFFh, the crosspoints between the specified source and all available destinations in the specified level of the specified matrix are connected.

4. More than one parameter may use the wildcard at the same time.

43h DISCONNECT CROSSPOINT

causes the crosspoint between the specified source (row) and the specified destination (column) in the specified level of the specified matrix to be disconnected.

Format:	<DISCONNECT CROSSPOINT>		
	<MATRIX>	1–byte number	(FFh = all)
	<LEVEL>	1–byte number	(FFh = all)
	<SOURCE>	2–byte number	(FFFFh = all)
	<DESTINATION>	2–byte number	(FFFFh = all)

Notes: 1. If the matrix is addressed with the wildcard FFh, the crosspoints between specified source and destination in the specified level of all available matrices are disconnected.

2. If the level is addressed with the wildcard FFh, the crosspoints between specified source and destination in all available levels of the specified matrix are disconnected.

3. If the source is addressed with the wildcard FFFFh, a crosspoint that might be set between a source and the specified destination in the specified level of the specified matrix is disconnected.

4. If the destination is addressed with the wildcard FFFFh, all crosspoints that might be set between the specified source and all available destinations in the specified level of the specified matrix are disconnected.

5. More than one parameter may use the wildcard at the same time, e.g. if both source and destination are addressed with the wildcard FFFFh, all crosspoints in the specified level of the specified matrix are disconnected.

44h not used

45h not used

46h SPECIFIC MUTE

directs the controlled Virtual Machine to switch off all responses previously initiated by a CYCLE or UPDATE command for the specified Information Field.

Format:	<SPECIFIC MUTE>
	<I/F NAME>

Notes: 1. This command supplements the MUTE command of the Common Message set, which is a general mute for all I/Fs.

2. *This command requires the same pre–definitions of the pointers as the UPDATE or CYCLE command it is intended to cancel.*

47h not used

48h TEST CROSSPOINT
causes the crosspoint between the specified source (row) and the specified destination (column) in the specified level of the specified matrix to be tested; the test result may be interrogated by READing the Information Field CROSSPOINT STATUS.

Format:	<TEST CROSSPOINT>		
	<MATRIX >	1–byte number	(FFh = all)
	<LEVEL >	1–byte number	(FFh = all)
	<SOURCE>	2–byte number	(FFFFh = all)
	<DESTINATION>	2–byte number	(FFFFh = all)

Notes: 1. *If the matrix is addressed with the wildcard FFh, the crosspoints between specified source and destination in the specified level of all matrices are tested.*

2. *If the level is addressed with the wildcard FFh, the crosspoints between specified source and destination in all levels of the specified matrix are tested.*

3. *If the source is addressed with the wildcard FFFFh, the crosspoints between all sources and the specified destination in the specified level of the specified matrix are tested.*

4. *If the destination is addressed with the wildcard FFFFh, the crosspoints between the specified source and all destinations in the specified level of the specified matrix are tested.*

5. *More than one parameter may use the wildcard at the same time, e.g. if both source and destination are addressed with the wildcard >FFFFh, all crosspoints in the specified level of the specified matrix are tested.*

50h
to } not used
5Fh

60h PRESET
presets the named Information Field to the given value.

Format: <PRESET>
<PERMITTED INFORMATION FIELD NAME>
<VALUE> format and coding defined by the I/F NAME
(see Section 3: Information Fields)

Permitted Information Field names for Routing Switchers are:

MATRIX POINTER
LEVEL POINTER
SOURCE POINTER
DESTINATION POINTER
SOURCE NAME
DESTINATION NAME

3. Information Fields

Notes: 1. The items of the Information Field are accessed by the Common messages:

READ, UPDATE, CYCLE or SIMULTANEOUS READ

These commands use the format:

<KEYWORD><PARAMETER NAME>

where the PARAMETER NAME uses the Information Field Name specified below.

Though several Parameter Names may be grouped together by means of a BEGIN/END construct, the command does not allow for carrying additional parameters in order to specify which item of a multidimensional Information Field shall be accessed.

Such additional information must therefore be transmitted in advance by presetting one or more pointers e.g. MATRIX POINTER, LEVEL POINTER, SOURCE POINTER, DESTINATION POINTER.

2. The items of the Information Field are tallied by the Common Messages:

I/F RESPONSE or SIMULTANEOUS READ RESPONSE

These commands use the format:

<KEYWORD><PARAMETER NAME><PARAMETER VALUE>

where the PARAMETER VALUE carries the Information Field contents specified below.

The parameters that subdefine the individual item of a multidimensional Information Field, e.g. MATRIX POINTER, LEVEL POINTER, etc., are carried within the Parameter Values. Therefore the content of the Pointers is of no effect in the instant when an Information Field is tallied.

Several names/values may be grouped together by means of a BEGIN/END construct.

3. Multidimensional Information Fields and their corresponding Pointers are individually noted in the description below.

40h not used

41h MATRIX POINTER
defines a pointer to a matrix.

Format:	<MATRIX POINTER>		
	<MATRIX>	1–byte number	(FFh = all matrices)

Note: *If this field is PRESET to the wildcard FFh, a READ command referring to this pointer will result in multiple I/F RESPONSES for all available matrices.*

42h LEVEL POINTER
defines a pointer to a level of the matrix specified in the MATRIX POINTER I/F.

Format:	<LEVEL POINTER>		
	<LEVEL >	1–byte number	(FFh = all levels)

Note: *If this field is PRESET to the wildcard FFh, a READ command referring to this pointer will result in multiple I/F RESPONSES for all available levels.*

43h SOURCE POINTER
defines a pointer to a source (row) in the level (specified in the LEVEL POINTER I/F) of the matrix (specified in the MATRIX POINTER I/F).

Format:	<SOURCE POINTER>		
	<SOURCE>	2–byte number	(FFFFh = all sources)

Note: If this field is PRESET to the wildcard FFFFh, a READ command referring to this pointer will result in multiple I/F RESPONSES for all available sources.

44h DESTINATION POINTER
defines a pointer to a destination (column) in the level (specified in the LEVEL POINTER I/F) of the matrix (specified in the MATRIX POINTER I/F).

Format:	<DESTINATION POINTER>		
	<DESTINATION>	2–byte number	(FFFFh = all destinations)

Note: If this field is PRESET to the wildcard FFFFh, a READ command referring to this pointer will result in multiple I/F RESPONSES for all available destinations.

45h not used

46h DESTINATIONS–TO–SOURCE
indicates all destinations (columns) in a specified level of a specified matrix that are currently connected to a specified source (row).

Format:	<DESTINATIONS–TO–SOURCE>	
	<MATRIX >	1–byte number
	<LEVEL >	1–byte number
	<SOURCE>	2–byte number specifying the source
	<PARAMETER COUNT>	2–byte number specifying the number *n* of parameters following
	<DESTINATION 1>	2–byte number specifying the 1st destination connected to the source
	...	
	<DESTINATION *n*>	2–byte number specifying the *n*th destination connected to the source

Notes: 1. Parameter Count = 0 means: no connection.

2. Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and SOURCE POINTER to be PRESET in advance.

47h SOURCES–TO–DESTINATION
indicates all sources (rows) in a specified level of a specified matrix that are currently connected to a specified destination (column).

Format:	<SOURCES–TO–DESTINATION>	
	<MATRIX>	1–byte number
	<LEVEL>	1–byte number
	<DESTINATION>	2–byte number specifying the destination
	<PARAMETER COUNT>	2–byte number specifying the number *n* of parameters following
	<SOURCE 1>	2–byte number specifying the 1st source connected to the destination
	...	
	<SOURCE *n*>	2–byte number specifying the *n*th source connected to the destination

Notes: 1. Parameter Count = 0 means: no connection.

2. Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and DESTINATION POINTER to be PRESET in advance.

48h CROSSPOINT STATUS
gives a list of the bad crosspoints corresponding to a specified destination (column) in a specified level of a specified matrix.

Format:	<CROSSPOINT STATUS>	
	<MATRIX>	1–byte number
	<LEVEL>	1–byte number
	<DESTINATION>	2–byte number
	<PARAMETER COUNT>	2–byte number specifying the number *n* of parameters following
	<SOURCE 1>	2–byte number specifying the 1st of a list of bad crosspoints
	...	
	<SOURCE *n*>	2–byte number specifying the *n*th and last one of a list of bad crosspoints

Notes: 1. Parameter Count = 0 means: no bad crosspoints.

2. Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and DESTINATION POINTER to be PRESET in advance.

49h SOURCE SIGNAL STATUS
indicates the signal quality of a specified source in a specified level of a specified matrix.

Format:	<SOURCE SIGNAL STATUS>	
	<MATRIX>	1–byte number
	<LEVEL>	1–byte number
	<SOURCE>	2–byte number
	<CODE>	1–byte special binary code: 00h = good else = bad (details may be reported using user–defined codes)

Note: Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and SOURCE POINTER to be PRESET in advance.

4Ah LEVEL CONFIGURATION
describes the start–up configuration in the specified level of the specified matrix by first defining the rectangular limits of the level and then detailing crosspoint blocks (typically card–related) which do not exist.

Format:	<LEVEL CONFIGURATION>		
	<MATRIX>	1–byte number	
	<LEVEL>	1–byte number	
	<FIRST SOURCE>	2–byte number	rectangular limits of the level
	<LAST SOURCE>	2–byte number	
	<FIRST DESTINATION>	2–byte number	
	<LAST DESTINATION>	2–byte number	
	<PARAMETER GROUP COUNT>	1–byte number specifying the number *n* of parameter groups following	
	<SOURCE# OF BLOCK BEG>	2–byte number	data of 1st non–existent block
	<SOURCE# OF BLOCK END>	2–byte number	
	<DEST# OF BLOCK BEG>	2–byte number	
	<DEST# OF BLOCK END>	2–byte number)	
	...		
	<SOURCE# OF BLOCK BEG>	2–byte number	data of *n*th non–existent block
	<SOURCE# OF BLOCK END>	2–byte number	
	<DEST# OF BLOCK BEG>	2–byte number	
	<DEST# OF BLOCK END>	2–byte number	

Note: Accessing this I/F requires the I/Fs MATRIX POINTER and LEVEL POINTER to be PRESET in advance.

4Bh LEVEL BLOCK STATUS
reports blocks of crosspoints (typically card–related) in the specified level of the specified matrix that are detected by the controlled Virtual Machine as missing relative to the start–up configuration.

Format:	<LEVEL BLOCK STATUS>		
	<MATRIX>	1–byte number	
	<LEVEL>	1–byte number	
	<PARAMETER GROUP COUNT>	1–byte number specifying the number *n* of parameter groups following	
	<SOURCE# OF BLOCK BEG>	2–byte number	data of 1st faulty block
	<SOURCE# OF BLOCK END>	2–byte number	
	<DEST# OF BLOCK BEG>	2–byte number	
	<DEST# OF BLOCK END>	2–byte number	
	...		
	<SOURCE# OF BLOCK BEG>	2–byte number	data of *n*th faulty block
	<SOURCE# OF BLOCK END>	2–byte number	
	<DEST# OF BLOCK BEG>	2–byte number	
	<DEST# OF BLOCK END>	2–byte number	

Notes: 1. Parameter Group Count = 0 means: no faulty blocks

2. Accessing this I/F requires the I/Fs MATRIX POINTER and LEVEL POINTER to be PRESET in advance.

4Ch not used

4Dh not used

4Eh SOURCE NAME
contains the name of the specified source in the specified level of the specified matrix.

Format:	<SOURCE NAME>	
	<MATRIX>	1–byte number
	<LEVEL>	1–byte number
	<SOURCE>	2–byte number
	<CHARACTER COUNT>	2–byte number specifying the number *n* of characters following
	<CHARACTER 1>	1–byte ASCII code; 1st character
	...	
	<CHARACTER *n*>	1–byte ASCII code; *n*th character

Notes: 1. This I/F can be PRESET by the controlling Virtual Machine and is to be used only for displaying the names at the switcher, where applicable.

2. Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and SOURCE POINTER to be PRESET in advance.

4Fh DESTINATION NAME
contains the name of the specified destination in the specified level of the specified matrix.

Format:	<DESTINATION NAME>	
	<MATRIX>	1–byte number
	<LEVEL>	1–byte number
	<DESTINATION>	2–byte number
	<CHARACTER COUNT>	2–byte number specifying the number *n* of characters following
	<CHARACTER 1>	1–byte ASCII code; 1st character
	...	
	<CHARACTER *n*>	1–byte ASCII code; *n*th character

Notes: 1. This I/F can be PRESET by the controlling Virtual Machine and is to be used only for displaying the names at the switcher, where applicable.

2. Accessing this I/F requires the I/Fs MATRIX POINTER, LEVEL POINTER, and DESTINATION POINTER to be PRESET in advance.

Editeur responsable : G.T. Waters, Ancienne Route 17A, CH–1218 Grand–Saconnex/Genève, Suisse.