Division	Pages	Issue Number
0-UPD	1	8.03
1	8	8.03
2	14	8.03
3	4	8.00
4	1	8.02
4.1	2	8.00
4.2	34	8.00
4.3	7	8.02
5	1	8.02
5.1	12	8.00
5.2	31	8.00
5.3	4	8.00
6	20	8.00
7	1	8.00
ABBREVS	2	8.00
INDEX	5	8.03

Lucent Technologies
Bell Labs Innovations

5ESS®-2000 Switch
Recent Change Operations
Systems Interface Specification

5E11 and Later Software Releases

235-900-304
Issue 8.03
June 1999

Notice

Every effort was made to ensure that the information in this IP was complete and accurate at the time of publication. However, information is subject to change.

This IP describes certain hardware, software, features, and capabilities of Lucent Technologies products. This IP is for information purposes; therefore, caution is advised that this IP may differ from any configuration currently installed.

Mandatory Customer Information

Interference Information: Part 15 of FCC Rules – Refer to the *5ESS*®-2000 Switch Product Specification IP.

Trademarks

5ESS is a registered trademark of Lucent Technologies.
UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Warranty

Warranty information applicable to the *5ESS*®-2000 switch may be obtained from the Lucent Technologies Account Management organization. Customer-modified hardware and/or software is not covered by this warranty.

Ordering Information

This IP is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana.

The order number for this IP is 235-900-304. To order, call:

1–888–LUCENT8 (1–888–582–3688) or fax to 1–800–566–9568 (From inside the continental United States)

1–317–322–6848 or fax to 1–317–322–6699 (From outside the continental United States).

Support Telephone Numbers

Information Product Support Telephone Number: To report errors or ask nontechnical questions about this or other IPs produced by Lucent Technologies, call 1–888–LTINFO6 (1–888–584–6366).

Technical Support Telephone Numbers: For initial technical assistance, call the North American Regional Technical Assistance Center (NARTAC) at 1–800–225–RTAC (1–800–225–7822). For further assistance, call the Customer Technical Assistance Management (CTAM) Center:

1–800–225–4672 (From inside the continental United States)

1–630–224–4672 (From outside the continental United States).

The CTAM center is staffed 24 hours a day, 7 days a week.

Acknowledgment

Developed by Lucent Technologies Customer Training and Information Products.

How Are We Doing?

Title: ***5ESS®*-2000 Switch Recent Change Operations Systems Interface Specification 5E11 and Later Software Releases**

Identification No.: **235-900-304** Issue No.: **8.03** Date: **June 1999**

Lucent Technologies welcomes your feedback on this Customer Information Product (CIP). Your comments can be of great value in helping us improve our CIPs.

1. Please rate the effectiveness of this CIP in the following areas:

	Excellent	Good	Fair	Poor	Not Applicable
Ease of Use					////////////////////
Clarity					////////////////////
Completeness					////////////////////
Accuracy					////////////////////
Organization					////////////////////
Appearance					////////////////////
Examples					
Illustrations					
Overall Satisifaction					////////////////////

2. Please check the ways you feel we could improve this CIP.

- ❒ Improve the overview/introduction
- ❒ Improve the table of contents
- ❒ Improve the organization
- ❒ Include more figures
- ❒ Add more examples
- ❒ Add more detail
- ❒ Make it more concise/brief
- ❒ Add more step-by-step procedures/tutorials
- ❒ Add more troubleshooting information
- ❒ Make it less technical
- ❒ Add more/better quick reference aids
- ❒ Improve the index

Please provide details for the suggested improvement. _____

_____

3. What did you like most about this CIP?

_____

_____

4. Feel free to write any comments below or on an attached sheet.

_____

_____

_____

_____

If we may contact you concerning your comments, please complete the following:

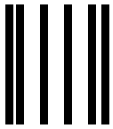
Name: _____ Telephone Number: _____

Company/Organization: _____ Date: _____

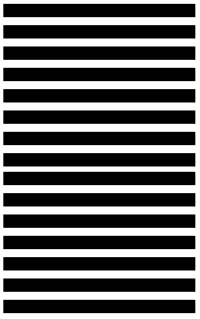
Address: _____

When you have completed this form, please fold, tape and return to address on back or Fax to: 336 727-3043.

---------- Do Not Cut — Fold Here And Tape ----------



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Lucent Technologies
Bell Labs Innovations

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1999 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, NC 27199-2029

How Are We Doing?

Title: ***5ESS®*-2000 Switch Recent Change Operations Systems Interface Specification 5E11 and Later Software Releases**

Identification No.: **235-900-304** Issue No.: **8.03** Date: **June 1999**

Lucent Technologies welcomes your feedback on this Customer Information Product (CIP). Your comments can be of great value in helping us improve our CIPs.

1. Please rate the effectiveness of this CIP in the following areas:

	Excellent	Good	Fair	Poor	Not Applicable
Ease of Use					////////////////////
Clarity					////////////////////
Completeness					////////////////////
Accuracy					////////////////////
Organization					////////////////////
Appearance					////////////////////
Examples					
Illustrations					
Overall Satisifaction					////////////////////

2. Please check the ways you feel we could improve this CIP.

- ❒ Improve the overview/introduction
- ❒ Improve the table of contents
- ❒ Improve the organization
- ❒ Include more figures
- ❒ Add more examples
- ❒ Add more detail
- ❒ Make it more concise/brief
- ❒ Add more step-by-step procedures/tutorials
- ❒ Add more troubleshooting information
- ❒ Make it less technical
- ❒ Add more/better quick reference aids
- ❒ Improve the index

Please provide details for the suggested improvement. _____

_____

3. What did you like most about this CIP?

_____

_____

4. Feel free to write any comments below or on an attached sheet.

_____

_____

_____

_____

If we may contact you concerning your comments, please complete the following:

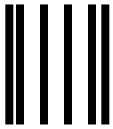
Name: _____ Telephone Number: _____

Company/Organization: _____ Date: _____

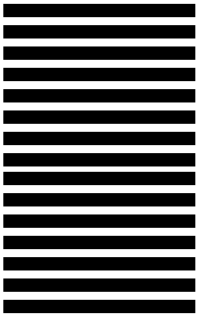
Address: _____

When you have completed this form, please fold, tape and return to address on back or Fax to: 336 727-3043.

---------- Do Not Cut — Fold Here And Tape ----------



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



Lucent Technologies
Bell Labs Innovations

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1999 GREENSBORO, NC

POSTAGE WILL BE PAID BY ADDRESSEE

DOCUMENTATION SERVICES
2400 Reynolda Road
Winston-Salem, NC 27199-2029

PREFACE

On May 15, 1986, the Federal Communications Commission (FCC) ended Phase 1 of Computer Inquiry III by conditionally removing structural separation requirements for the competitive provisioning of enhanced services. In an order released on June 16, 1986, the main conditions imposed are the long-term implementation of an Open Network Architecture (ONA) plan and, in the interim, the availability of Comparably Efficient Interconnection (CEI) arrangements for individual enhanced services. Lucent Technologies expects to evolve its products to meet the needs of its customers by providing standard interfaces to meet ONA and CEI requirements.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 PURPOSE

This document defines the Recent Change (RC) interface requirements between the *5ESS*®-2000 switch and a remote processor that provides the RC-related capabilities that are also defined in this document. Those who intend to design, manufacture, or supply systems and equipment compatible with a *5ESS*-2000 switch and a supported software release may use this interface specification as a guide.

This interface specification is expected to change as requirements and standards evolve. Therefore, Lucent Technologies reserves the right to change or delete any portions of the document, or to add information in future issues.

NOTE: LUCENT TECHNOLOGIES EXPRESSLY DISCLAIMS ANY WARRANTY AS TO THE ACCURACY OR COMPLETENESS OF INFORMATION CONTAINED IN THIS DOCUMENT, AND ASSUMES NO RESPONSIBILITY FOR THE USE OF THE INFORMATION BY OTHERS.

1.2 UPDATE INFORMATION

1.2.1 NEW IN THIS ISSUE

Section 2.4.3.2, “Connection Establishment Service,” and Section 2.4.4, “Security,” have been revised for clarity.

Where technical content has been changed, vertical bars in the outside margin mark the affected pages.

1.2.2 LUCENT TECHNOLOGIES

As a result of the AT&T divestiture, the AT&T Network Systems division became Lucent Technologies, a separate and independent corporation. The *5ESS*-2000 switch, and many other network and transmission products became products of Lucent Technologies. The marketing, sales, engineering, delivery, installation, support, and future development of these products are now provided by Lucent Technologies.

Therefore, the corporate name and logo on this document's cover, spine, and title page are changed from the AT&T brand to the Lucent Technologies brand. The AT&T name is being removed from the content (where appropriate), as well as from the 9-digit document order number.

Not all pages of this document are being reissued to make these changes; instead, the pages will be reissued over time, as technical and other changes are required. Customers on standing order for this document may see references to the AT&T name on previous-issue pages. Customers receiving new orders for this document will see the references changed to the Lucent Technologies name throughout the document as appropriate. Exceptions may exist in software-influenced elements such as input/output messages, master control center screens, and recent change/verify screens. These elements will not be changed in this document until such time as they are changed in the software code.

Document updates will not be made specifically to remove historical references to AT&T, especially in cases where the Network Systems division of AT&T, now Lucent Technologies, provided the product or service in question.

1.2.3 SUPPORTED SOFTWARE RELEASES

In accordance with the *5ESS*-2000 Switch Software Support Plan, the 5E10 software release was rated Discontinued Availability (DA) as of November 13, 1998. The information supporting 5E10 and earlier is being removed over time, instead of concurrently, from all documentation.

If you are supporting offices that use a software release prior to 5E11 and have a need for the information that is being removed, retain the associated pages as they are removed from the paper documents, or retain the earlier copy of the CD-ROM.

1.2.4 TERMINOLOGY

Effective with the 5E9(1) software release, the name of the *5ESS* switch was changed to *5ESS*-2000 switch; therefore, the name *5ESS* switch is no longer valid. The name revision will be accomplished over time as other technical changes are required. In the interim, assume that any reference to the *5ESS* switch is also applicable to the *5ESS*-2000 switch. Note that this name change may not have been carried forward into software-influenced elements such as input/output messages, master control center screens, and recent change/verify screens.

National ISDN is an evolving platform in which new features will continue to be introduced for new revenue opportunities, for improved operational efficiencies, and for support of specific applications. NI 1, NI 2, and NI 3 represent specific features as documented in Bellcore SRs 1937, 2120, and 2457. The industry is migrating to an additional terminology to more specifically denote the availability of National ISDN features: NI 95, NI 96, etc. A feature is included in a specific version (such as, NI 96) if the switch vendors can offer it by the first quarter of the year.

As a result of the World Telecommunications Standardization Conference held March 1-12, 1993, the International Telegraph and Telephone Consultative Committee (CCITT), no longer exists as an organization under the International Telecommunication Union (ITU). According to the ITU, the CCITT is now referred to as the International Telecommunication Union - Telecommunication Standardization Sector (ITU-TS).

For new and revised recommendations issued by the ITU-TS, the term “CCITT Recommendation X.xxx” will be replaced by the “ITU-T Recommendation X.xxx” designation. For a transition period from 1993 to 1997, if the recommendation had a previous CCITT designation, the new name included “(formerly CCITT Recommendation X.xxx)”. Names of existing CCITT recommendations will not change unless revised.

1.3 ORGANIZATION

This document contains the following sections:

- Section 2, "LOWER LAYERS"
- Section 3, "SESSION LAYER"
- Section 4, "APPLICATION LAYER—RC & OFR CAPABILITIES"
- Section 5, "MESSAGE STRUCTURE DEFINITIONS"
- Section 6, "MESSAGE"
- Section 7, "REFERENCES"

1.4 USER FEEDBACK

The producers of this specification are constantly striving to improve quality and usability. Please use the enclosed comment form (available in paper documents only) to give us your comments and to advise us of any errors. Alternatively, you can write to us at the following address:

Lucent Technologies
Documentation Services
2400 Reynolda Road
Winston-Salem, NC 27106

Please include the issue number and/or date of the specification, your complete mailing address, and telephone number. We will answer all correspondence within 30 days, informing you of the disposition of your comments.

You may also call our Documentation HOTLINE if you need an immediate answer to a documentation question. This HOTLINE is intended to enhance the comment process. The HOTLINE number is **1-888-LTINFO6** (1-888-584-6366) and it is available from 7:30 a.m. to 6:00 p.m. eastern standard time. Outside of those hours, the line is served by an answering machine. You can leave a message on the answering machine and someone will return your call the following business day.

Document users who have access to electronic mail facilities may send comments via electronic mail to **hotline5@wrddo.lucent.com**. When sending comments via electronic mail, please make sure that the document title, document number, document issue number, and/or document date are included in the mail along with your name, phone number, and address.

The comment form and address are for technical comments only. Questions concerning distribution should be handled through your local Lucent Technologies Account Representative or sent directly to our Customer Information Center.

1.5 DISTRIBUTION

This specification is distributed by the Lucent Technologies Customer Information Center in Indianapolis, Indiana. Most operating companies should place orders through their documentation coordinator. Some companies may allow customers to order directly from the Customer Information Center, however, the majority do not. Companies that use documentation coordinators to manage their orders receive a significant discount. If you do not know how to reach your company's documentation coordinator, you may call **1-888-582-3688** to obtain the coordinator's name and telephone number.

Employees of Lucent Technologies and of companies not represented by a documentation coordinator can order the documentation for the *5ESS*-2000 switch directly from the Customer Information Center. Proper billing information must be provided. These orders may be mailed to:

Customer Information Center
Order Entry
2855 N. Franklin Road
Indianapolis, IN 46219

Orders may also be called in on **1-888-LUCENT-8 (1-888-582-3688)** from within the continental United States and on **+1-317-322-6416** from outside the continental United States. Orders may be faxed in on **1-317-322-6484**.

1.6 TECHNICAL ASSISTANCE

Technical assistance for the *5ESS*-2000 switch can be obtained by calling the North American Regional Technical Assistance Center (NARTAC) at **1-800-225-RTAC**. This telephone number is monitored 24 hours a day, 7 days a week. During regular business hours, your call will be answered by your local NARTAC. Outside of normal business hours, all calls will be answered at a centralized technical assistance center where service-affecting problems will be dispatched immediately to your local NARTAC. All other problems will be referred to your local NARTAC on the next regular business day.

1.7 CAPABILITIES

The protocol messages specified in this specification support the following capabilities:

- RC Capabilities - The RC capabilities provide access to query and update the *5ESS*-2000 switch office-dependent data base, for example, line, trunk and equipment data.
- Office Records (OFR) Capabilities - The OFR capabilities provide access to print various office records forms, and to change/list OFR operating and scheduling parameters.
- Automatic Customer Station Rearrangement (ACSR) RC Capabilities - The ACSR RC capabilities provide access to obtain information on daily ACSR activities.
- Retrieve ASCII files from the *5ESS*-2000 switch.

1.8 CONFIGURATIONS

From the remote processor perspective, Figure 1-1 shows the remote processor/network configurations that are supported by the network. Basically,

- A single remote processor is permitted to be connected to several *5ESS*-2000 switches.
- For a single remote processor, the maximum number of data links via input/output port (IOP) to a single *5ESS*-2000 switch is engineerable and limited by traffic, performance, and availability of main memory.

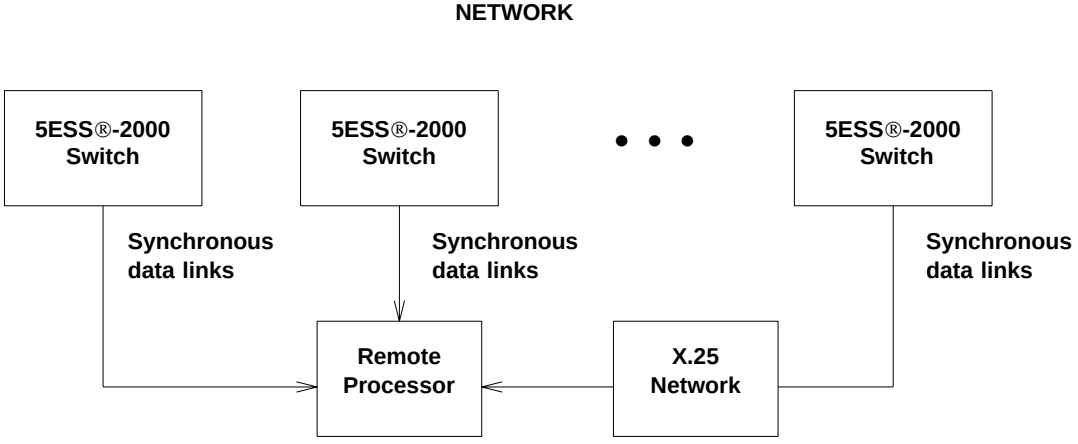


Figure 1-1 — Remote Processor/Network Configurations - Remote Processor Perspective

Figure 1-2 shows the various interfaces between the network (*5ESS*-2000 switch), a remote processor, the attendant terminal, and the end-user. The interfaces between the user and the remote processor, and between the attendant console and the remote processor are not subjects of this interface specification. This document is limited to the interface between the network and the remote processor.

1.8.1 CONFIGURATIONS FOR BX.25 PROTOCOLS

The *5ESS*-2000 switch RCOS interface uses BX.25 protocols for the Layer 1 (physical layer), Layer 2 (link layer), Layer 3 (packet layer), and Layer 4 (session layer). These BX.25 protocol messages are specified in the documents "Operations Systems Network Communications Protocol Specification BX.25 Issue 3" and "Operations Systems Network Communications Protocol Specification BX.25 Issue 3 Addendum A" (document numbers PUB.54001 and PUB.54001A).

In this document, "LOWER LAYERS", Section 2, describes the protocol parameters and options for the BX.25 lower three layers that are used in the *5ESS*-2000 switch RCOS interface. "SESSION LAYER", Section 3, describes the protocol parameters and options for the BX.25 session layer that are used in the *5ESS*-2000 switch RCOS interface. "APPLICATION LAYER - RECENT CHANGE AND OFFICE RECORDS CAPABILITIES", Section 4, describes the protocol messages for the application layer that provide the RC and OFR capabilities.

The remote processor connects to the network via the BX.25 interface directly over synchronous data links or indirectly through X.25 packet switch subnetwork, using switched virtual circuits (SVCs) or permanent virtual circuits (PVCs) up to 9.6 kbps, using the current TN75C circuit pack on the *5ESS*-2000 switch. Transfer rates up to 56 kbps can be accomplished using the TN82 circuit pack on the *5ESS*-2000 switch. The network session layer serves up to 7 SVCs or up to 4 PVCs simultaneously over one data link that connects the remote processor with the network. Since the session layer in the *5ESS*-2000 switch RCOS interface does not use recovery option, SVCs and PVCs can coexist over a single data link. The maximum number of simultaneous sessions is 7.

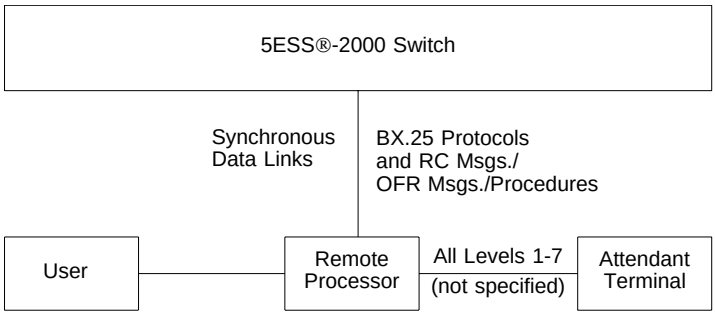


Figure 1-2 — *5ESS*-2000 Switch to Remote Processor, Attendant Terminal and User, BX.25 Protocols

1.8.2 CONFIGURATIONS FOR TCP/IP PROTOCOLS

Beginning with the 5E13 software release, the *5ESS*-2000 switch Administrative Services Module (ASM) supports a Transmission Control Protocol/Internet Protocol (TCP/IP) stack for the RCOS interface. This protocol stack, supplied by the ASM platform vendor, meets the industry standards for TCP/IP and is based on:

- RFC 791, Internet Protocol

- RFC 792, Internet Control Message Protocol
- RFC 793, Transmission Control Protocol
- RFC 1122, Requirements for Internet Hosts—Communications Layers.

1.9 INTERACTIONS AMONG THE LAYERS

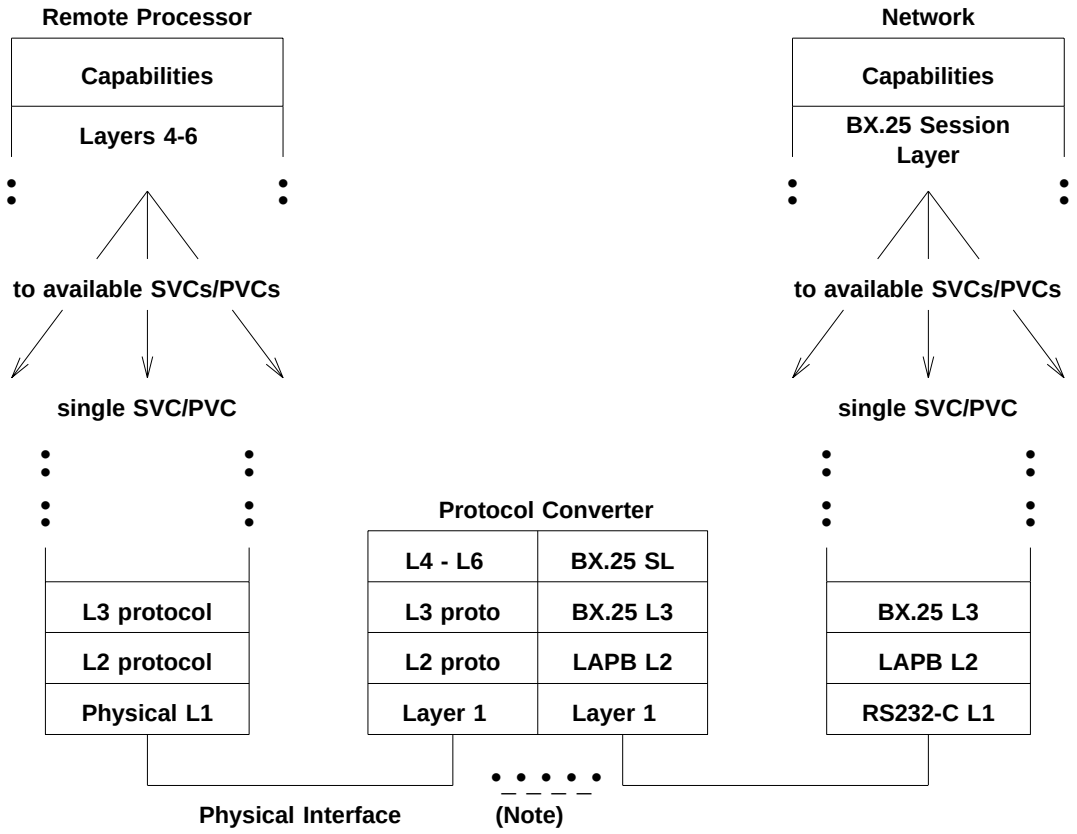
From the network side, the network session layer receives capabilities messages (for example, RC) from the application. The network session layer packetizes the capabilities messages and then passes the resulting packets to the lower layers for further processing and ultimate transmission to the remote processor.

1.9.1 INTERACTIONS AMONG BX.25 LAYERS

The network session layer sends its resulting messages (that is, packetized messages) to Layer 3 for processing by the BX.25 packet layer functions at the network side. The resulting packets are then passed to the network link layer (Layer 2) for balanced link access procedure (LAPB) processing. Finally, at the physical layer (Layer 1) the messages are sent over one of the network physical links that is connected to the appropriate remote processor or protocol converter that is to receive the message.

Figure 1-3 indicates the relationships among the layers on the remote processor side and the network side that are required for providing the capabilities on the remote processor.

If the remote processor does not use the same protocols that the network uses, a protocol converter is required to convert from the network protocols to the remote processor protocols. Such conversion is not covered in this document. The lower/higher layers on the remote processor are likewise not covered in this document.



Notes:

1. **If the protocols for the remote processor and the network are not compatible, a protocol converter is required; otherwise, the remote processor physical interface connects directly with the network.**
2. **There are up to 7 SVCs and/or up to 4 PVCs over a single data link.**
3. **SL: Session Layer**
   L1: Layer 1
   L2: Layer 2
   L3: Layer 3
   L4 - L6: Layer 4 - Layer 6

Figure 1-3 — Remote Processor/BX.25 Network Layered Protocols

1.9.2 INTERACTIONS AMONG TCP/IP/ETHERNET LAYERS

Table 1-1 compares the protocol stacks of the BX.25 layers to those of TCP/IP layers in the *5ESS*-2000 switch ASM—RCOS interface.

Table 1-1 — Comparison of BX.25 and TCP/IP Protocol Stacks

BX.25 STACK	LAYER	TCP/IP STACK
BX.25	Session	TCP
BX.25	Packet Transport	TCP
BX.25	Network	IP
BX.25	Data Link	Ethernet Data Link Layer, IEEE 802.2
BX.25	Physical	Ethernet Physical Layer, IEEE 802.3

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF FIGURES

LIST OF TABLES

2. LOWER LAYERS

The BX.25 is a layered data communication protocol, employing synchronous communication. The BX.25 protocols provide an interface compatible with the 1980 issue of CCITT Recommendation X.25 at Layers 1, 2, and 3 which are as follows for the *5ESS*®-2000 switch RCOS interface:

- Layer 1 — The physical layer uses the EIA standard RS232-C up to 9.6 kbps on a TN 75C circuit pack or a V.35 or RS449 connection for transfer rates above 9.6 kbps on a TN 82 circuit pack.
- Layer 2 — The link layer uses LAPB.
- Layer 3 — The packet layer uses the BX.25 packet layer protocol.

The physical and data link layers in the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack are based on two commonly-used Ethernet standards that can coexist on the same cable, but cannot interwork:

- Ethernet Version 2.0, a de facto standard
- ISO 8802.3, identical to IEEE Standard 802.3, adopted by ISO as an International standard.

Both standards use the carrier sense multiple access/collision detection (CSMA/CD) for medium-access mechanism. Although the hardware for both standards is compatible, the frame content is not.

2.1 PHYSICAL LAYER

2.1.1 PHYSICAL LAYER IN BX.25 PROTOCOL STACK

The Physical Layer of BX.25 provides a standard electrical and procedural interface. The physical layer provides the *5ESS*-2000 switch and the remote processor with a means of exchanging data through a synchronous, full duplex physical connection.

The Electronic Industries Association (EIA) RS-232C interface serves as BX.25 Layer 1 interface up to the data rate of 9.6 kbps. A subset of the RS-232C pins (pins 1 through 8, 15, 17, and 20) has been implemented in the *5ESS*-2000 switch RCOS interface. Pins 1 through 8 and 20 are included because they are required for basic operation; pins 15 and 17 are included because they are required for synchronous operation.

Table 2-1 lists the subset of the RS-232C pins supported by the *5ESS*-2000 switch:

Table 2-1 — RS-232C Circuits Supported by the *5ESS*-2000 Switch

PIN	CIRCUIT MNEMONIC	CIRCUIT NAME ACRONYM	CIRCUIT NAME	REASON REQUIRED
1	AA	-	Protective Ground	a
2	BA	TD	Transmitted Data	a
3	BB	RD	Received Data	a
4	CA	RTS	Request To Send	a
5	CB	CTS	Clear To Send	a
6	CC	DSR	Data Set Ready	a
7	AB	-	Signal Ground (Common Return)	a
8	CF	CD	Received Line Signal Detector (Carrier Detect)	a
15	DB	-	Transmitter Timing (DCE source)	b
17	DD	-	Receiver Timing	b
20	CD	DTR	Data Terminal Ready	a

Note(s):
a. Circuits required for basic operation
b. Circuits required for synchronous operation

Table 2-2 lists the physical layer options supported by the *5ESS*-2000 switch.

Table 2-2 — RS-232C Physical Layer Parameters

PARAMETER	PARAMETER VALUE
Data Rate	Up to 56 kbps (other options - 1.2, 2.4, 4.8, 9.6 kbps)
Full/Half Duplex Transmission	Full
Data Set	RS-232C, V.35, or RS-449 compatible data set
Continuous/Switched Carrier	Continuous
Mode of Operation (block or send/receive mode)	Block mode

2.1.2 PHYSICAL LAYER IN TCP/IP PROTOCOL STACK

The *5ESS*-2000 switch ASM physical layer in the TCP/IP protocol stack uses IEEE 802.3 10BaseT hub cable type with the following characteristics:

- Data rate: 10 Mbps
- Signaling method: baseband
- Media: unshielded twisted-pair wire
- Maximum segment length: 100 m
- Topology: star.

Note: The customer is responsible for providing the equipment needed to connect the OS to the ASM Ethernet hub.

2.2 DATA LINK LAYER

The purpose of the data link layer is to assure that frames are transported reliably across a data link. In transporting the frames, the data link layer performs the following major functions:

- Link initiation/termination
- Error control using cyclic redundancy check (CRC) data assurance, frame sequence numbering and frame retransmission
- Flow control.

2.2.1 DATA LINK LAYER IN BX.25 PROTOCOL STACK

BX.25 link layer supports link access procedure B (LAPB), which is compatible with CCITT Recommendation X.25 LAPB. The *5ESS*-2000 switch RCOS Layer 2 interface supports the single link LAPB protocol in the basic operation mode (that is, modulo 8 sequence numbering). The multiple link protocol is not supported.

The *5ESS*-2000 switch RCOS interface supports synchronous data link. Dial-up connection is not supported. In a DTE/DTE (point-to-point) connection, the network (*5ESS*-2000 switch) uses station address B (that is, DCE address) or station address A (that is, DTE address). Addressing must be successfully resolved with the remote processor before the data link is brought up.

Table 2-3 lists the data link layer LAPB parameters (for example, timers and counters) supported by the *5ESS*-2000 switch.

Table 2-3 — Data Link Layer LAPB Parameters

PARAMETER	FUNCTION	PARAMETER VALUE[a]
T1 Acknowledgement timer	Max. time to wait for Layer 2 protocol (Acknowledge-acknowledgment. At time-out, resend the frame acknowledgement timer); for example, SABM/SABME, DM, DISC FRMR, I, or REJ frame; or a frame with P-bit set to 1.	3 secs
T2 (Response Delay Timer)	Max. time to delay a response at receiver side.	(<T1)
T3 (Inactive Link Timer)	Max. time to allow data link to be idle. When T1 is not running, T3 timer starts. At time-out, send an S-format frame with P-bit set to 1 over the data link.:	not used
T4 (Idle Channel Timer)	Max. time before idle link is considered as disconnected (out of service). Applicable to only SVC.	not used
N1[b]	Max. number of bits in an I-frame	2104 bits
N2	Max. number of transmitting a frame (N2 includes original transmission and: retransmissions).	7 (2-7)
k	Max. number of outstanding (that is, unacknowledged) frames.	3 (1-7)
Note(s): a. The table lists the network values and the values inside () indicate the valid range. b. N1 (2104 bits) is based on a packet size of 256 octets of user data.		

2.2.2 DATA LINK LAYER IN TCP/IP PROTOCOL STACK

The data link layer in the TCP/IP protocol stack consists of two sub-layers:

- Logical Link Control (LLC), which differentiates among the different types of network layer protocols. LLC has three variants: LLC1, LLC2, and LLC3. Of these variants, only LLC1 is used for TCP/IP. The LLC functionality is described in IEEE 802.2.
- Medium Access Control (MAC), which provides MAC addresses. The MAC addresses identify the *destination* and *source* network cards between which the frame travels. A MAC address is configured into the hardware card by the manufacturer, typically on programmable-read-only memory. The MAC sub-layer functionality is described in IEEE 802.3.

Figure 2-1 displays the IEEE 802.2 and IEEE 802.3 LLC frame formats:

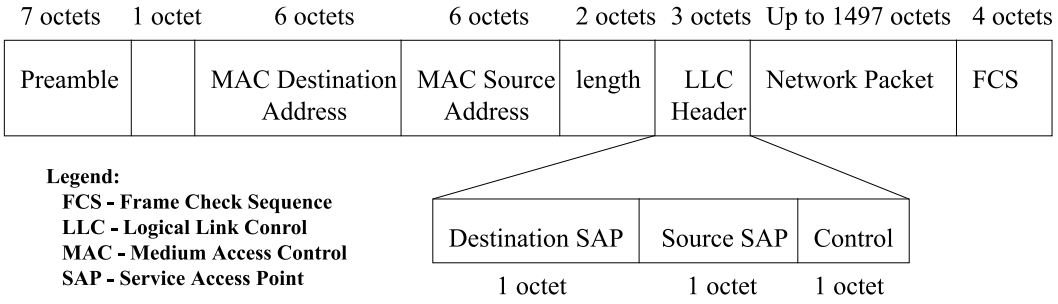


Figure 2-1 — IEEE 802.3/802.2 Frame Format

The *destination* and *source* Service Access Points (SAPs) identify the destination and source protocols, whose values are typically identical. SAPs are registered with ISO to ensure that systems do not interfere with each other.

2.3 NETWORK LAYER IN TCP/IP PROTOCOL STACK

The Internet Protocol (IP) is the Network Layer, Layer 3, in the TCP/IP communications protocol model. The IP provided by the vendor of the ASM platform is based on:

- RFC 791, Internet Protocol, defines the IP protocol and provides an introduction to the architecture of Internet.
- RFC 792, Internet Control Message Protocol, defines ICMP, which provides routing, diagnostic, and error functionality for IP.
- RFC 1122, Requirements for Internet Hosts—Communications Layers, defines the requirements for Internet Host Software.

The IP Layer is primarily responsible for routing (transferring data to the correct machine on the correct physical network) through:

- Addressing. The IP address is typically allocated to a network connection when the TCP/IP software is installed. The IP address is configured to represent a network number and a unique host connection on that network; if multiple connections are made to a computer, each connection requires a unique IP address. An ASM with a single Applications Processor (AP) has a single IP address; an ASM with multiple APs has one IP address for each AP.
- Fragmentation/re-assembly. When the higher layer protocol sends a datagram larger than the maximum size that the medium can handle, the sending IP divides the datagram into smaller datagrams, and the receiving IP reassembles the original datagram for its higher layer protocol. The maximum size datagram that IP can use, the Maximum Transmission Unit (MTU), must be set to at least 576 octets. The MTU must be set to 1492 octets, the IEEE 802.3 maximum data size, if the network supports this size.
- Obtaining a suitable type of service (TOS). IP provides different types of performance for datagrams, low delay, high throughput, high reliability, and low cost. Although the higher level protocol, such as Transmission Control Protocol (TCP), can set option bits to select these types of performance, most routers ignore the selection.

Note: The customer is responsible for configuring the IP parameters for an ASM when the ASM is installed. These parameters may include only the IP address and the MTU. The mandatory requirements for IP are specified in the Internet Protocol section of RFC 1122.

2.4 PACKET TRANSPORT LAYER

The packet layer governs the transfer of packets and facilitates reliable and efficient data communications by providing the functions:

- Multiplexing over a single physical link
- Error control
- Flow control
- Reset and restart.

The following parameters/options of the packet layer are set for the network. The recommended value is marked by "*".

- D-bit (Delivery Confirmation Bit): The network will set the D-bit to one in the call accept (layer 3) packet, and in the data (Layer 3) packets of the Session Normal Disconnect Message (SNDM), Layer 4, the Session Abort Disconnect Message (SADM), Layer 4, and the Session Normal Disconnect Message Reject Message (SNDMRM), Layer 4. The D-bit should be zero for all other Layer 3 data packets. Because all remote systems can not set the D-bit as specified in PUB.54001 and PUB.54001A, the network will accept a Layer 3 call request packet and Layer 3 Data packets for the SNDM, SADM, or the SNDMRM with or without the D-bit set to one.
- M-bit (More Data Bit): M-bit is set on if a data packet is not the last packet in a sequence; and it is set off if it is the last packet or the only packet in a sequence.
- Q-bit (Qualified Data Bit): Q-bit is not used.
- Packet size: 256* octets (other options - 128, 512 octets)
- Window size: 3* (other option - 2)
- Maximum number of virtual circuits: The *5ESS*-2000 switch RCOS interface supports either up to 7 SVCs and/or up to 4 PVCs over a single data link.
- Fast Select Facility: It is required for PVCs and SVCs.
- Numbering of packets: The packet sequence numbers cycle through the entire range 0 through 7.
- Recovery from nonacknowledgment of data packets: If the data packet acknowledgment timer (T25) times out, the network resets the logical channel. To decrease the probability of a lost window-rotation indication packet, the network will send a receive ready (RR), receive not ready (RNR), or data packet every T24 time units while the logical channel is in the flow control ready state.
- Recovery from receipt of erroneous data packets: The network resets the logical channel.

The following timers (Table 2-4) and attempt counters (Table 2-5) are currently used in the *5ESS*-2000 switch RCOS interface. They are defined because a response to a transmitted packet may be required within a stated maximum time.

Table 2-4 — Packet Layer Timers

TIMER	TIMER NAME	TIME-OUT VALUE [a]
T20	Restart Request Response Timer	180 secs (30-180 secs)
T21	Call Request Response Timer	200 secs (100-200 secs)
T22	Reset Request Response Timer	180 secs (30-180 secs)
T23	Clear Request Response Timer	180 secs (90-180 secs)
T24	Window Status Transmission Timer	60 secs (30-60 secs)
T25	Data Packet Acknowledgment Timer	150 secs (70-150 secs)
T26	Interrupt Response Timer	Not used
T27	Reject Response Timer	Not supported

Note(s):
a. The values inside () indicate the valid range, and the stated values are network values.

Table 2-5 — Packet Layer Attempt Counters

COUNTER	COUNTER NAME	VALUE[a]
R20	Restart Request Retransmission Count	1 (1-5)
R22	Reset Request Retransmission Count	1 (1-2)
R23	Clear Request Retransmission Count	1 (1-10)
R25	Data Packet Retransmission Count	Not supported
R27	Reject Retransmission Count	Not supported

Note(s):
a. The values inside () indicate the valid range, and the stated values are network values.

2.4.1 PACKET TRANSPORT LAYER IN BX.25 PROTOCOL STACK

The packet layer of BX.25 provides two distinct services: the permanent virtual circuit (PVC) and switched virtual circuit (SVC). The packet types needed to support PVC or SVC are specified in the BX.25 documents (PUB.54001 and PUB.54001A) except for the following:

- Diagnostic packets are not generated and are ignored.
- Interrupt packet is not used by the network.
- Throughput classes are not supported.

2.4.2 PACKET TRANSPORT LAYER IN TCP/IP PROTOCOL STACK

The Transmission Control Protocol (TCP) is the Transport Layer, Layer 4, in the TCP/IP communications protocol model, providing a reliable connection-oriented transport service. The vendor of the ASM platform provides the TCP based on:

- RFC 793, Transmission Control Protocol, which defines the TCP protocol.
- RFC 1122, Requirements for Internet Hosts—Communications Layers, which defines the requirements for Internet Host Software.

Communication between two machines requires that a connection be established. A segment (block of data) exchanged between TCP and IP contains the TCP header and the application layer data. TCP provides reliability through:

- Error detection and correction for segments corrupted by the medium or lower layers.
- Flow control to prevent the transmitter from overrunning the receiver.
- Re-sequencing to handle segments delivered out of order from IP. For example, IP can route via different routes successive datagrams in a segment.
- Removing duplicate segments that result from the error-recovery mechanisms TCP uses.

TCP helps the above facilities achieve reliability by:

- Using sequence numbers to identify data.
- Using positive acknowledgments of data received in the correct sequence.
- Retransmitting segments that have not been acknowledged within a specified time.

2.4.2.1 TCP Configurable Parameters

2.4.2.1.1 Maximum Segment Size (MSS)

The MSS must be set to at least 576 octets and, if supported by the network, to 1024 octets.

2.4.2.1.2 Receive Maximum Window Size

TCP uses a sliding window mechanism to allow multiple segments to be transmitted without waiting for each individual acknowledgment. The receive maximum-window-size must be 3.

2.4.2.1.3 Time to Live (TTL)

A TCP sender sets a time-to-live, TTL, field in a datagram, and a router decrements this TTL as the datagram passes through. To prevent a datagram looping in the network, a router discards a TTL that has been reduced to 0. The TTL value, recommended at 64, must be configurable. Refer to RFC 1700, Assigned Numbers.

2.4.2.1.4 Retransmission Time-out

The ASM platform vendor's TCP includes the implementations of:

- Jacobson's transmission algorithm that combines *slow start* with *congestion avoidance*
- Karn's algorithm and Jacobson's algorithm for computing the retransmission time-out (RTO)
- Jacobson's algorithm for computing smoothed round trip (RTT) time
- The *exponential backoff* for successive RTO values for the same segment.

The following values must be used to initialize the estimation parameters for a new connection:

- RTT = 0 seconds
- RTO = 3 seconds.

2.4.2.1.4.1 Connection Failures

Two threshold values, R1 and R2, measure the amount of retransmissions in TCP. The procedures that use these values are implemented in the ASM platform vendor's TCP as described in RFC 1122. The values of R1 and R2 must be:

- R1, at least 3 retransmissions at the current RTO value
- R2, at least 100 seconds for data, or at least 180 seconds for connection establishment (SYN segments).

2.4.2.1.5 Keep Alive

The optional TCP Keep Alive procedure, which verifies that a connection is operational, is not used for the RCOS interface. Instead, a *Heartbeat Check* mechanism is used in the application level.

2.4.3 APPLICATION PROGRAM INTERFACE—TRANSPORT LAYER INTERFACE

The *5ESS*-2000 switch Recent Change process uses the Transport Layer Interface (TLI) of *UNIX*[1] System V, Release 4.0, to access the services of the transport protocol, TCP. This TLI interface, based on the ISO Transport Service Definition, is described in *UNIX Network Programming*, W. Richard Stevens, Prentice Hall Software Series, ISBN 0-13-949876-1. The ISO Transport Service Definition provides the following services:

- Connection establishment
- Data transfer
- Expedited data transfer
- Normal release
- Abnormal release.

For the RCOS interface, the OS and *5ESS*-2000 switch act as client and server, respectively. A TLI application can use any of the following services:

- Connection-oriented service without orderly release (T_COTS)
- Connection-oriented service with orderly release (T_COTS_ORD)
- Connectionless service (T_CLTS).

A server can be either a concurrent server in which a client's request may be handled by another process, or an iterative server in which a client's request is handled by the server process itself. *5ESS*-2000 switch Recent Change OS is a concurrent server that uses the T_COTS_ORD type of service.

2.4.3.1 TLI Functions

The *5ESS*-2000 switch Recent Change interface uses the connection establishment, data transfer, and orderly release services of TCP. To obtain these services, RC invokes the following TLI functions. The *5ESS*-2000 switch Recent Change interface also uses the non-blocking I/O functionality supported by TLI. These TLI functions are listed in Table 2-6 available in the *UNIX* System V Release 4.0 Network Services Library (NSL):

---

1. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

Table 2-6 — Transport Layer Interface (TLI) Functions

TYPE OF SERVICE	FUNCTION	PARAMETERS
Connection establishment	t_open t_bind t_listen t_accept	(char **pathname*, int *oflag*, struct t_info **info*) (int *fd*, struct t_bind **request*, struct t_bind **return*) (int *fd*, t_call **call*) (int *fd*, int *connfd*, struct t_call **call*)
Data transfer	t_send t_rcv	(int *fd*, char **buf*, unsigned int *nbytes*, int *flags*) (int *fd*, char **buf*, unsigned int *nbytes*, int *flags*)
Release	t_sndrel t_rcvrel	(int *fd*) (int *fd*)
Connection establishment, Data transfer, Release	t_getstate	(int *fd*)
Close transport endpoint	t_close	(int *fd*)

- t_open(char ***pathname*, int *oflag*, struct t_info ***info*) - This function opens the *UNIX* system file that identifies the transport provider, and returns a file descriptor to be used for the other TLI functions.
  - *pathname* - This value depends on the transport provider. For TCP, this value is /dev/tcp.
  - *oflag* - This value indicates the *UNIX* system open flags. The *5ESS*-2000 switch Recent Change interface uses O_RDWR (open for reading and writing) and O_NDELAY (do not block on t_connect, t_listen, t_rcv or t_snd).
  - *info* - This value provides information about the transport provider.Since the *5ESS*-2000 switch Recent Change OS does not use this information, the pointer is NULL.
- t_bind(int *fd*, struct t_bind ***request*, struct t_bind ***return*) - This function assigns an address to a transport endpoint.
  - *fd* - This value provides the file descriptor of the transport endpoint.
  - *request* - This value points to a t_bind structure, which may contain a protocol-specific address. If the request is NULL or the address length is 0, this indicates that the caller is asking the transport provider to assign the address. The *5ESS*-2000 switch Recent Change OS provides the address.
  - *return* - This value points to a t_bind structure that contains information about the connection.
- t_listen(int *fd*, struct t_call ***call*) - This function waits for a connection request.
  - *fd* - This value provides the file descriptor of the transport endpoint.
  - *call* - This value points to a t_call structure; on return from the receipt of a connect request, this value indicates connection parameters, such as the client's address.

- t_accept(int *fd*, int *connfd*, struct t_call **call*) - This function accepts a connection request.
  - *fd* - This value provides the file descriptor of the transport endpoint where the connect request arrived.
  - *connfd* - This value provides the file descriptor of the transport endpoint where the connection is to be established.
  - *call* - This value points to a t_call structure that contains the address assigned, whether by the caller or the transport provider. If the request is Null, the provider does not return the address. *5ESS*-2000 switch Recent Change OS uses a NULL value for request.
- t_snd(int *fd*, char **buf*, unsigned int *nbytes*, int *flags*) - This function sends data.
  - *fd* - This value provides the file descriptor of the transport endpoint to send data.
  - *buf* -This value identifies the buffer for the data.
  - *nbytes* - This value indicates the number of bytes to send.
  - *flags* - This value sets the flags to 0, or a combination of the T_EXPEDITED flag for "send expedited (out-of-band) data" and the T_MORE flag for "more data to send." The *5ESS*-2000 switch Recent Change OS processes the 0 or the T_MORE flag, but ignores the T_EXPEDITED flag.
- t_rcv(int *fd*, char **buf*, unsigned int *nbytes*, int *flags*) - This function receives data.
  - *fd* - This value provides the file descriptor of the transport endpoint to receive data.
  - *buf* -This value identifies the buffer for the data.
  - *nbytes* - This value indicates the number of bytes to receive.
  - *flags* - This value sets the flags to 0, or a combination of the T_EXPEDITED flag for "send expedited (out-of-band) data" and the T_MORE flag for "more data to send." The *5ESS*-2000 switch Recent Change OS processes the 0 or the T_MORE flag, but ignores the T_EXPEDITED flag.
- t_sndrel(int *fd*) - This function sends an orderly release request.
  - *fd* - This value provides the file descriptor of the transport endpoint to send an orderly release request.
- t_rcvrel(int *fd*) - This function receives an orderly release request.
  - *fd* - This value provides the file descriptor of the transport endpoint to receive an orderly release request.
- t_getstate(int *fd*) - This function queries the current state of the transport connection, returning the current state of the transport endpoint. Table 2-7 lists the states.
  - *fd* - This value provides the file descriptor of the transport endpoint.

Table 2-7 — Transport Connection States

STATE	DESCRIPTION
T_DATAXFER	Data transfer
T_IDLE	Bound, but idle
T_INCON	Incoming connection pending for server
T_INREL	Incoming orderly release (waiting for an orderly release request)
T_OUTCON	Outgoing connection pending for client
T_OUTREL	Outgoing orderly release (waiting for an orderly release indication)
T_UNBND	Initialized, but unbound

- t_close(int fd) - This function closes the transport endpoint. A transport endpoint can be closed without terminating a connection. (Refer to t_accept for a description of the two transport enpoints).
  - fd - This value provides the file descriptor of the transport endpoint.

2.4.3.2 Connection Establishment Service

Lucent Technologies provides a DSlogin() process to the RCOS to be used in establishing a connection. (Refer to Section 2.4.4, “Security.”) The parameters for DSlogin() are a user-id, a password, and TCP port numbers (such as 30000 or 30001).

When the ASM receives a valid user-id and password for the RCOS interface, it will start an instance of the *5ESS*-2000 switch Recent Change server. (Refer to Section 2.4.4, “Security.”) The arguments specified in the ASM configuration file, including the TCP Port Number associated with the user-id, are passed to the *5ESS*-2000 switch Recent Change server.

Note: The customer is responsible to specify the TCP port numbers associated with the RCOS user-id. The user-id, passwords, and TCP port numbers must be included in the ASM configuration file and used by the RCOS Client.

The *5ESS*-2000 switch supports multiple Recent Change server sessions, but for each connection RCOS must use a port number that is unique at its local host.

2.4.3.3 Data Transfer Service

Once DSlogin() to RCOS has been completed, data can be sent. Figure 2-2 illustrates the data transfer service.

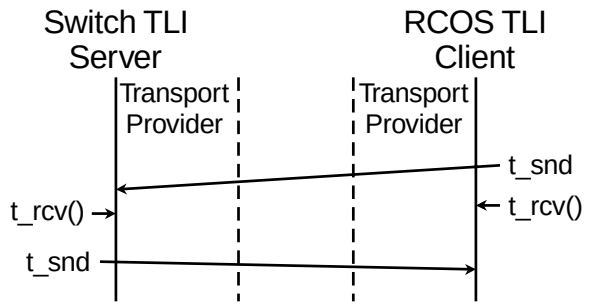


Figure 2-2 — Transport Layer Interface (TLI) Data Transfer

The RCOS client/server processes for the BX.25 protocol interface rely on the BX.25 session layer to provide functionality to process RCOS message structures that are

larger than the size of the application input buffer. To support this functionality in the RCOS TCP/IP interface, each capabilities message structure exchanged between the RCOS TCP/IP client/server processes must contain a 3-byte header of the following format:

BYTE 1	BYTE 2	BYTE 3
Header-id	Length	

In this format, “Header-id” indicates 0x01, the [ctrl-key]-A character; “Length” indicates the length of the message, not including the three bytes in this header.

The messages exchanged between the client and server have the following format:

RCOS TCIP/IP 3-byte Header	RCOS Message Header	RCOS Message Data

2.4.3.4 Normal Release Service

To terminate the session, RCOS issues an orderly release. Figure 2-3 illustrates a TLI orderly release.

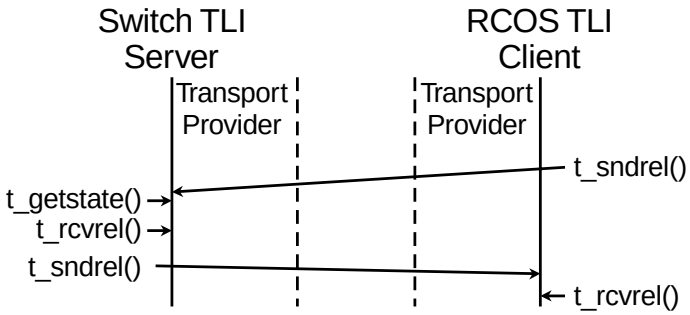


Figure 2-3 — Transport Layer Interface (TLI) Orderly release

2.4.4 SECURITY

The *5ESS*-2000 switch TCP/IP protocols open the interface to the switch and require additional security measures to control access. From the switch perspective, the *5ESS*-2000 switch is the server and the Operations Systems are the clients.

The first level of security for a remote OS that connects via a WAN is the firewall provided by the router network administrator. The next level of security, the connection to the ASM’s LAN, a private service provider-owned Intranet, is described in this section. Additional levels of security can be provided by the TCP/IP or User Datagram Protocol (UDP)/IP application.

As an ASM client, RCOS must implement the ASM-imposed security method described in this section. The client process must log in through IP Port DSLOGINPORT (Port 24064) using the DSlogin() function, and must provide a user-id, password, TCP port number, and security-class. The ASM, which supports only CLEARTEXT (plain text) communication between the client and the server, authenticates users through a standard *UNIX* login program. The RC on ASM application assumes the use of a number of standard logins for accessing RC over the ASM. The ASM validates the user-id and password according to the security-class specified.

RCOS uses the CLEAR_PASSWORD security class for which the password is sent as a clear (unencrypted) value. CLEAR_PASSWORD is the only security-class implemented in the initial release of the ASM in the 5E13 software release.

The RCOS TCP port associated with the user-id and specified in the ASM configuration file must be used. The client then uses the other TLI functions to transfer data and to release the connection. Software that includes the following application program interfaces (APIs) resides on the operations support system client and is used to log in and communicate with the ASM security server process DSlogind:

- DSlogin() — for logging in to an ASM
- DSremwrite() — for sending information to the ASM
- DSremread() — for retrieving information from an ASM
- DSlogout() — for terminating the login session.

2.4.4.1 ASM Server Security Process

ASM provides the DSlogin() process for the *5ESS*-2000 switch Recent Change process, which uses the TCP port. When the ASM receives a request for that port, the DSlogind server process is spawned. DSlogind checks the *user-id* and *password* according to the *security-class* specified and, if these are valid, starts the *5ESS*-2000 switch Recent Change process.

2.4.4.2 OS Client Security Process

For the OS client, Lucent Technologies provides, in addition to the DSlogin() API, source code for establishing a connection for an ASM application such as the *5ESS*-2000 switch Recent Change server. The DSlogin() function encapsulates the bind() and connect() functions to establish a connection to the ASM, and provides and encapsulates the security classes that the ASM allows.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

3. SESSION LAYER

3.1 BX.25 SESSION LAYER PROTOCOL MESSAGES

The BX.25 Session Layer provides two categories of protocol services.

- It establishes, administers, and terminates communication between two higher-level entities (that is, application processes).
- It provides end-to-end control and synchronization of the transfer of data.

The session layer messages, message formats, and encoding are described in the documents “Operations Systems Network Communications Protocol Specification BX.25 Issue 3” and “Operations Systems Network Communications Protocol Specification Issue 3 Addendum A” (document numbers PUB.54001 and PUB.54001A).

The services of the session layer may be divided into three phases; namely, the session establishment phase, the data exchange phase, and the session termination phase.

The unit of data transferred between a pair of session layers is called the session protocol data unit (SPDU). There are two basic types of SPDUs:

- Data SPDU - SPDUs, which contain data and control information for a higher-level entity (for example, application process) engaged in the session are referred to as data SPDUs.
- Control SPDU - SPDUs, which are intended for the receiving session layer and consumed by the receiving session layer are referred to as control SPDUs. Control SPDUs contain Session Layer Control Messages, such as protocol messages used for session establishment, session termination, session recovery, and session resynchronization.

The following describes the protocol messages that must be implemented during these three phases of the session layer services, and the protocol messages that are not supported:

1. **Session Establishment Phase:** During the session establishment phase, a communication path is established between two peer application processes, residing on the network and the remote processor, respectively.

   The communication path or session has attributes associated with it and agreed upon by the peer application processes and session layers. These attributes (or parameters) are conveyed through the session initiation protocol messages. See “SESSION LAYER PARAMETERS”, Section 3.2, for their specified values used in the *5ESS*®-2000 switch RCOS interface.

   During the session establishment phase, Session Connect Message (SCM), Session Accept Message (SAM), and Session Reject Message (SRM) are used.

   For SVCs, the first octet of the call user data field in the Layer 3 call request packet is a hexadecimal C1. The remainder of the call user data field is the Session Connect Message. If the session is accepted, the Session Accept Message is returned to the remote processor in the Layer 3 call accepted packet as the called user data portion. If the session is rejected, then the Session Reject Message is returned to the remote processor in the Layer 3 clear request packet as the clear user data portion.

   Application process creation services are offered as a part of the session establishment procedure. A process that provides access to the network

capabilities (for example, recent change capabilities) will be created by the session layer procedures when this process is the designated destination address in the Session Connect Message. It should be noted that a process will be created on the receiving end of the Session Connect Message and that the remote process originating the session establishment request must have been created by the user and "running" when it requests the session establishment services.

2. **Data Exchange Phase:** The data exchange phase is entered upon successful completion of session establishment. During the data exchange phase, the peer application processes can exchange normal and control data.

   The SPDU, including the more data flag (MDF), is implemented, and the size of SPDU (1024 octets) is fixed. The optional facilities resynchronization and session recovery are not used. Session expedited data procedure (data transmission via the Layer 3 INTERRUPT packet) is not used.

   See "SESSION LAYER PARAMETERS", Section 3.2, for the parameters associated with data exchange procedure.

3. **Session Termination Phase:** During the session termination phase, the established session is terminated normally or abortively by either session partner. The session layers can cause only an abortive termination of the session due to the detection of any unusual condition, for example, link failures.

   The normal termination procedure should not cause any loss of data, since the procedure is done after the data exchange phase and completed only through the mutual consent of both session partners. However, the abortive termination of a session may result in some loss of data.

   During session termination, Session Normal Disconnect Message (SNDM), Session Abort Disconnect Message (SADM), and Session Normal Disconnect Message Reject Message (SNDMRM) are used.

   The network sends a reset packet before sending the SADM.

3.2 SESSION LAYER PARAMETERS

The following parameters and options must be set at session establishment time for the *5ESS*-2000 switch RCOS interface:

- Called Session Address = must be specified (It will be provided by *5ESS*-2000 switch administration.)
- Unique session connection identifier for calling side (It will be provided by *5ESS*-2000 switch administration.)
- Origin Type = BX.25 Application Level services provided by Session Layer (required in an SCM)
- Session Type (or Dialogue Type) = Two-way simultaneous (required in an SCM)
- Presentation Type of Session Data = ASCII (This attribute may be omitted from the SCM and the default, ASCII, assumed.)
- SPDU Size = 1024 octets (This includes two octets of session header and 1022 octets of user data.)
- Session Resynchronization: not used

- Session Recovery: not used
- User Data of SCM: This field contains the keyword supplied by *5ESS*-2000 switch administration.

With the Origin Type set as BX.25 Application Level services provided by Session Layer, and without Resynchronization option in effect, the session layer uses the single octet header.

The remote process is required to send the specific keyword to the network at session establishment time. The keyword contained in the user data of an SCM will be supplied to users by *5ESS*-2000 switch administration at the time capabilities are equipped.

The MDF, indicating whether a data SPDU is the last SPDU in a sequence of logically related SPDUs or not, is used in the *5ESS*-2000 switch RCOS interface.

The control data flag (CDF), denoting a data SPDU containing control data for the application process, is not used in the *5ESS*-2000 switch RCOS interface.

3.3 SESSION LAYER TIMERS AND ATTEMPT COUNTERS

The following timers and attempt counters are currently defined for the BX.25 Session Layer:

1. Session Establishment

   The network never initiates a session; it is always a called session.

   - ST2 (SCM Response Timer) - A timer is set by the calling session layer (the remote processor), and it times out if no response is received from the called session layer (network).
   - A4 (SCM Transmission Count) - An attempt counter is used by the calling session layer to determine the number of times to attempt to set up a session. If the attempt count has not been reached, the session layer will secure a new SVC from the packet layer and retransmit the SCM. If the count has been reached, the session is considered rejected; the session layer informs the application process and releases the SVC.

2. SPDU Flow
   - ST3 (SPDU Response Timer)
   - A1 (SPDU Retransmission Count).

3. SPDU Resynchronization
   - ST4 (SRESM Response Timer)
   - A2 (SRESM Transmission Count).

4. Session Recovery
   - ST5 (Wait for SRECM Timer)
   - ST6 (SRECM Response Timer)
   - A3 (SRECM Transmission Count).

Tables 3-1 and 3-2 list the values for the BX.25 session layer timers and attempt counters set in the network for the *5ESS*-2000 switch RCOS interface. In the

5ESS-2000 switch RCOS interface, the resynchronization and recovery options are not used and, therefore, the corresponding timers and counters are not in effect.

Table 3-1 — Session Layer Timers

TIMER	TIMER NAME	TIME-OUT VALUE[a]
ST2	SCM Response Timer	260 secs (30-600 secs)
ST3	SPDU Response Timer	Not used
ST4	SRESM Response Timer	Not used
ST5	Wait for SRECM Timer	Not used
ST6	SRECM Response Timer	Not used

Note(s):
a. The values inside () indicate the valid range, and the stated values are network values.

Table 3-2 — Session Layer Attempt Counters

COUNTER	COUNTER NAME	VALUE[a]
A1	SPDU Retransmission Count	Not used
A2	SRESM Transmission Count	Not used
A3	SRECM Transmission Count	Not used
A4	SCM Transmission Count	1 (1-10)

Note(s):
a. The values inside () indicate the valid range, and the stated values are network values.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

4. APPLICATION LAYER - RECENT CHANGE AND OFFICE RECORDS CAPABILITIES

This section consists of the following subsections:

- Section 4.1, “Capabilities Definition”

  The recent change (RC) capabilities allow customers to manipulate *5ESS*®-2000 switch office-dependent data, such as line, trunk or equipment data. Refer to the RC Text Interface document (see “REFERENCES,” Section 7, for the document number) for information about the *5ESS*-2000 switch data.

- Section 4.2, “Capabilities Invocation Procedures”

  This section describes the capabilities invocation procedures that the remote processor must follow in order to perform the desired functions. The required messages during invocation processes are defined in “MESSAGE STRUCTURE DEFINITIONS”, Section 5, and the information fields contained within the messages are defined in “MESSAGE FIELD DEFINITIONS”, Section 6.

- Section 4.3, “RCOS Interface Feature Capabilities.”

  Recent Change (RC) Operations Systems (OS) software availability often follows the availability of the switch software. In addition, the current operations environment requires local area service providers to coordinate the simultaneous installation of the *5ESS*-2000 switch and supporting OS software releases. This condition was addressed in the 5E8 software release by the RCOS Downward Compatibility (DC) feature, and is addressed in the 5E9(1) and later software releases by the RCOS APPTEXT Forward Translator (AFT) feature. These features de-couple the simultaneous development and deployment of the *5ESS*-2000 switch and supporting RC OSs.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

4.1 CAPABILITIES DEFINITION

The RC capabilities provided by the network can be categorized as follows:

- Change/add/delete/verify line data, such as
  - — Assign/remove line features
  - — Activate/deactivate line features
  - — Change line feature parameters
  - — Verify line features and parameters
  - — Station swap/rearrangement
- Change/add/delete/verify trunk data
- Change/add/delete/verify equipment data.

Any RC that can be performed on the *5ESS*®-2000 switch can also be performed through the RCOS interface.

The *5ESS*-2000 switch RCOS interface supports Immediate Recent Change, Batch Recent Change, Batch History and ASCII File Retrieval capabilities by allowing users to perform the following functions:

1. Immediate Text Recent Change
   - Enter recent changes to be applied immediately to the *5ESS*-2000 switch data base.
2. Batch Text Recent Change
   - Enter recent changes to be batched and applied to the *5ESS*-2000 switch data base at a specified release date and time (Batch Mode of Time Release type).
   - Enter recent changes to be batched and applied on demand to the *5ESS*-2000 base (Batch Mode of Demand type). With the demand type, no release time is associated with the batch file. The batch file can be released by using the batch mode release command, that is, MODE=BMR, CLERK=clerkid.
   - Change the release date, release time, or both for Batch Mode of Time Release type recent changes.
   - Delete recent changes that were entered into a batch file as “Batch Mode of Time Release type” recent changes.
   - Release Demand type or Time Release type recent changes that were batched.
   - Change the input mode (that is, immediate or batch mode) of recent changes from one to the other.
3. Batch History
   - Obtain reports on the history of batch recent changes that were entered into the delayed release clerk files (for a given clerk or for all clerks).
4. ASCII File Retrieval
   - Retrieve any ASCII file in the *5ESS*-2000 switch by specifying the full path of the directory containing the file(s) and the file name(s) to be retrieved.

The 5E8 software release introduced the Recent Change Operations System (RCOS) Downward Compatibility (DC) feature. This feature is used with immediate text RC messages. An RCOS DC feature description is provided in "RCOS DC/AFT FEATURE", Section 4.3.1.

The 5E9(1) software release introduced the Recent Change Operations System (RCOS) APPTEXT Forward Translation (AFT) feature. This feature is used with immediate text RC messages. An RCOS AFT feature description is provided in "RCOS DC/AFT FEATURE", Section 4.3.1. This feature replaces the DC feature.

The office records (OFRs) capabilities provide access to print *5ESS*-2000 records in various forms (for example, line forms, trunks forms). The *5ESS*-2000 switch RCOS interface supports the OFR capabilities by allowing users to perform the following functions:

1. Print OFR form(s) by form type or by category
   - To obtain *5ESS*-2000 switch OFR forms related to switch office dependent data such as lines, trunks, and equipment
2. List output requests/status/parameters
   - List all OFR output requests or OFR output requests of a specific status (for example, completed, pending, processing)
   - List all the internal operating and scheduling parameters of OFR capabilities
   - List the status of a specific request
3. Input/change OFR operating and scheduling parameters
4. Cancel an OFR print request.

The automatic customer station rearrangement (ACSR) RC capabilities provide access to retrieve information on ACSR RC line data due to daily integrated service digital network (ISDN) ACSR activities.

To accomplish these functions for RC, OFR, and ACSR RC capabilities, the *5ESS*-2000 switch RCOS interface supports messages in the Program Documentation Standards (PDS) (5E4 software release only) or Man-Machine Language (MML) (all software releases) format (see "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2) depending on *5ESS*-2000 switch offices.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

4.2 CAPABILITIES INVOCATION PROCEDURES

Before any functions can be performed, the remote processor must establish a session with the network. Once the session is established, it remains established until the remote processor or the network ends it.

During a session, a text interface channel is established when the network receives the first Text Recent Change request message. This interface channel will remain active until the network receives a text interface termination request message, abort request message, a session disconnect message, or until the network times out.

During a session, a craft shell interface channel is established when the network receives the first craft shell request message (for example, RC batch history or OFR processing request message). This interface channel will remain active until the network receives a craft shell interface termination request message or a session disconnect message, or until the network times out.

During a session, both interface channels (that is, craft shell interface and text interface channels) can be active in the same session simultaneously. Each interface termination request will release the network resources, but not terminate the network session.

Text RC request messages will be sent from the remote processor to the network to query or update the switch data base immediately if the recent change input is set to immediate mode. Text RC response messages will be sent from the network to the remote processor to acknowledge the completion of the requested action.

Text RC request messages will be sent from the remote processor to the network and will be stored in the network until they are to be released, if the recent change input is set to batch mode. These requests will be released for use by the user via batch mode release request for Demand type or Time Release type recent change, or will be released at a specific date and time for Time Release type recent change. During a session, the remote processor sends Batch History requests to obtain reports on the history of the batch recent changes that were batched and/or have been applied to the *5ESS*®-2000 switch data base.

During a session, the recent change input mode can be changed from immediate mode to batch mode or vice versa.

While the craft shell interface channel is active, OFR processing request messages will be sent from the remote processor to the network to obtain OFR form(s), to change/list the operating and scheduling parameters, to list all queued requests or the status of a scheduled request, and to cancel OFR job. At any time in a session after the completion of OFR processing, OFR output request messages will be sent to retrieve the output of processing.

Note: The craft shell interface channel may be inactive at this time.

To retrieve any ASCII file on the *5ESS*-2000 switch, the remote processor sends File Retrieval Output Request Message to the network by specifying the full path to the directory containing the file and the file name of the file to be retrieved. The remote processor sends back the File Retrieval Output Response Message that contains the requested ASCII file.

To retrieve information on daily ACSR activities, the remote processor sends OFR/ACSR RC Output Request Message to the network. The remote processor sends

back the OFR/ACSR RC Output Response Message that contains ACSR RC line data. The information will be transmitted back immediately after the ACSR RC file is processed.

During a session, the remote processor can send an RC view time-out request message to the network to query the time-out values for RC views. (See "RECENT CHANGE VIEW TIME-OUT MESSAGES", Section 4.2.6, and "RECENT CHANGE VIEW TIME-OUT MESSAGES", Section 5.2.5.)

The remote processor can send a status check message in order to check the progress of a request sent to the network, if the response time is longer than the preset time interval (that is, retrieved from the RC view time-out request message). Also the remote processor can check the status of the current session by sending a status check message.

During a session, the remote processor sends an abort request message to abort a request or an activity. The interface channel (text RC interface channel or craft shell interface channel) associated with the aborted request/activity will be terminated, but the session will remain active.

The remote processor sends a Session Abort Disconnect Message (SADM) to abort a session.

During a session, the session time outs if it is idle for 15 minutes (that is, not processing a request or sending/receiving messages). When the time-out occurs, the network sends an SADM message to the remote processor indicating that the session is being aborted due to inactivity.

The network does not send a message to the remote processor of its own accord unless the network has problems or when the time-out condition occurs, that is, the network is always replying to a request from the remote processor.

4.2.1 TEXT IMMEDIATE RECENT CHANGE MESSAGES

This section specifies the interface messages for the remote processor to access the text immediate RC capabilities. The requests will be processed and the changes will be immediately applied to the switch data base, and the responses will be sent to the remote processor.

The remote processor sends Text Immediate Recent Change/Verify Request Messages to the network to access the *5ESS*-2000 switch data base. A request may consists of one or more RC input lines. [See "(Immediate) Recent Change/Verify Request Message", Section 5.2.1.1.]

In response to the remote processor request, the network sends Recent Change or Verify Response Message(s) (see "Recent Change Response Message", Section 5.2.1.2, and "Verify Response Message", Section 5.2.1.3) to the remote processor. Each RC input line in the request message from the remote processor results in one response message from the network. In addition, the network sends one more response message (that is, Finish Request Process Response Message) to indicate the finish of the request process. Therefore, if the network receives a Recent Change Request Message containing five RC input lines, it will send five separate Recent Change Response Messages to the remote processor and one more response message indicating the finish of the request process.

To send multiple response messages in response to one single request message, the continuation flag (see “CONTINUATION FLAG”, Section 6.4) is set on until the last response message is sent.

4.2.1.1 Recent Change/Verify Request Message

To change, delete, insert, or verify data in the switch data base immediately, the remote processor sends the network a Recent Change/Verify Request Message (see “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1). This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMRC and the request type (see “REQUEST TYPE”, Section 6.2) set to RCMRCTXT. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1. To verify, change, delete, or insert data in the switch data base, the message data consists of the “MODE=IM” input line (optional) and one or more form action lines (mandatory), where each input line (MODE input line or form action line) is delimited by a control character “carriage return” (<cr>).

The input line “MODE=IM” (see “GENERAL FORMAT FOR MESSAGE DATA”, Section 5.1.2) sets the input mode to “immediate” mode such that subsequent form entries are applied to the switch data base immediately. This is the default input mode if mode was not previously set. If the “MODE=IM” input line is not specified, the immediate mode will be automatically assumed by the network. This input mode will remain active until the input mode is requested to change or the session/text recent change interface is terminated. Therefore, the subsequent recent change/verify messages do not need to specify this input line in the request messages.

Each form action line contains the following major subfields: view, action, attribute name, and value pairs (see “VIEW”, Section 6.5, “ACTION”, Section 6.6, and “ATTRIBUTE NAME AND VALUE PAIRS”, Section 6.7) followed by action at the end of a form action line. Each attribute name and value pair is separated from the other pair by a separator comma “,”. Each form action line is responsible for one action (change, delete, insert, or verify) upon the specified attributes.

To change, delete, or insert data in the switch data base, the view field is set to an appropriate value (see “VIEW”, Section 6.5). The action field (see “ACTION”, Section 6.6) is set to CHG for changing attribute values in the data base, set to OUT for deleting attributes from the data base and set to NEW for inserting attributes into the data base. Each attribute name and value pair is set to appropriate value.

To query the switch data base for data verification, the view field (see “VIEW”, Section 6.5) is set to an appropriate value. The action field (see “ACTION”, Section 6.6) is set to VFY, MVFY, or SHVFY for verifying the switch data. Attribute name and value pair(s) (see “ATTRIBUTE NAME AND VALUE PAIRS”, Section 6.7) are set to appropriate values of the key fields as specified in individual views. Refer to the RC Text Interface document (see “REFERENCES”, Section 7, for the document number) for the key fields of individual views.

4.2.1.2 Recent Change Response Message

In response to the remote processor request, the network sends Recent Change Response Message(s) (see “Recent Change Response Message”, Section 5.2.1.2) to the remote processor. Each form action line for insert, delete, or change operation in the request message from the remote processor generates a Recent Change Response Message.

If the header of the request message is valid and no error is encountered on a form action line of the Recent Change Request Message (see “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1), the network sends a Recent Change Response Message (see “Recent Change Response Message”, Section 5.2.1.2) with the reply field (see “REPLY”, Section 6.8) set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN (that is, continuation flag on). The message data format is described in “Recent Change Response Message”, Section 5.2.1.2. The msg.# (message number) is an integer indicating the corresponding input line in the request. The <nl> (new line) character separates msg.# from OK, which indicates a successful RC insert, delete, or change.

If the header of the request message is valid and an error is encountered on a form action line of the Recent Change Request Message (see “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1), the network sends a Recent Change Response Message (see “Recent Change Response Message”, Section 5.2.1.2) with the reply field (see “REPLY”, Section 6.8) set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN. The message data (see “Recent Change Response Message”, Section 5.2.1.2) is populated with error code (see “ERROR CODES FOR REPLY OPCODE RCMRCERROR”, Section 6.25.2), msg.# (message number), <nl> (new line), error input, and an error message, which is an ASCII string.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify

Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.1.3 Verify Response Message

There are two different formats for Verify Response Messages, one for a successful verify request and one for an erroneous verify request. Each form action line for data verification in the request message will generate a Verify Response Message.

If the header of the request message is valid and no error is encountered on a “verify” form input line of the Recent Change/Verify Request Message (see “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1), the network sends a Verify Response Message (see “Verify Response Message”, Section 5.2.1.3) containing the requested verification data to the remote processor.

The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN.

The message data for a successful verify request has the format as a FORM input line, that is, the message data is comprised of the FORM name and attribute name and value pair(s).

If the output, including message header, is longer than the message size of 1024 characters, the network breaks the output into segments. The network does not attempt to break the output on line or page boundaries; it segments the output at boundaries of size 1024 octets, wherever that may be.

If the header of the request message is valid and an error is encountered on a “verify” form input line of the Recent Change/Verify Request Message (see “(Immediate) Recent Change/Verify Request Message”, Section 5.2.1.1), the network sends a Verify Response Message (see “Verify Response Message”, Section 5.2.1.3) indicating the reason of failure (that is, the error) to the remote processor. The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN. The message data (see “Verify Response Message”, Section 5.2.1.3) is populated with error code (see “ERROR CODES FOR REPLY OPCODE RCMRCERROR”, Section 6.25.2), msg.# (message number), <nl> (new line), error input, and an error message, which is an ASCII string.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section

6.4) is set to RCMNOCONT. The message data contains an error code (see “ERROR CODES FOR REPLY OPCODE RCMRCERROR”, Section 6.25.2) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change/Verify Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.1.4 Finish Request Process Response Message

In response to the Recent Change/Verify Request Message from the remote processor, the network sends Recent Change Response Message(s) (see “Recent Change Response Message”, Section 4.2.1.2) and/or Verify Response Message(s) (see “Verify Response Message”, Section 4.2.1.3) to the remote processor. The network will respond with a message for each RC input line in the request message.

In addition, the network sends one more response message (n+1th response message, assuming that total number of RC input lines in the request message is n) to indicate the finish of the request process.

The information fields for Finish Request Process Response Message (see “Finish Request Process Response Message”, Section 5.2.1.4) are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to null (that is, ‘\0’).

4.2.2 TEXT BATCH RECENT CHANGE MESSAGES

This section specifies the interface messages for the remote processor to access the text batch RC capabilities. The requests will be processed and the RCs will be batched and stored in the batch files identified by clerk IDs. The RCs will be released and applied to the switch data base on demand or at the preset release date and time.

After the batched RCs are applied to the switch data base, they become historical records. The results (success or failure messages) are stored and can be reviewed via Recent Change Batch History capabilities messages (see “BATCH CRAFT SHELL MESSAGES”, Section 4.2.3).

The batch input mode allows users to create a batch file of time release (TIMEREL) type or of DEMAND type. The DEMAND type batch file can be released on demand via “MODE=BMR,CLERK=clerkname” command. The TIMEREL type batch file can be released at the preset release date and time, and can also be released on demand (that is, the preset release date and time can be overridden via demand command).

The new RCs can be added to the TIMEREL type batch file. The existing RCs can be deleted from the TIMEREL type batch file before they are applied to the switch data base. The erroneous RCs in the TIMEREL type batch file can be replaced by the correct RCs.

4.2.2.1 DEMAND Type Batch Create Request Message

To create an RC DEMAND type batch file, the remote processor sends the network a DEMAND Type Batch Create Request Message (see "DEMAND Type Batch Create Request Message", Section 5.2.2.1). This message has the opcode field (see "OPCODE", Section 6.1) set to RCMRC and the request type set (see "REQUEST TYPE", Section 6.2) to RCMRCTXT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "DEMAND Type Batch Create Request Message", Section 5.2.2.1. The batch mode input line "MODE=BMI,DEMAND,CLERK=clerkid<cr>" is mandatory. It specifies the type of batch mode input, DEMAND (see "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2), and the batch file name, clerkid (see "CLERK IDENTIFICATION OR CLERK NAME", Section 6.10). The batch mode input line is followed by one or more RC forms.

Each form action line contains the following major subfields: view, action, attribute name and value pairs (see "VIEW", Section 6.5, "ACTION", Section 6.6, and "ATTRIBUTE NAME AND VALUE PAIRS", Section 6.7). Commas are used to separate the attribute name and value pairs. Each form action line is responsible for one action (change, delete, or insert) upon the specified attributes. Each form action line begins with a keyword FORM and terminates with the action followed by a carriage return.

To change, delete, or insert data in the switch data base, the view field is set to an appropriate value (see "VIEW", Section 6.5). The action field (see "ACTION", Section 6.6) is set to CHG for changing attribute values in the data base, set to OUT for deleting attributes from the data base, and set to NEW for inserting attributes into the data base. Each attribute name and value pair is set to appropriate value.

Errors generated when batched form entries are released to the switch data base are stored in the clerk file. They may be reviewed via the Batch History request (see "Batch History for a Clerk Request Message", Section 4.2.3.1).

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.2 TIMEREL Type Batch Create Request Message

To create an RC TIMEREL type batch file, the remote processor sends the network a TIMEREL Type Batch Create Request Message (see "TIMEREL Type Batch Create Request Message", Section 5.2.2.2). This message has the opcode field (see "OPCODE", Section 6.1) set to RCMRC and the request type set (see "REQUEST TYPE", Section 6.2) to RCMRCTXT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "TIMEREL Type Batch Create Request Message", Section 5.2.2.2. The batch mode input line "MODE=BMI,TIMEREL,CLERK=clerkid<cr>" is mandatory. It specifies the type of batch mode input, TIMEREL (see "GENERAL FORMAT FOR MESSAGE DATA",

Section 5.1.2), and the batch file name, clerkid (see “CLERK IDENTIFICATION OR CLERK NAME”, Section 6.10). The batch mode input line must be followed by an insertion of service order form BV1 (or RC_SRVOR in view name). The BV1 form is followed by one or more RC forms that appear to be normal RC requests. This service order information is used by RC to tag subsequent RCs and correlate them to release time and date.

The service order information in the BV1 form consists of the following attributes: an order number (see “SERVICE ORDER NUMBER”, Section 6.11), an item number (see “ITEM NUMBER”, Section 6.12), a message number (see “MESSAGE NUMBER”, Section 6.13), and the release date and release time (see “RELEASE DATE”, Section 6.14, and “RELEASE TIME”, Section 6.15) that the RC is to be released. The order number indicates the actual service order. The item number is a sequential number representing an individual request within a service order. The message number is used when there are several dependent RCs within an item; otherwise, a zero should be input. If one RC depends upon the previous one having succeeded, it uses the same item number and service order number, but must be assigned a larger message number, which is automatically incremented by 10 (the current default increment). The date and time indicate when the RCs should be released and installed in the *5ESS*-2000 switch data base. Time is expressed as military time of hours and minutes (0000 to 2359). The date includes the month, day, and year.

If the message number of the BV1 form is greater than zero (that is, the subsequent form entries will be dependent) and any form entry fails during application to the switch data base, then all form entries containing the same service order number, item number but a larger message number after that will be automatically withheld. Errors generated when batched form entries are applied to the switch data base are stored in the clerk file. They can be reviewed via the Batch History request (see “Batch History for a Clerk Request Message”, Section 4.2.3.1).

Should any RC form entry fail field check (level 1 check) or cross-field check (level 2 check), then that RC will not be placed in the batch file identified by clerk identifier and the message number will not be incremented if the RC form entries are dependent. The missing RC can be added later by inserting a BV1 service order form that has the same order number and item number, and a message number to indicate placement between existing batched RCs. The RC form entries made subsequent to the insertion of a BV1 service order form that has a different order number and/or item number are not dependent.

Each form action line contains the following major subfields: view, action, attribute name and value pairs (see “VIEW”, Section 6.5, “ACTION”, Section 6.6, and “ATTRIBUTE NAME AND VALUE PAIRS”, Section 6.7). Commas are used to separate the attribute name and value pairs. Each form action line is responsible for one action (for example, change, delete, or insert) upon the specified attributes. Each form action line begins with a keyword FORM and terminated with the action followed by a carriage return.

To change, delete, or insert data in the switch data base, the view field is set to an appropriate value (see “VIEW”, Section 6.5). The action field (see “ACTION”, Section 6.6) is set to CHG for changing attribute values in the data base, set to OUT for deleting attributes from the data base and set to NEW for inserting attributes into the data base. Each attribute name and value pair is set to appropriate value.

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.3 Batch Add Request Message

To add RCs to an existing TIMEREL type batch file, the remote processor sends the network a Batch Add Request Message (see "Batch Add Request Message", Section 5.2.2.3). This message has the opcode field (see "OPCODE", Section 6.1) set to RCMRC and the request type set (see "REQUEST TYPE", Section 6.2) to RCMRCTXT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The formats of the message data are described in "Batch Add Request Message", Section 5.2.2.3. The first format for the message consists of a MODE=BMI input line, a BV1 form line, and one or more of form input lines. The second format is another example of adding RCs to an existing batch file. It is the recurrence of a group of input lines consisting of BV1 form and RC form input lines.

The batch mode input line "MODE=BMI,TIMEREL,CLERK=clerkid<cr>" is mandatory. It specifies the type of batch mode input, TIMEREL (see "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2), and the batch file name, clerkid (see "CLERK IDENTIFICATION OR CLERK NAME", Section 6.10). The batch mode input line must be followed by an insertion of form BV1 (Service Order). The BV1 form is followed by one or more RC forms. This service order information is used by RC to tag subsequent RCs and correlate them to release time and date.

The first format is identical to the batch create format. However, care must be exercised to ensure that the service order information specified in the Batch Add Request Message must not conflict with ones that already exist.

The RC form entries made subsequent to the insertion of a BV1 service order form that has a different order number and/or item number are not dependent. All RCs following a BV1 insert will form a list of logically dependent RCs if the message number for given service order number and item number is greater than zero.

The service order information in the BV1 form consists of the following attributes: an order number (see "SERVICE ORDER NUMBER", Section 6.11), an item number (see "ITEM NUMBER", Section 6.12), a message number (see "MESSAGE NUMBER", Section 6.13), and the release date and release time (see "RELEASE DATE", Section 6.14, and "RELEASE TIME", Section 6.15) that the RC is to be released. The order number indicates the actual service order. The item number is a sequential number representing an individual request within a service order. The message number is used when there are several dependent RCs within an item; otherwise, a zero should be input. If one RC depends upon the previous one having succeeded, it uses the same item number, but must be assigned a larger message number, which is automatically incremented by 10 (the current default increment). The date and time indicate when the RCs should be released and installed in the *5ESS*-2000 switch data base. Time is expressed as military time of hours and minutes (0000 to 2359). The date includes the month, day, and year.

Each form action line contains the following major subfields: view, action, attribute name, and value pairs (see "VIEW", Section 6.5, "ACTION", Section 6.6, and "ATTRIBUTE NAME AND VALUE PAIRS", Section 6.7). Commas are used to separate the attribute name and value pairs. Each form action line is responsible for one action

(for example, change, delete, or insert) upon the specified attributes. Each form action line begins with a keyword FORM and terminates with the action followed by a carriage return.

To change, delete, or insert data in the switch data base, the view field is set to an appropriate value (see “VIEW”, Section 6.5). The action field (see “ACTION”, Section 6.6) is set to CHG for changing attribute values in the data base, set to OUT for deleting attributes from the data base, and set to NEW for inserting attributes into the data base. Each attribute name and value pair is set to appropriate value.

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.4 Batch Delete Request Message

To delete RCs from an existing TIMEREL type batch file, the remote processor sends the network a Batch Delete Request Message (see “Batch Delete Request Message”, Section 5.2.2.4). This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMRC and the request type set (see “REQUEST TYPE”, Section 6.2) to RCMRCTXT. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch Delete Request Message”, Section 5.2.2.4. The batch mode input line “MODE=BMI,TIMEREL,CLERK=clerkid<cr>” is mandatory. It specifies the type of batch mode input, TIMEREL (see “GENERAL FORMAT FOR MESSAGE DATA”, Section 5.1.2), and the batch file name, clerkid (see “CLERK IDENTIFICATION OR CLERK NAME”, Section 6.10). The batch mode input line is followed by one or more BV1 form (Service Order) action lines with the action field set to “OUT.”

The service order information in the BV1 form consists of the following key attributes: an order number (see “SERVICE ORDER NUMBER”, Section 6.11), an item number (see “ITEM NUMBER”, Section 6.12), and a message number (see “MESSAGE NUMBER”, Section 6.13). Commas are used to separate the attribute name and value pairs.

The RCs corresponding to the keys, which are made up of order number, item number, and message number and are set to appropriate values, will be deleted from the batch file named clerkid. Care must be taken to ensure that the service order to be deleted exists.

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.5 Batch Replacement Request Message

To replace RCs in an existing TIMEREL type batch file, the remote processor sends the network a Batch Replacement Request Message (see “Batch Replacement Request Message”, Section 5.2.2.5). For a batch replacement, it is necessary to first delete the existing RC form from the batch file, and then to add the changed version of the RC form in its place.

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMRC and the request type set (see “REQUEST TYPE”, Section 6.2) to RCMRCTXT. The sequence

number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch Replacement Request Message”, Section 5.2.2.5. The batch mode input line “MODE=BMI,TIMEREL,CLERK=clerkid<cr>” is mandatory. It specifies the type of batch mode input, TIMEREL (see “GENERAL FORMAT FOR MESSAGE DATA”, Section 5.1.2), and the batch file name, clerkid (see “CLERK IDENTIFICATION OR CLERK NAME”, Section 6.10). The batch mode input line is followed by form action lines.

The service order information in the BV1 form consists of the following key attributes: an order number (see “SERVICE ORDER NUMBER”, Section 6.11), an item number (see “ITEM NUMBER”, Section 6.12), and a message number (see “MESSAGE NUMBER”, Section 6.13). Commas are used to separate the attribute name and value pairs. The values for the key attributes (ordno, itno, and msgno) are set appropriately.

The BV1 form deletion (that is, OUT) input line initiates the deletion of RC associated with the service order specified by the key (made up of ordno, itno, and msgno). The BV1 form (that is, NEW) insertion input line initiates the insertion of the RCs, which are specified by the subsequent form action lines, and which will be tagged with the ordno, itno, and msgno listed in this BV1 insertion form. Since this message is for a replacement, the key values in a pair of BV1 deletion form and BV1 insertion form must be the same.

Following the two BV1 forms, the new version of the RC is listed. This version will effectively replace the version that was deleted. Dependent RCs (message number > 0) must have the same release date (rdate) and release time (rtime).

The format shown in “Batch Replacement Request Message”, Section 5.2.2.5, illustrates that more than one replacement can be performed in a request message. Care must be taken to ensure that the service order to be replaced already exists.

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.6 Batch Change Time and Date Request Message

To change the release time and date for RCs in an existing TIMEREL type batch file, the remote processor sends the network a Batch Change Time and Date Request Message (see “Batch Change Time and Date Request Message”, Section 5.2.2.6).

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMRC and the request type set (see “REQUEST TYPE”, Section 6.2) to RCMRCTXT. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch Change Time and Date Request Message”, Section 5.2.2.6. The message data is comprised of one or more BV1 form lines. The service order information in the BV1 form consists of the following attributes: an order number (see “SERVICE ORDER NUMBER”, Section 6.11), an item number (see “ITEM NUMBER”, Section 6.12), a message number (see “MESSAGE NUMBER”, Section 6.13), and the release date and release time (see “RELEASE DATE”, Section 6.14, and “RELEASE TIME”, Section 6.15). Commas are

used to separate the attribute name and value pairs. The attribute values for the individual attributes (ordno, itno, msgno, rdate, and rtime) are set appropriately.

The RCs associated with the keys (made up of ordno, itno, and msgno) will have their release schedule updated to the new release date (rdate) and time (rtime). Care must be taken to ensure that the service orders to be changed exist. Dependent RCs (message number > 0) must have the same release date (rdate) and release time (rtime).

With the exception of the carriage return line delimiter, all characters are ASCII non-control printable characters.

4.2.2.7 Text Batch Recent Change Response Messages

In response to the remote processor text batch RC request, the network sends Text Batch Recent Change Response Message(s) (see “Text Batch Recent Change Response Message”, Section 5.2.2.7) to the remote processor. The network will respond with a message for each RC input line in the request message.

The general format for a Text Batch Recent Change Response Message is described in “Text Batch Recent Change Response Message”, Section 5.2.2.7. If the header of the request message is valid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the text batch RC request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN.

The message data takes one of three formats described in “Text Batch Recent Change Response Message”, Section 5.2.2.7. The message data contains the following major elements:

- Message number (msg.#) indicates the #th line in the request message containing the various commands (for example, MODE=BMI..., FORM...).
- The RC output represents the result returned from RC pertaining to the corresponding input line in the request message.
  - — If no errors are found in the RC input line, associated with this response message, the returned result will be “OK” or “OK” plus message text. That is, the message data takes format 1 or format 2
  - — If syntax, domain check, or cross-field check errors are found in the RC input line, associated with this response message, the error messages will be sent to the remote processor. Each error will have the attribute name prepended to it where appropriate. That is, the message data takes format 3. (The data base related errors will not be found until the time the batch job is activated and applied to the switch data base.)

Table 4.2-1 lists the message data in individual response messages corresponding to various RC input lines that may appear in a text batch RC request. See the explanation in “GENERAL FORMAT FOR MESSAGE DATA”, Section 5.1.2, for the RC input lines and the notation.

Table 4.2-1 — Message Data for Text Batch Recent Change Response Messages

LINE #	RC INPUT LINE (MML FORMAT)[a]	MESSAGE DATA IN RESPONSE MSG.
	MODE=BMI,DEMAND	
(1a)	FORM=bVc&d[,e,...,e],d	msg.#<nl>OK
(1b)	FORM=viewname&d[,e,...,e],d	msg.#<nl>OK
	MODE=BMI,TIMEREL	
(1a)	FORM=bVc&d[,e,...,e],d	msg.#<nl>OK ORDNO=ordno ITNO=itno MSGNO=msgno
(1b)	FORM=viewname&d[,e,...,e],d	msg.#<nl>OK ORDNO=ordno ITNO=itno MSGNO=msgno
(2a)	MODE=BMI,DEMAND,CLERK=f	msg.#<nl>OK
(2b)	MODE=BMI,TIMEREL,CLERK=f	msg.#<nl>OK
(3)	MODE=IM	msg.#<nl>OK
(4)	MODE=BMR,CLERK=f	msg.#<nl>OK
(5)	VFYMODE	msg.#<nl>OK IM msg.#<nl>OK BMI TIMEREL CLERK=f msg.#<nl>OK BMI DEMAND CLERK=f
(6)	LASTSO	msg.#<nl>OK ORDNO=ordno ITNO=itno MSGNO=msgno
(7)	NEXTSO	msg.#<nl>OK ORDNO=ordno ITNO=itno MSGNO=msgno

Note(s):
a. The input lines “LASTSO” and “NEXTSO” are applicable only to MODE=BMI,TIMEREL type.

In addition, the network sends one more response message (n+1th response message, assuming that total number of RC input lines in the request message is n) to indicate the finish of the request process. The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Recent Change Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to null (that is, ‘\0’).

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the text batch RC request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set

to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the text batch RC request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.3 BATCH CRAFT SHELL MESSAGES

The batch input feature for the *5ESS*-2000 switch allows RC changes to be input and stored. These RCs can be released on demand or at a preset date and time. Once released, RCs are kept as history records until the system performs batch file administration processing that removes them.

The history (for example, complete, error, pending) of batched RCs can be reviewed via the Batch History Messages. The historical activity of RCs entered into the delayed release clerk files can be reported in a summary or detail format for a specified clerk or for all clerks.

The DEMAND type batch RCs can be released on demand when the network receives the BATCH RELEASE craft shell message from the remote processor. The DEMAND type batch RCs can be removed on demand when the network receives the BATCH REMOVE craft shell message from the remote processor.

This section specifies the interface messages for the remote processor to access the RC batch history capabilities and to release/remove RC DEMAND type batch file capabilities. There are two types of batch history messages: (1) batch information for a given clerk, and (2) batch information for all existing clerks.

4.2.3.1 Batch History for a Clerk Request Message

To obtain an RC history report for a given clerk, the remote processor sends the network a Batch History for a Clerk Request Message (see “Batch History for a Clerk Request Message”, Section 5.2.3.1).

This request results in a history report of selected RCs entered by specified clerk into the delayed release clerk files. It generates a summary or detailed report specific to the RC clerk who entered the delayed release RCs.

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMCSH and the request type (see “REQUEST TYPE”, Section 6.2) set to RCMCFTMSG. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch History for a Clerk Request Message”, Section 5.2.3.1. The input line in the message data begins with the keyword (or command) REPT:RCHIST (report for RC History), which is followed by the keyword CLERK. The parameter value for clerkname (see “CLERK IDENTIFICATION OR CLERK NAME”, Section 6.10) must be replaced by an actual

value that represents a batch history report and also represents the batch file name, where the RCs were entered and stored.

The selection parameter for the batch history report format is optional and can be selected as a “SUMMARY” or “DETAIL” report. The default is a “SUMMARY” report.

The selection parameter for RC status is optional (defaults are COMPLETE and ERROR) and can be set to one or more of the following:

- PENDING - select only the pending RC records for reporting.
- COMPLETE - select only the complete RC records for reporting.
- ERROR - select only the RC records with error for reporting.
- DEMAND - select only the demand RC records (that is, batch mode input of DEMAND type RCs) for reporting.
- ALL - select all RC records for reporting. These RCs could have the following status: PENDING, COMPLETE, ERROR, or DEMAND.

The selection parameter TIME for the range of release time is optional. The TIME parameter selects the RC records that were/are to be released between the range of time as specified in the format of mmddyyhhmm, where mm (01-12) is month, dd (01-31) is day, yy (last two digits of calendar year) is year, hh (00-23) is hour, and mm (00-59) is minute. Leading zeros must be specified. The first time field must specify a time earlier than the second time field. Choosing 0101700000 as the first time field and 1231692359 as the second field would select all records, and this range of time (selecting all records) is default when no time is specified.

4.2.3.2 Batch History for All Clerks Request Message

To obtain an RC history report for all clerks, the remote processor sends the network a Batch History for All Clerks Request Message (see “Batch History for All Clerks Request Message”, Section 5.2.3.2).

This request results in a history report for the activity of all RCs entered by all clerks into the delayed release clerk files. It generates a report, listing the activity of RCs for every clerk who has entered RCs in the delayed release clerk files. This report provides a summary listing of all clerk files in the system, and it details by status the count of all RCs in the system.

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMCSH and the request type (see “REQUEST TYPE”, Section 6.2) set to RCMCFTMSG. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch History for All Clerks Request Message”, Section 5.2.3.2. The input line in the message data begins with the keyword (or command) REPT:RCHIST (report for RC History), which is followed by the keyword ACTIVITY and terminated with <cr> (carriage return).

This report provides a summary listing of all clerk files, and it details by status the count of all RCs.

4.2.3.3 Batch Remove/Release Request Message

To remove/release an RC DEMAND type batch file, the remote processor sends the network a Batch Remove/Release Request Message. (See “Batch Remove/Release Request Message”, Section 5.2.3.3.)

The batch remove request message results in the removal of an RC delayed release clerk file. The batch release request message results in the release of an RC delayed release clerk file.

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMCSH and the request type (see “REQUEST TYPE”, Section 6.2) set to RCMCFTMSG. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The format of the message data is described in “Batch History for All Clerks Request Message”, Section 5.2.3.2. The input line in the message data begins with the keyword (or command) EXC:RCRLS (RC Release) or EXC:RCRMV (RC Remove), which is followed by CLERK=clerkid and terminated with <cr> (carriage return).

4.2.3.4 Batch History Response Messages

In response to the remote processor batch history request, the network sends Batch History Response Message(s) (See “Batch Remove/Release Request Message”, Section 5.2.3.3) to the remote processor.

If the header of the request message is valid and no error is encountered on the input line of the batch history request message (See “Batch History for a Clerk Request Message”, Section 5.2.3.1, or “Batch History for All Clerks Request Message”, Section 5.2.3.2), the network sends a Batch History Response Message (see “Batch Remove/Release Request Message”, Section 5.2.3.3) with the reply field (see “REPLY”, Section 6.8) set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the batch history request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN (that is, continuation flag on).

The message data (see “Batch Remove/Release Request Message”, Section 5.2.3.3) consists of msg.# (message number), control character <nl> (new line) and batch history report. The msg.# (message number) is an integer indicating the corresponding input line in the request. The <nl> (new line) character separates msg.# from batch history report.

A batch history report consists of one or more pages. Each page is delimited with an “end of message” () character, which is represented in ASCII code as Octal 031. Each page consists of one or more lines. Each line is delimited by a carriage return (<cr>).

The network will not attempt to break up batch history output on line or page boundaries when the size of the batch history report is very large and the report is to be delivered by more than one response messages. Since the number of pages in the output is not returned to the remote processor and there is no delimiter that depicts the end of the output, the remote processor will need to rely on the number of characters read from the lower layers.

If the header of the request message is valid and an error is encountered on an input line of a batch history request (for example, Batch History for a Clerk Request Message or Batch History for All Clerk Request Messages) (See “Batch History for a Clerk Request Message”, Section 5.2.3.1, and “Batch History for All Clerks Request Message”, Section 5.2.3.2), the network sends a Batch History Response Message (see “Batch Remove/Release Request Message”, Section 5.2.3.3) with the reply field (see “REPLY”, Section 6.8) set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the batch history request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN. The message data (see “Recent Change Response Message”, Section 5.2.1.2) is populated with an error code (see “ERROR CODES FOR REPLY OPCODE RCMRCERROR”, Section 6.25.2), msg.# (message number), <nl> (new line), error input and an error message, which is an ASCII string.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

In response to a batch history request (for example, Batch History for a Clerk Request Message or Batch History for All Clerk Request Messages) (See “Batch History for a Clerk Request Message”, Section 5.2.3.1, and “Batch History for All Clerks Request Message”, Section 5.2.3.2), from the remote processor, the network sends Batch History Response Message(s) (see “Batch Remove/Release Request Message”, Section 5.2.3.3) to the remote processor. The network will respond with a message for each batch history input line in the request message. In addition, the network sends one more response message (n+1th response message, assuming that total number of input lines in the request message is n) to indicate the finish of the request process.

The information fields for Finish Request Process Response Message (see “Finish Request Process Response Message”, Section 5.2.1.4) are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the batch history request message. The continuation flag field (see “CONTINUATION FLAG”,

Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to null (that is, '\0').

4.2.3.5 Batch Remove/Release Response Messages

In response to the remote processor batch remove/release request, the network sends Batch REMOVE/RELEASE Response Message(s) (see "Batch Remove/Release Response Message", Section 5.2.3.5) to the remote processor.

If the header of the request message is valid and no error is encountered on the input line of the batch remove/release request message (see "Batch Remove/Release Request Message", Section 5.2.3.3), the network sends a Batch Remove/Release Response Message (see "Batch Remove/Release Response Message", Section 5.2.3.5) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) set to RCMRCOK. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as those in the batch history request message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN (that is, continuation flag on).

The message data (see "Batch Remove/Release Request Message", Section 5.2.3.3) consists of msg.# (message number), control character <nl> (new line) and the ASCII string "clerkid removed" or "clerkid release". The msg.# (message number) is an integer indicating the corresponding input line in the request. The <nl> (new line) character separates msg.# from the ASCII string.

If the header of the request message is valid and an error is encountered on an input line of a batch remove/release request (see "Batch Remove/Release Request Message", Section 5.2.3.3), the network sends a Batch Remove/Release Response Message (see "Batch Remove/Release Response Message", Section 5.2.3.5) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) to RCMRCERROR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the batch history request message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN. The message data (see "Recent Change Response Message", Section 5.2.1.2) is populated with an error code (see "ERROR CODES FOR REPLY OPCODE RCMRCERROR", Section 6.25.2) msg.# (message number), <nl> (new line), error input and an error message, which is an ASCII string.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQFAIL. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as those in the request message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT. The message data contains an error code (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE

NUMBER", Section 6.3) have the same values as those in the request message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT. The message data contains an error code (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) whose value equals RCMREQINV.

In response to a batch remove/release request (see "Batch Remove/Release Request Message", Section 5.2.3.3), from the remote processor, the network sends Batch Remove/Release Response Message(s) (see "Batch Remove/Release Response Message", Section 5.2.3.5) to the remote processor. In addition, the network sends one more response message (n+1th response message, assuming that total number of input lines in the request message is n) to indicate the finish of the request process.

The information fields for Finish Request Process Response Message (see "Finish Request Process Response Message", Section 5.2.1.4) are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCMRCOK. The opcode, request type and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as those in the batch history request message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to null (that is, '\0').

4.2.4 OFFICE RECORDS, ACSR RC, AND ASCII FILE RETRIEVAL MESSAGES

This section specifies the interface messages for the remote processor to access the OFR, ACSR RC, and ASCII file retrieval capabilities.

The remote processor sends Office Records Processing Request Messages and/or Office Records Output Request Messages to the network to access OFR capabilities. If an OFR Processing Request Message is to request the printing of OFR form(s), this request is assigned a unique request identification (ID) number and scheduled by the network, and the request ID number is sent back to the remote processor. After the completion of OFR processing, the remote processor sends OFR Output Request Message to retrieve output, that is, the result created during OFR processing, by specifying a request ID. An OFR request consists of one or more OFR input lines ("GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2). Each input line of OFR Processing Request Message contains one input message (IM) command. Each input line of OFR Output Request Message contains one request ID. In response to the remote processor request, the network sends OFR Processing Response Message(s) (see "Office Records Processing Response Message", Section 5.2.4.2) or OFR Output Response Message(s) (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4) to the remote processor.

To retrieve ACSR RC data file, the remote processor sends OFR/ACSR RC Output Request Message to the network by specifying the request ID that designates the current or previous ACSR RC activity files. In response to the request, the network sends OFR/ACSR RC Output Response Message, containing ACSR RC data, to the remote processor.

To retrieve an ASCII file on the *5ESS*-2000 switch, the remote processor sends an ASCII File Retrieval Request Message to the network by specifying the full path of the directory that contains the file and the file name of the file to be retrieved.

In response to the request, the network sends an ASCII File Retrieval Response Message, containing the requested ASCII file, to the remote processor.

Each input line in the request message from the remote processor results in at least one response message from the network. In addition, the network sends one more response message (that is, Finish Request Process Response Message) to indicate the finish of the request process.

To send multiple response messages in response to one single request message, the continuation flag (see "CONTINUATION FLAG", Section 6.4) is set on until the last response message is sent.

4.2.4.1 Office Records Processing Request Message

The remote processor sends an OFR Processing Request Message (see "Office Records Processing Request Message", Section 5.2.4.1) to the network for accessing OFR capabilities, such as:

- Print OFR form(s) by form type or by category

  For example: to obtain trunk-related OFR forms, TRUNK category or 5200 series trunk form must be specified.

- List output requests/status/parameters
- Input/change OFR operating and scheduling parameters
- Cancel an OFR print request.

This message has the opcode field (see "OPCODE", Section 6.1) set to RCMCSH and the request type set (see "REQUEST TYPE", Section 6.2) to RCMCFTMSG. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "Office Records Processing Request Message", Section 5.2.4.1. The message data consists of one or more OFR input lines. Each OFR input line contains an IM command and delimited by a control character "carriage return" (<cr>). In the following, the message ID enclosed in the double quotes is used to represent an IM command whose format is described in "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2, and detailed in document 235-600-700, **5ESS**-*2000 Switch Input Messages*.

To request the printing of all categories or one category of OFR forms, the input line consists of the "OP:OFR-CAT" input message command. To request the printing of an OFR form, the input line consists of the "OP:OFR-FORM" input message command. To list output requests/status/parameters, the input line consists of the "OP:OFR-STATUS" input message command. To input or change the internal operating and scheduling parameters of OFR control process, which affect the print requests of the OP:OFR input message commands, the input line consists of the "IN:OFR-PARM" input message command. To cancel a job invoked by a printing command ("OP:OFR-CAT" or "OP:OFR-FORM" input message command), the input line consists of the "STP:OFR" input message command.

In all cases of printing requests (OP:OFR-CAT and OP:OFR-FORM), it is important that the device parameter is specified as "FILE" in order to retrieve output later by using OFR Output Request Message.

4.2.4.2 Office Records Processing Response Message

In response to the remote processor OFR processing request, the network sends OFR Processing Response Message(s) (see "Office Records Processing Response Message", Section 5.2.4.2) to the remote processor. Each input line in the request message from the remote processor generates an OFR Processing Response Message.

If the header of the request message is valid and no error is encountered on an input line of OFR Processing Request Message (see "Office Records Processing Request Message", Section 5.2.4.1), the network sends an OFR Processing Response Message (see "Office Records Processing Response Message", Section 5.2.4.2) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) set to RCMRCOK. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the OFR Processing Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN (that is, continuation flag on). The message data format is described in "Office Records Processing Response Message", Section 5.2.4.2. The msg.# (message number) is an integer indicating the corresponding input line in the request. The <nl> (new line) character separates msg.# from system response that indicates the response to the input message command specified in the OFR input line of the request message. The documents 235-600-700, **5ESS**-*2000 Switch Input Messages*, and 235-600-750, **5ESS**-*2000 Switch Output Messages*, detail system responses to individual IMs.

If the header of the request message is valid and an error is encountered on an input line of OFR Processing Request Message (see "Office Records Processing Request Message", Section 5.2.4.1), the network sends an OFR Processing Response Message (see "Office Records Processing Response Message", Section 5.2.4.2) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) to RCMRCERROR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the OFR Processing Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN. The message data (see "Office Records Processing Response Message", Section 5.2.4.2) is populated with an error code (see "ERROR CODES FOR REPLY OPCODE RCMRCERROR", Section 6.25.2), msg.# (message number), <nl> (new line), error input and an error message, which is an ASCII string.

If the header of the request message and the input line are both valid and an error is encountered during processing, the network sends an OFR Processing Response Message (see "Office Records Processing Response Message", Section 5.2.4.2) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field ("REPLY OPCODE", Section 6.9) to RCMRCRTRERR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the OFR Processing Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN. The message data (see "Office Records Processing Response Message", Section 5.2.4.2) is populated with an error code (see "ERROR CODES FOR REPLY OPCODE RCMRCRTRERR", Section 6.25.5), msg.# (message number), <nl> (new line), input line, and an error message, which is an ASCII string.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the OFR Processing Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the OFR Processing Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

In response to the OFR processing request from the remote processor, the network sends OFR Processing Response Message(s) (see “Office Records Processing Response Message”, Section 5.2.4.2) to the remote processor. The network will respond with a message for each OFR input line in the request message. In addition, the network sends one more response messages (n+1th response message, assuming that total number of OFR input lines in the request message is n) to indicate the finish of the request process.

The information fields for Finish Request Process Response Message (see “Finish Request Process Response Message”, Section 5.2.1.4) are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC, and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the OFR Processing Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to null (that is, ‘\0’).

4.2.4.3 Office Records, ACSR RC Output and ASCII File Retrieval Request Message

4.2.4.3.1 Office Records/ACSR RC Output Request Message

For each OFR printing command “OP:OFR-CAT” or “OP:OFR-FORM”, which is contained in the OFR Processing Request Message, a unique request identification number is returned to the remote processor at request. Later, an output file and/or an error file are created during OFR processing. These files can be retrieved only if the device parameter was specified as “FILE” in the previous OFR Processing Request Message.

An output file is a data file that contains the actual office records data. An error file contains any errors that may have occurred during processing. If no errors occurred during processing, then the message stating that no errors occurred is written into the error file.

To retrieve the output of an OFR printing command that requests the printing of switch office data, the remote processor sends an OFR Output Request Message (see

"Office Records/ACSR RC Output Request Message", Section 5.2.4.3) to the network. The data fields are set as follows: This message has the opcode field (see "OPCODE", Section 6.1) set to RCMOFR and the request type set (see "REQUEST TYPE", Section 6.2) to RCMOFROUT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "Office Records/ACSR RC Output Request Message", Section 5.2.4.3. The message data consists of one or more OFR input lines. Each OFR input line contains one name/value pair of request identification number (see "REQUEST IDENTIFICATION NUMBER", Section 6.16) followed by two name/value pairs of starting positions, one for the output file (see "OUTPUT BYTES", Section 6.17) and one for the error file (see "ERROR BYTES", Section 6.18). The delimiter for each OFR input line is the "carriage return" (<cr>) control character.

To retrieve the ACSR RC data file directly, the remote processor sends an OFR/ACSR RC Output Request Message (see "Office Records/ACSR RC Output Request Message", Section 5.2.4.3) to the network. The data fields are set as follows: This message has the opcode field (see "OPCODE", Section 6.1) set to RCMACSR and the request type set (see "REQUEST TYPE", Section 6.2) to RCMACSROUT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "Office Records/ACSR RC Output Request Message", Section 5.2.4.3. The message data consists of one or more input lines. Each input line contains one name/value pair of request identification number followed by two name/value pairs of starting positions, one for the output file (see "OUTPUT BYTES", Section 6.17) and one for the error file (see "ERROR BYTES", Section 6.18) The delimiter for each input line is the "carriage return" (<cr>) control character. The request ID number (see "REQUEST IDENTIFICATION NUMBER", Section 6.16), which designates the current or previous ACSR RC activity files, is set to the appropriate value.

4.2.4.3.2 ASCII File Retrieval Request Message

To retrieve an ASCII file on the *5ESS*-2000 switch, the remote processor sends an ASCII File Retrieval Request Message (see "ASCII File Retrieval Request Message", Section 5.2.4.3.1) to the network.

This message has the opcode field (see "OPCODE", Section 6.1) set to RCM5E and the request type set (see "REQUEST TYPE", Section 6.2) to RCMFLOUT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT.

The format of the message data is described in "ASCII File Retrieval Request Message", Section 5.2.4.3.1. The message data consists of DIR=<path>,FILE=<filename> (where <path> is the full path to the directory containing the file to be retrieved, <filename> is the name of the file to be sent back to the remote processor).

The delimiter for the message data is the "carriage return" (<cr>) control character. Multiple files can be returned in one request by continuing the DIR=<path>,FILE=<filename> <cr> scenario.

4.2.4.4 Office Records/ACSR RC Output Response and ASCII File Retrieval Response Message

In response to the remote processor OFR/ACSR RC output or ASCII file Retrieval request, the network sends OFR/ACSR RC Output or ASCII File Retrieval Response Message(s) (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4) to the remote processor. Each input line in the request message from the remote processor generates at least one OFR/ACSR RC Output or ASCII File Retrieval Response Message.

If the header of the OFR/ACSR RC Output or ASCII File Retrieval Request Message (see "Office Records/ACSR RC Output Request Message", Section 5.2.4.3) is valid and no error is encountered on any of the input lines in the request message, the network sends the following series of OFR/ACSR RC Output or ASCII File Retrieval Response Messages (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4):

- For each input line without an error, the four response messages are sent from the network to the remote processor if output file and error file are requested.

  — Output File Response Message - This message conveys the output file that contains the office records data, ACSR RC data, or ASCII data file, depending on the output request message. (See Note 1.)

  — Output File Transmission Complete Response Message - This message indicates the end of transmission for the output file. (See Note 1.)

  — Error File Response Message - This message conveys the error file that contains any error messages generated during the OFR processing or the ACSR RC data retrieval. (See Note 2.)

  — Error File Transmission Complete Response Message - This message indicates the end of transmission for the error file. (See Note 2.)

  Note 1: This message will not be returned if output file is not requested in the previous OFR/ACSR RC Output or ASCII File Retrieval Request Message.

  Note 2: This message will not be returned if error file is not requested in the previous OFR/ACSR RC Output or ASCII File Retrieval Request Message.

- After responding to the last input line of the request message, the following message is sent to the remote processor:

— Finish Request Process Response Message — This message indicates the finish of the OFR/ACSR RC output or ASCII file retrieval request process.

Table 4.2-2 lists the values of the information fields for the previous messages responding to an input line.

Table 4.2-2 — OFR/ACSR RC Output and ASCII File Retrieval Response Message Information Field

INFORMATION FIELD [a,b,c,d]	#1 OUTPUT FILE RESPONSE MSG.	#2 TRANSMISSION COMPLETE RESPONSE MSG.	#3 ERROR FILE RESPONSE MSG.	#4 TRANSMISSION COMPLETE RESPONSE MSG.
Reply	RCMREQSUCC	RCMREQSUCC	RCMREQSUCC	RCMREQSUCC
Reply Opcode	RCMSRVSUCC	RCMSRVSYNC	RCMSRVSUCC	RCMSRVSYNC
Reserved	blanks	blanks	blanks	blanks
Opcode	*	*	*	*
Request Type	*	*	*	*
Sequence No.	*	*	*	*
Continuation Flag	RCMCONTIN	RCMCONTIN	RCMCONTIN	RCMCONTIN
Message Data	msg.#<nl> output file	msg.#<nl> input line<nl> x'mission msg.	msg.#<nl> error file	msg.#<nl> input line<nl> x'mission msg.

Note(s):

a. * - The information field in the response message has the same value as in that field of the request message.

b. msg.# (message number) is an integer indicating the corresponding input line in the request.

c. <nl> - "new line" control character

d. x'mission msg. - It indicates the completion of transmitting an output/error file. For OFR/ACSR RC Output Response message, it is an ASCII string of "THE TRANSMISSION OF THE RC/OS OUTPUT FILE HAS FINISHED FOR REQID: xxx" or of "THE TRANSMISSION OF THE RCOS ERROR FILE HAS FINISHED FOR REQID: xxx", where xxx is the request ID number. For ASCII File Retrieval Response message, it is an ASCII string of "THE TRANSMISSION OF THE RCOS OUTPUT FILE HAS FINISHED".

If the header of the request message is valid and an error is encountered on an input line of OFR/ACSR RC Output or ASCII File Retrieval Request Message (see "Office Records/ACSR RC Output Request Message", Section 5.2.4.3), the network sends an OFR/ACSR RC Output or ASCII File Retrieval Response Message (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) set to RCMSRVFAIL. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the OFR/ACSR RC Output or ASCII File Retrieval Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMCONTIN. The message data (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4) is populated with error code RCMNOERRC (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) msg.# (message number), <nl> (new line), error input, and error message, which is an ASCII string.

If the header of the request message and the input line are both valid and an error is encountered during processing, the network sends an OFR/ACSR RC Output or ASCII File Retrieval Response Message (see "Office Records, ACSR RC, and ASCII File Retrieval Output Response Message", Section 5.2.4.4) with the reply field (see "REPLY", Section 6.8) set to RCMREQSUCC and the reply opcode field (see "REPLY OPCODE", Section 6.9) set to RCMRCRTRERR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the OFR/ACSR RC Output or ASCII File Retrieval Request Message. The continuation

flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN. The message data (see “Office Records, ACSR RC, and ASCII File Retrieval Output Response Message”, Section 5.2.4.4) is populated with error code (see “ERROR CODES FOR REPLY OPCODE RCMRCRTRERR”, Section 6.25.5), msg.# (message number), <nl> (new line), input line, and error message (see “ERROR MESSAGES FOR OFR OUTPUT RESPONSE MESSAGE”, Section 6.26.1), which is an ASCII string

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the OFR/ACSR RC Output or ASCII File Retrieval Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMSRVFAIL. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the OFR/ACSR RC Output or ASCII File Retrieval Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

In response to the OFR/ACSR RC output or ASCII file retrieval request from the remote processor, the network sends OFR/ACSR RC Output or ASCII File Retrieval Response Message(s) (see “Office Records, ACSR RC, and ASCII File Retrieval Output Response Message”, Section 5.2.4.4) to the remote processor. The network will respond with a message for each input line in the request message. In addition, the network sends one more response message (n+1th response message, assuming that total number of input lines in the request message is n) to indicate the finish of the request process.

The information fields for Finish Request Process Response Message (see “Finish Request Process Response Message”, Section 5.2.1.4) are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMSRVSUCC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the OFR/ACSR RC Output or ASCII File Retrieval Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT that is, continuation flag off). The message data contains an ASCII string “THE PROCESSING OF THE REQUEST(S) HAS BEEN COMPLETED.”

4.2.5 INTERFACE TERMINATION MESSAGES

This section specifies the interface messages for the remote processor to terminate any text RC activities or any craft shell activities (for example, batch history or OFR processing).

The remote processor sends Text Recent Change Interface Termination Request Message to terminate the text RC interface channel. The remote processor sends Craft Shell Interface Termination Request Message to terminate the craft shell interface channel.

During a session, both interface channels can be active in the same session simultaneously. The interface termination request will release the network resources, but not terminate the current session.

4.2.5.1 Text Recent Change Interface Termination Request Message

To terminate any text (immediate/batch) RC activities, the remote processor sends the network a Text Recent Change Interface Termination Request Message (see "Text Recent Change Interface Termination Request Message", Section 5.2.4.5).

Both the text RC (immediate/batch mode) activities and the craft shell activities can coexist during a session. If the remote processor fails to issue a Text Recent Change Interface Termination Request before switching to craft shell activities, the text RC interface channel will remain active until the interface termination request is issued or the session is terminated gracefully or abnormally. If the remote processor fails to issue a Text Recent Change Interface Termination Request before requesting session disconnection or session abort, the network will discard any active interface channels (for example, text RC or craft shell interface channel) and then terminate the session.

This message has the opcode field (see "OPCODE", Section 6.1) set to RCMRC and the request type set (see "REQUEST TYPE", Section 6.2) to RCMTERMTXT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any ASCII character. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT. The message data is set to null ('\0').

4.2.5.2 Text Recent Change Interface Termination Response Message

A Text Recent Change Interface Termination Response Message (see "Text Recent Change Interface Termination Response Message", Section 5.2.4.6) is sent from the network to the remote processor to indicate the termination of text (immediate or batch) RC activities.

If the request is processed successfully, the information fields are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the Text Recent Change Interface Termination Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to the null character '\0'.

If the request is processed unsuccessfully, the information fields are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the Text Recent Change Interface Termination Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data contains an error code (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) whose value equals RCMREQINV.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.5.3 Craft Shell Interface Termination Request Message

To terminate any craft shell activities (for example, batch history, OFR processing), the remote processor sends the network a Craft Shell Interface Termination Request Message (see “Craft Shell Interface Termination Request Message”, Section 5.2.4.7).

Both the text RC (immediate/batch mode) activities and the craft shell activities can coexist during a session. If the remote processor fails to issue a Craft Shell Interface Termination Request before switching to text RC activities, the craft shell interface channel will remain active until the interface termination request is issued or the session is terminated gracefully or abnormally. If the remote processor fails to issue a Craft Shell Interface Termination Request before requesting session disconnection or session abort, the network will discard any active interface channels (for example, text RC or craft shell interface channel) and then terminate the session.

This message has the opcode field (see “OPCODE”, Section 6.1) set to RCMCSH and the request type set (see “REQUEST TYPE”, Section 6.2) to RCMTERMCS. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data is set to null (‘\0’).

4.2.5.4 Craft Shell Interface Termination Response Message

A Craft Shell Interface Termination Response Message (Section 5.2.4.8) is sent from the network to the remote processor to indicate the termination of craft shell activities.

If the request is processed successfully, the information fields are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Craft Shell Interface Termination Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to the null character ‘\0’.

If the request is processed unsuccessfully, the information fields are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Craft Shell Interface Termination Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Craft Shell Interface Termination Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Craft Shell Interface Termination Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.6 RECENT CHANGE VIEW TIME-OUT MESSAGES

The RC system uses time-outs for individual view operations so that the remote processor will not wait forever for an RC request to complete. The time-out value for a view is the length of the time that the network front end will wait for the completion of an RC before it decides that the RC system is having problems and should, therefore, abort the operation.

The remote processor can use these values (plus a constant number of seconds added for overhead) to determine how long to wait for a response from the network before deciding that some problem has been encountered.

4.2.6.1 Recent Change View Time-Out Request Message

A Recent Change View Time-Out Request Message (see “Recent Change View Time-Out Request Message”, Section 5.2.5.1) is sent by the remote processor to the network for requesting the time-out values for RC view operations.

The Recent Change View Time-Out Request Message consists of a message header (opcode, request type, sequence number, continuation flag) and the message data. The opcode field (see “OPCODE”, Section 6.1) is set to RCM5E. The request type field (see “REQUEST TYPE”, Section 6.2) is set to RCM5ETIMERS. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any appropriate ASCII character.

The continuation flag (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT to indicate no continuation. The message data field is set to the null character (that is, ‘\0’).

4.2.6.2 Recent Change View Time-Out Response Message

A Recent Change View Time-Out Response Message (see “Recent Change View Time-Out Response Message”, Section 5.2.5.2) is sent from the network to the remote processor to indicate the time-out values of RC view operations.

If the request is successful, the information fields are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCM5EOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Recent Change View Time-Out Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMCONTIN (that is, continuation flag on) if this message is continued by forthcoming message(s); otherwise, it is set to RCMNOCONT (that is, continuation flag off). The message data consists of the following information:

- Read time: The time (in units of second) to query the switch data base for data verification. Each RC view for verify (VFY) operation takes the same amount of time.
- Default time: The time (in units of second) to update the switch data base, insert data into or delete data from the switch data base. Each RC change view for CHG, NEW, or OUT operation takes the default time unless it is specified in the following description of view name and time-out pairs:
- View name: The name of a RC view [refer to the RC Text Interface document (See “REFERENCES”, Section 7, for the document number) for view names].
- Time-out: The time (in units of second) to update the switch data base, insert data into, or delete data from the switch data base for a given RC view.

Read time, default time, and a pair of view name/time-out are separated by an <ack> character (Octal 6) from each other. View name and time-out are separated by a <del> character (‘=’ if MML format, space ‘ ‘ if PDS format).

If the request fails during processing, the network returns a Recent Change View Time-Out Response Message (see “Recent Change View Time-Out Response Message”, Section 5.2.5.2) with the following fields set to appropriate values. The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCRTRERR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as in those fields of the Recent Change View Time-Out Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “ERROR CODES FOR REPLY OPCODE RCMRCRTRERR”, Section 6.25.5).

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE

NUMBER", Section 6.3) have the same values as in those fields of the Recent Change Time-Out Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT. The message data contains an error code (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCM5EFAIL. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as in those fields of the Recent Change Time-Out Request Message. The continuation flag field (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT. The message data contains an error code (see "MISCELLANEOUS ERROR CODES", Section 6.25.1) whose value equals RCMREQINV.

4.2.7 STATUS CHECK MESSAGES

4.2.7.1 Status Check Request Message

A Status Check Request Message (see "Status Check Request Message", Section 5.2.6.1) is sent by the remote processor to the network, when it is necessary.

A Status Check Message is used to check the status of the network. It is useful in situations where the remote processor times out waiting for a response from the network. In times of traffic congestion or system problems, the network response time to the remote processor requests could be longer than a preset time. The remote processor has the option of checking the status of request (that is, the progress of the request) by sending a Status Check Message.

The Status Check Request Message consists of a message header (opcode, request type, sequence number, continuation flag) and a null message data field. The opcode field (see "OPCODE", Section 6.1) is set to RCM5E. The request type field (see "REQUEST TYPE", Section 6.2) is set to RCM5ESTAT. The sequence number field (see "SEQUENCE NUMBER", Section 6.3) is set to any appropriate ASCII character. The continuation flag (see "CONTINUATION FLAG", Section 6.4) is set to RCMNOCONT to indicate no continuation. The message data is set to the null character (that is, '\0').

4.2.7.2 Status Check Response Message

A Status Check Response Message (see "Status Check Response Message", Section 5.2.6.2) is sent from the network to the remote processor to indicate the status of the network.

If the header of the request message is valid and the network is not processing any request (that is, idle), the network returns a Status Check Response Message (see "Status Check Response Message", Section 5.2.6.2) with the following fields set to appropriate values. The reply field (see "REPLY", Section 6.8) is set to RCMREQSUCC. The reply opcode field (see "REPLY OPCODE", Section 6.9) is set to RCM5EIDLE. The opcode, request type, and sequence number fields (see "OPCODE", Section 6.1, "REQUEST TYPE", Section 6.2, and "SEQUENCE NUMBER", Section 6.3) have the same values as those in the Status Check Request Message. The

continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data is set to the null character ‘\0’.

If the header of the request message is valid and the network is busy in processing a request, the network returns a Status Check Response Message (see “Status Check Response Message”, Section 5.2.6.2) with the following fields set to appropriate values. The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCM5EBUSY. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Status Check Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT.

The message data consists of three fields: subsystem, activity, and state. The subsystem field (see “SUBSYSTEM”, Section 6.22) has the same value as the opcode field (see “OPCODE”, Section 6.1) of the previous request message sent from the remote processor. The activity field (see “ACTIVITY”, Section 6.23) has the same value as the request type field of the previous request message sent from the remote processor. The state field is set to one of the values described in “STATE”, Section 6.24.

If the remote processor times out waiting for the status check response, it may be necessary to send a Session Abort Disconnect Message to disconnect the session and then to try to establish a new session.

If the network returns “idle” status (that is, reply opcode = RCM5EIDLE) and the remote processor expects the network to be working on the previous request, the request message was lost somewhere; the remote processor needs to resend the request.

If the network returns “working” status (that is, reply code = RCM5EBUSY), the remote processor should reset its timer and allow more time for the request to complete.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

4.2.8 ABORT MESSAGES

4.2.8.1 Abort Request Message

An Abort Request Message is sent from the remote processor to the network to abort an activity requested by the remote processor when the remote processor has a problem. Whatever the network is doing at the time will be aborted, and the network will send the reply information remaining in the output buffer to the remote processor.

If an abort request is successful, the interface channel (text RC interface channel or craft shell interface channel) associated with the request activity will be terminated.

Abort Request Message (see “Abort Request Message”, Section 5.2.7.1) consists of a message header and a null message data field. The opcode field (see “OPCODE”, Section 6.1) is set to RCM5E. The request type field (see “REQUEST TYPE”, Section 6.2) is set to RCM5EABORT. The sequence number field (see “SEQUENCE NUMBER”, Section 6.3) is set to any ASCII character. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT to indicate no continuation. The message data is set to the null character (that is, ‘\0’).

4.2.8.2 Abort Response Message

An Abort Response Message is sent by the network to the remote processor in response to the abort request.

If the abort request is successful, an Abort Response Message (see “Abort Response Message”, Section 5.2.7.2) has the message data containing three fields (subsystem, activity, and state). The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCM5EAOK. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Abort Request Message (see “Abort Request Message”, Section 5.2.7.1) The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The subsystem field (see “SUBSYSTEM”, Section 6.22) has the same value as the opcode field in the previous request message. The activity field (see “ACTIVITY”, Section 6.23) has the same value as the request field in the previous request message. The state field is set to one of the values described in “STATE”, Section 6.24.

If the abort request is not successful due to inconsistency with the previous request (see “ERROR CODES FOR REPLY OPCODE RCM5EAERROR”, Section 6.25.3 for failure reason), an Abort Response Message (see “Abort Response Message”, Section 5.2.7.2) sent from the network has the message data containing four fields (subsystem, activity, state, and error code). The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCM5EAERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Abort Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The subsystem field (see “SUBSYSTEM”, Section 6.22) has the same value as the opcode field in the previous request message. The activity field (see “ACTIVITY”, Section 6.23) has the same value as the request field in the previous request message. The state field is set to one of the values described in “STATE”, Section 6.24. The error code field (see “ERROR CODES FOR REPLY OPCODE RCM5EAERROR”, Section 6.25.3) is set to appropriate value.

If the header of the request message is valid and an error is encountered during processing, an Abort Response Message (see “Abort Response Message”, Section 5.2.7.2) sent from the network has the message data containing four fields (subsystem, activity, state, and error code). The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCRTRERR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Abort Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The subsystem field (see “SUBSYSTEM”, Section 6.22) has the same value as the opcode field in the previous request message. The activity field (see “ACTIVITY”, Section 6.23) has the same value as the request field in the previous request message. The state field is set to one of the values described in “STATE”, Section 6.24. The error code field (see “ERROR CODES FOR REPLY OPCODE RCMRCRTRERR”, Section 6.25.5) is set to appropriate values.

If the network is idle and an abort request message is received, the reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC, and the reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCM5EFAIL. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the Abort Request Message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT (that is, continuation flag off). The message data (see “Abort Response Message”, Section 5.2.7.2) contains an error code (see “ERROR CODE”, Section 6.25) set to RCMREQINV.

If the opcode field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQFAIL. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMNOROC. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMOPCINV.

If the request type field of the request message header is invalid, the information fields of the response message are set as follows: The reply field (see “REPLY”, Section 6.8) is set to RCMREQSUCC. The reply opcode field (see “REPLY OPCODE”, Section 6.9) is set to RCMRCERROR. The opcode, request type, and sequence number fields (see “OPCODE”, Section 6.1, “REQUEST TYPE”, Section 6.2, and “SEQUENCE NUMBER”, Section 6.3) have the same values as those in the request message. The continuation flag field (see “CONTINUATION FLAG”, Section 6.4) is set to RCMNOCONT. The message data contains an error code (see “MISCELLANEOUS ERROR CODES”, Section 6.25.1) whose value equals RCMREQINV.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

4.3 RCOS INTERFACE FEATURE CAPABILITIES

4.3.1 RCOS DC/AFT FEATURE

The DC/AFT feature provides a recent change/verify (RC/V) text interface software translation mechanism that allows the service provider to retrofit the *5ESS*®-2000 switch with the current software release while the supporting RC OS remains on the previous software release. The APPTEXT translation mechanism of the DC/AFT feature is based on the RC evolution process used during *5ESS*-2000 switch retrofits. The DC/AFT feature converts RC/V text interface commands generated in the format of one *5ESS*-2000 switch software release into the format of a later *5ESS*-2000 switch software release.

The DC/AFT translator may be used on either a synchronous or asynchronous network connection. Specific information on an asynchronous network connection is provided in document 235-118-xxx, *5ESS*-2000 *Recent Change Procedures*.

The translator is intended for use on a temporary basis after a retrofit and prior to updating the applicable OS software. Lucent Technologies expects the service provider to minimize the time between a retrofit and OS software update to the greatest degree practical. Specifically, Lucent Technologies expects the OS software to be updated within three months of a retrofit when the OS software is not available at the time of the retrofit. If the OS software is available at the time of the retrofit, the OS software should be updated within one month.

4.3.2 RC/V REQUEST MESSAGE

The translator must be invoked at the beginning of each applicable RC/V text interface session. Invocation of the translator may be discontinued by terminating the session. The feature user may change the translator version to another switch software release by terminating and reestablishing the session with the appropriate parameter conditions.

The *5ESS*-2000 switch translator is invoked with session parameter “VERSION=X”. (See Table 4.3-5.) The variable “X” used with this parameter identifies an integer value representing the *5ESS*-2000 switch software release with which the supporting OS was designed to be used. A synchronous session may be established using the message, “1100VERSION=6,FORM=...”. For more information, see “INTERFACE TERMINATION MESSAGES”, Section 4.2.5, on the interface termination messages, “GENERAL FORMAT FOR MESSAGE DATA”, Section 5.1.2, on general format for message data, or “Text Recent Change Interface Termination Request Message”, Section 5.2.4.5, on the text RC interface termination request message.

If no switch version is selected, the translator is not invoked and the RC/V text interface messages are processed by the *5ESS*-2000 switch in the current software release format. This is the expected method of operation when both the OS and *5ESS*-2000 switch software are on concurrent releases.

There are no changes to the request or response messages as a result of this interface capability.

4.3.3 RC VIEWS SUPPORTED

The views that are supported by the AFT feature are summarized in Tables 4.3-1, 4.3-2, 4.3-3, and 4.3-4.

Table 4.3-1 — RC Views Supported from 5E10 OS to 5E11 *5ESS*-2000 Switch

CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 7	CLASS 9	CLASS 10	CLASS 12	CLASS 21	CLASS 22	CLASS 23
1.5V	2.5V	3.3	4.1V	7.3	9.40	10.4	12.5	21.32	22.14V	23.1V
1.6		3.5	4.2	7.4	9.41		12.9V			23.2
1.8			4.14							23.3
1.9			4.24							23.8
1.10			4.39							23.11
1.11										23.16
1.12V										23.20
1.20										23.21
1.21										23.22V
1.22										23.23
1.23										23.30
1.24										23.33
1.25V										23.34
1.26										
1.27V										
1.28										
1.29										
1.38										
1.45										
1.47										
1.48										
1.60										
1.61										
1.62										
1.64										
NOTE: V indicates verify-only support.										

Table 4.3-2 — RC Views Supported from 5E11 OS to 5E12 *5ESS*-2000 Switch

CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 7	CLASS 9	CLASS 10	CLASS 12	CLASS 21	CLASS 22	CLASS 23
1.5V	2.5V	3.3	4.1V	7.3	9.40	10.4	12.5	21.32	22.14V	23.1V
1.6		3.5	4.2	7.4	9.41		12.9V			23.2
1.8			4.14							23.3
1.9			4.24							23.8
1.10			4.39							23.11
1.11										23.16
1.12V										23.20
1.20										23.21
1.21										23.22V
1.22										23.23
1.23										23.30
1.24										23.33
1.25V										23.34
1.26										
1.27V										
1.28										
1.29										
1.38										
1.45										
1.47										
1.48										
1.60										
1.61										
1.62										
1.64										

NOTE: V indicates verify-only support.

Table 4.3-3 — RC Views Supported from 5E12 OS to 5E13 *5ESS*-2000 Switch

CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 7	CLASS 9	CLASS 10	CLASS 12	CLASS 21	CLASS 22	CLASS 23
1.5V	2.5V	3.3	4.1V	7.3	9.40	10.4	12.5	21.32	22.14V	23.1V
1.6		3.5	4.2	7.4	9.41		12.9V			23.2
1.8			4.14							23.3
1.9			4.24							23.8
1.10			4.39							23.11
1.11										23.16
1.12V										23.20
1.20										23.21
1.21										23.22V
1.22										23.23
1.23										23.30
1.24										23.33
1.25V										23.34
1.26										
1.27V										
1.28										
1.29										
1.38										
1.45										
1.47										
1.48										
1.60										
1.61										
1.62										
1.64										

NOTE: V indicates verify-only support.

Table 4.3-4 — RC Views Supported from 5E13 OS to 5E14 *5ESS*-2000 Switch

CLASS 1	CLASS 2	CLASS 3	CLASS 4	CLASS 7	CLASS 9	CLASS 10	CLASS 12	CLASS 21	CLASS 22	CLASS 23
1.5V	2.5V	3.3	4.1V	7.3	9.40	10.4	12.5	21.32	22.14V	23.1V
1.6		3.5	4.2	7.4	9.41		12.9V			23.2
1.8			4.14							23.3
1.9			4.24							23.8
1.10			4.39							23.11
1.11										23.16
1.12V										23.20
1.20										23.21
1.21										23.22V
1.22										23.23
1.23										23.30
1.24										23.33
1.25V										23.34
1.26										
1.27V										
1.28										
1.29										
1.38										
1.45										
1.47										
1.48										
1.60										
1.61										
1.62										
1.64										
NOTE: V indicates verify-only support.										

4.3.4 PROPER VERSION SELECTION

Table 4.3-5 displays the proper version to be entered in the format “VERSION=X” at the beginning of the APPTEXT session.

Table 4.3-5 — APPTEXT Session Entries for Version Selection

FOR RELEASE NUMBER	SET VERSION TO
5E11	10
5E12	11
5E13	12
5E*X*	5E*X - 1*

4.3.5 COMMON ERROR CONDITIONS

4.3.5.1 Parameter Placement

The RC/V text interface session parameters must be entered at the beginning of an APPTEXT session before the first "FORM=KEYWORD" name/value pair. Similarly, the "VERSION=X" session parameter must be entered at the beginning of the session. An error results if the "VERSION=X" parameter is sent at other than the beginning of the session.

4.3.5.2 Version Selection Incompatible

If a *5ESS*-2000 switch version that is not compatible is selected, an error message is sent to the OS from the switch and the session is not established. That is, an error message is sent if a version is selected such that "X" equals an alpha character or a numeric value for a nonexistent or unsupported software release of the *5ESS*-2000 switch.

4.3.6 OTHER CONDITIONS

4.3.6.1 Switch Features or Attributes

The service provider may wish to perform RC activity using *5ESS*-2000 switch software release features or attributes that are not consistent with the existing OS implementation. In this case, the service provider personnel should open a direct asynchronous channel to the *5ESS*-2000 switch. This RC channel may be established directly to the switch or via a pass-through from the OS, using the communication software for OSs. On a direct RC channel using the *5ESS*-2000 switch menu interface, the translator is not invoked and all interface outputs (such as views or error messages) are in the current software release format.

4.3.6.2 Version Selection Equals Current *5ESS*-2000 Switch Software Release

If a *5ESS*-2000 switch version that equals the current *5ESS*-2000 switch software release is entered, the translator is not invoked and the parameter is ignored. Specifically, if an OS establishes a session to a current software release and sends "VERSION=<current software release>", the translator is not invoked.

4.3.6.3 Unsupported Capabilities

The DC/AFT feature is an APPTEXT feature, consequently, the completeness of other RCOS activity is not guaranteed when using the DC/AFT feature. The ACSR and office records activity, for example, may or may not succeed, depending on the changes made in the *5ESS*-2000 switch software release.

The output of any miscellaneous RCOS activity, such as APPTEXT verify, is presented in the current software release.

4.3.6.4 Response Messages Through the Translator

There is no reverse translation of a verify request to present the verify results in the format of the earlier software release.

Warning or error messages and data returned to the OS are not manipulated or reverse translated in any way, but are passed back to the originating OS in the format resident on the current software release. For example, the failure of a *5ESS*-2000 switch RC results in an error message in the current release format.

Essentially, all data returned to the OS is in the format of the current software release.

4.3.7 RC TIME-OUT MESSAGES

The use of the translator may add time to the RC interaction between OS and switch. The RC transaction time-outs implemented by the OSs may require adjustment to account for the possible additional response time incurred by using the translator.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

5. MESSAGE STRUCTURE DEFINITIONS

This section consists of the following subsections:

- Section 5.1, “Overview”

  The basic protocol interface message structures are described as follows:

  BX.25 Protocol Interface — The message structures for the lower three layers and session layer protocols are defined in the BX.25 Issue 3 and addendum document Issue 3 (PUB.54001 and PUB.54001A).

  The capabilities messages are contained in the data portion of the BX.25 data session protocol data unit (SPDU) and are, in turn, delivered by BX.25 Layer 3 data packets to the remote processor. The capabilities message structures are defined in this section.

  TCP/IP Interface — The BX.25 session layer provides functionality to accommodate capabilities message structures that are larger than the size of the application input buffer. To support this functionality in the RCOS TCP/IP interface, each capabilities message structure exchanged between the RCOS TCP/IP client/server processes must contain a 3-byte header in the following format:

BYTE 1	BYTE 2	BYTE 3
Header-id	Length	

  In this format, “Header-id” indicates 0x01, the [ctrl-key]-A character; “Length” indicates the length of the message, not including the three bytes in this header. The RCOS TCP/IP client/server processes must also follow the rules for setting the *continuation flag* in the capabilities message header as specified in the following sections.

- Section 5.2, “Recent Change and Office Records Capabilities Message Structure”

  The remote processor accesses the switch data base via invoking the RC capabilities procedures. The message structures for these RC capabilities messages are defined in this section.

- Section 5.3, “Global Recent Change (GRC) Capability Message”

  The message data format for GRC capabilities and the GRC processing request and response messages are provided in this section.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

5.1 OVERVIEW

Each definition contained in this section includes:

- A brief description of the message direction and use.
- Tables list the information fields contained in the message. For each information field, the tables indicate:
  - — The **section** in this specification that describes the information fields
  - — Whether **inclusion** is mandatory (M), optional (O), or dependent on other circumstances (indicated and explained by footnotes)
  - — The **length**, in octets, where "?" means the length is variable
  - — Explanatory footnotes, as necessary
  - — Key words (or command words) that are used in request messages are described in "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2.
- In each individual message structure, only the Man-Machine Language (MML) format is shown. The Program Documentation Standards (PDS) formatted message structure can be mapped to the MML formatted structure by following the one-to-one relationship in the tables described in "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2. The following conventions are used in the message formats shown in this document:
  - — Uppercase letters are used for keywords (or commands) in each input/output line contained in a message. They must be entered exactly as they are shown and can be entered either in uppercase or in lowercase.
  - — Lowercase letters are used for variables in a message. They must be replaced by actual values.
  - — Three types of format notation are used to show the choices in entering the message.

    [ ] Brackets enclose optional entries that may be included in the message or may be omitted.

    { } Braces enclose two or more entries where one entry, but only one, must be included in the message.

    | Bars separate a selection of entries enclosed by braces or brackets. One of the entries separated by OR bars, but only one, may be selected; whether or not a selection must be made depends on whether brackets or braces enclose the selection.

The information fields are listed in order of appearance in the message.

The recent change (RC) capabilities requests are carried out via Text Recent Change request messages or Batch History request messages. The office records (OFRs) capabilities requests are carried out via OFR processing request messages and/or OFR output request messages. The ACSR RC capabilities requests are carried out via OFR/ACSR RC output request messages. These messages will be sent to the network with the PDS or MML formatted input lines in the message data. An input line is considered to be a character string delimited by a <cr> (carriage return) control character.

The network sends at least one response message to the remote processor with the "continuation flag" set on for each input line in the request message. In addition, the network sends one more response message with the "continuation flag" reset to indicate the finish of the request processing.

The general formats for Recent Change and Office Records request/response messages and the associated message data are described in the following paragraphs.

5.1.1 GENERAL FORMAT FOR REQUEST AND RESPONSE MESSAGES

All request messages sent from the remote processor to the network have a general format as follows:

General Format for Request Message

<–––––– Message Header ––––––>				
Opcode	Request Type	Sequence Number	Continuation Flag	Message Data

All response messages sent from the network to the remote processor have a general format as follows:

General Format for Response Message

<–––––– Message Header ––––––>							
Reply	Reply Opcode	Reserved Field	Opcode	Request Type	Sequence Number	Continuation Flag	Message Data

The opcode determines which system inside the network handles the request and processes the message data.

The request type determines the type of service desired for a particular opcode.

The continuation flag is used both by the network and the remote processor to split messages into small segments due to their large size.

The continuation flag is used when the network sends more than one response messages to reply to a single request from the remote processor. When this is necessary, the continuation flag is set to RCMCONTIN (flag on) for each response message corresponding to each input line in a request message. On the last response message, the continuation flag is set to RCMNOCONT (that is, flag off).

The remote processors can also continue input request messages by setting the continuation flag to RCMOCONT on their input request message. This requires the remote processor to completely fill the message data portion of the first request message. Subsequent continuation messages will contain no message header but will contain up to 1,024 bytes of data. The network will recognize that the input request message has been completely received when a subsequent message received from the remote processor is less than 1,024 bytes.

The continuation flag can also be used to continue input requests without the remote processor completely filling the message data portion of the first and subsequent requests. In this case, the remote processor must set the continuation flag to RCMCONTIN (flag on), for each continuation of the input request message and set the continuation flag to RCMNOCONT (that is, flag off) for the last continuation of the input request message.

The message data format for request messages is different from that for response messages. The message data format also varies from one type of request/response message to another type.

5.1.2 GENERAL FORMAT FOR MESSAGE DATA

The general message data formats for request messages are described in this section. The message data for response messages are also explained in this section and are detailed in the sections for individual response message structures.

The message data for request messages can contain one or more input lines in the PDS or MML format or can contain a null character. An input line begins with a keyword (or command) and is delimited by a <cr>. The message data for a response message is terminated with a null character ('\0').

5.1.2.1 Message Data Format for RC Capabilities

The message data for a Text Recent Change request message is comprised of one or more of seven text RC input lines as shown in Table 5.1-1 in the PDS and MML formats, depending on the Lucent Technologies *5ESS*®-2000 switch office. Any input line may be used at any time to perform its intended function, and may be used any number of times during the session.

Table 5.1-1 — Text Recent Change Input Lines

LINE #	PDS FORMAT	MML FORMAT
(1a)		VERSION=g
(1b)	FORM(bVc,d)[,e,...,e],d	FORM=bVc&d[,e,...,e],d
(1c)	FORM(viewname,d)[,e,...,e],d	FORM=viewname&d[,e,...,e],d
(2a)	MODE BMI,DEMAND,CLERK f	MODE=BMI,DEMAND,CLERK=f
(2b)	MODE BMI,TIMEREL,CLERK f	MODE=BMI,TIMEREL,CLERK=f
(3)	MODE IM	MODE=IM
(4)	MODE BMR,CLERK f	MODE=BMR,CLERK=f
(5)	VFYMODE	VFYMODE
(6)	LASTSO	LASTSO
(7)	NEXTSO	NEXTSO

The message data for a Recent Change Batch History request message is comprised of one of two RC batch history input lines shown in Table 5.1-2 in the PDS and MML formats. Any input line may be used at any time to perform its intended function, and may be used any number of times during the session. Refer to the RC Batch Release document for detail (See "REFERENCES", Section 7, for the document number).

Table 5.1-2 — Batch History Input Lines

LINE #	PDS FORMAT	MML FORMAT
(8)	REPT:RCHIST,CLERK clerkname [,FORMAT {SUMMARY|DETAIL}] [[,ALL]|[,PENDING][,COMPLETE] [,ERROR][,DEMAND]] [,TIME (mmddyyhhmm,mmddyyhhmm)]	REPT:RCHIST,CLERK=clerkname [,FORMAT={SUMMARY|DETAIL}] [[,ALL]|[,PENDING][,COMPLETE] [,ERROR][,DEMAND]] [,TIME (mmddyyhhmm,mmddyyhhmm)]
(9)	REPT:RCHIST,ACTIVITY	REPT:RCHIST,ACTIVITY
(10)	EXC:RCRMV,CLERK clerkid EXC:RCRLS,CLERK clerkid	EXC:RCRMV,CLERK=clerkid EXC:RCRLS,CLERK=clerkid

The following gives a brief explanation of the input lines mentioned in Tables 5.1-1 and 5.1-2:

Line 1 Enters Recent Change forms. A form is a format used to query or update the *5ESS*-2000 switch data base. These forms are applied immediately to the *5ESS*-2000 switch data base if the request is in the immediate RC mode. (See Line 3.)

Line 2 Sets the input mode such that subsequent form entries are batched to be applied at a later time. If the batch mode input (BMI) is batched as demand type (Line 2a), the batched RCs will be released on demand via "MODE=BMR,CLERK=f" (Line 4) command. If the BMI is batched as time release type (Line 2b), the batched RCs will be released at the preset release date and time.

This input line must be entered on a line by itself (that is, must be delimited by a <cr>). Otherwise, other input following on the same line will cause the entire line to become invalid (fail), and an error message will be sent to the remote processor. If this input line fails, all subsequent input lines except "MODE=BMR,CLERK=f" (Line 4) will be flushed with an error message until another successful "MODE=BMI,{DEMAND|TIMEREL}, CLERK=f" (Line 2), "MODE=IM" (Line 3) or the end of the request message is encountered.

When TIMEREL is specified, the input line (Line 2b) must be followed by an insertion of form BV1 (Service Order form). If the message number of the BV1 form is greater than zero, then subsequent form entries will be dependent. That is, if any form entry fails during application to the *5ESS*-2000 switch data base, then all form entries after that will be automatically withheld. Errors generated when batched form entries are applied to the *5ESS*-2000 switch data base are stored with those batched form entries. They may be reviewed via the batch history request message (that is, REPT:RCHIST input line) (see Table 5.1-2).

Line 3 Sets the input mode to immediate mode (IM) such that subsequent form entries are applied to the *5ESS*-2000 switch data base immediately. This is the default mode if mode was not previously set.

This input line must be entered on a line by itself (that is, delimited by a <cr>). Otherwise, other input following on the same line will cause the entire line to become invalid (fail), and an error message will be returned to the remote processor. If this input line fails, all subsequent input lines except "MODE=BMR, CLERK=f" (Line 4) will be flushed with an error message until another successful "MODE=BMI,{DEMAND|TIMEREL},CLERK=f" (Line 2), "MODE=IM" (Line 3) or the end of the request message is encountered.

Line 4 Releases previously batched form entries. This input line does not alter existing mode.

Line 5 Verifies the current input mode. The response to this query will be one of the following:

```
OK IM
OK BMI DEMAND CLERK=f
OK BMI TIMEREL CLERK=f
```

Line 6 Queries the key of the Service Order form of the **last** RC accepted for

batch. The key of a service order is the combination of service order number, item number, and message number.

This input line is used with batch type of TIMEREL after inserting the BV1 form (Service Order form) and one or more RC forms. The response to this query is the following:

```
OK ORDNO=ordno ITNO=itno MSGNO=msgno
```

Where: ordno = service order number
itno = item number
msgno = message number

This input line is valid only when mode is BMI of TIMEREL type (that is, batch mode input of TIMEREL type). It returns an error message when entered in any other mode (that is, MODE IM or MODE BMI of DEMAND type).

Line 7 Queries the key of the Service Order form of the **next** RC accepted for batch. The key of a service order is the combination of service order number, item number, and message number.

This input line is used with batch type of TIMEREL after inserting the BV1 form (Service Order form) and one or more RC forms. The response to this query is the following:

```
OK ORDNO=ordno ITNO=itno MSGNO=msgno
```

Where: ordno = service order number
itno = item number
msgno = message number

This input line is valid only when mode is BMI of TIMEREL type (that is, batch mode input of TIMEREL type). It returns an error message when entered in any other mode (that is, MODE IM or MODE BMI of DEMAND type).

b Decimal digits that specify the class of the RC/V view that is to be referenced. See the RC Menu Mode or RC Text Interface document for a list of existing classes (See “REFERENCES”, Section 7, for the document numbers).

c Decimal digits that specify the view within a class of RC/V views that is to be referenced. See the RC Menu Mode or RC Text Interface document for a list of views assigned to each existing class (See “REFERENCES”, Section 7, for the document numbers).

viewname As an alternative to the bVc format, the view may also be represented by the viewname, for example, RC_LINE0. Refer to the RC Menu Mode or RC Text Interface document for a list of view names (See “REFERENCES”, Section 7, for the document numbers). If viewname contains a special character (for example, “_”), double quotes are required around the viewname.

d Specifies a desired operation mode (or action) for each RC form. The six valid operations are:

- NEW - operation mode to insert new data.
- CHG - operation mode to update existing data.
- OUT - operation mode to delete existing data.

- ATTR - operation mode to list all the attributes on a specified form.
- VFY - operation mode to initiate a verify of existing data.
- MVFY - operation mode to initiate a verify of existing data. Name and value pairs are output only when the value is non-default and non-null.
- SHVFY - operation mode to initiate a verify of existing data. Name and value pairs are output only when the value is non-default, non-null, and changeable.

A mode of operation is specified twice within each form. The choice of operation mode that is used to begin the form must be the same as the one that is used at the end to process and terminate the form.

e A form can have one or more different or repeated form inputs. Corrections can be made by repeating inputs. Each of these form specification inputs is either a data value input or a form control command and the options listed as follows:

PDS format
{name value|name|SET (“name”,“value”)|LNEW (“name”,“value”)|
LCHG (“name1”,“value1”)[-“name2”]|LCHG (“name1”,“value1”)-
(“name2”,“value2”)|LOUT (“name1”,“value1”)|MSG}

MML format
{name=value|name|SET=“name”&“value”|LNEW=“name”&“value”|
LCHG=“name1”&“value1”[-“name2”]|LCHG=“name1”&“value1”-
“name2”&“value2”|LOUT=“name1”&“value1”|MSG}

As an example:
FORM=BV1&NEW,ORDNO=1,ITNO=2,MSGNO=10,RDATE=031587,\p
RTIME=1200,NEW

This input line has five specification inputs (ORDNO, ITNO, MSGNO, RDATE, and RTIME) which are in name value pairs.

Where:

- name value - Name and value pairs are used to specify attributes (either key or data items) and to assign values to attributes. One or more key data items are required to specify a particular collection of data called a view tuple within the view being worked on. Nonkey names are used to enter or change nonkey data items within a particular view tuple. Key and (nonkey) data item names and their value ranges are dependent on the particular view being accessed by the current form. Those names and associated value domains for each view are available in the RC Text Interface document (See “REFERENCES”, Section 7, for the document number).
- name - A name not paired with a value will enter a null value. This form of input can be used on nonrequired data items only.
- SET - This is a special command that is used to enter name value pairs that may have special characters embedded within the name. For example, there are some views that have data item names which contain the character “.”. Other special characters are “[”, “]”, “/”, “\”

"-", and "_". Double quotes are required around the names containing special character(s). If special characters are embedded in the value and not the name, then double quotes should be used around the value and the SET command need not be used.

- Special characters (".", "[","]", "/", "\", "-", and "_") and space - If any of these characters is embedded in attribute name/value or remark field, double quotes are required around the name/value or remark field.
- LNEW - This is a special command that is used to insert a new name-value pair (element) within a list without specifying the element's position. The value will be inserted in the first available row within the specified list.
- LCHG - This is a special command that is used to remove or change an existing element within a list without specifying the element's position. The value specified in the first name-value pair is taken to be the "match" value. The value specified in the second name-value pair is taken to be the "replacement" value. If there is not a second value specified, it is taken to be null. For an example, given the following expression in MML.

  LCHG="LISTA.ABC"&"DOG"-"LISTA.ABC"&"CAT"

  Every occurrence (that is, row) of the value "DOG" in the list element ABC will be replaced with the value "CAT." If no value is specified for the "replacement" value, the list element specified will be nulled out (that is, deletion). The list name specified in each of the name-value pairs must be the same. However, the list element names may differ, as in the following example:

  LCHG="LISTA.ABC"&"DOG"-"LISTA.XYZ"

  Every occurrence (that is, row) of the value "DOG" in the list element ABC will result in the list element XYZ being replaced with a null value.

  When the LCHG command is being used to delete elements of a list and the second list element name is the same as the first list element name, it is optional to specify the second list element as in the following examples. Both would produce the same results:

  LCHG="LISTA.ABC"&"DOG"-"LISTA.ABC"

  LCHG="LISTA.ABC"&"DOG"
- LOUT - Delete (remove) a specific row of data or list structure attribute(s) for a given value.
- MODE - Allow the user to change from/to immediate (default) text RC to/from batch text RC[1].
- VFYMODE - Return information about the current mode along with the clerkid if applicable[1].

1. These commands are primarily for using text RC to perform batch recent change.

- LASTSO - Return the value of the service order variables for the last RC[1].
- NEXTSO - Return the value of the service order variables for the next RC[1].
- MSG - Output one error line.

  Warning: This command should not be used by the remote processor because the network furnishes the remote processor with error messages (if any) upon the network's response to the remote processor's request.

- IGN - Ignore a warning message generated on the last input line. If the warning is not ignored, the last input line will be discarded.

  Warning: This command should not be used by the remote processor because the network will automatically ignore all warning messages and return a response to the remote processor indicating so.

- ABORTFORM - Request to abort the current Recent Change Form.

  Warning: This command should not be used by the remote processor because the network will abort any recent change request that fails and provide a response to the remote processor indicating so.

f — Clerk Identification under which the batched form entries are stored. Entered as one to nine ASCII characters.

g — Version number of the network OS sending text RC to the switch. Allowable values are 6, 7, and 8. The VERSION parameter is optional. See "RCOS DC/AFT FEATURE", Section 4.3.1, for a complete description.

BMI — (Batch Mode Input).
Cause subsequent form entries to be batched so that they may be applied at a later time.

DEMAND — Set batch type such that the batched form entries can be released on demand.

TIMEREL — Set batch type such that the batched form entries are released at the date and time specified by the BV1 form (Service Order form) which must be inserted following the switch to batched mode.

IM — (Immediate).

Cause subsequent form entries to be applied immediately.

BMR — (Batch Mode Release).

Release previously batched form entries. Typically used to release DEMAND batch type batched form entries, but may also release TIMEREL batch type batched form entries (that is, the preset release date and time are overridden).

Line 8 — Reports on the history of selected RCs entered by a specified clerk into the delayed release clerk file identified by file name as "clerkname". It can generate a summary or detailed report with specified parameters, such as RC status (PENDING, COMPLETE, ERROR, or ALL), demand or time released RCs.

Line 9 Reports the activity of all RCs entered into the delayed release clerk files. It generates a report, listing the activity of RCs for every clerk who has entered RCs in the delayed release clerk files. This report provides a summary listing of all clerk files in the system, and it details by status the count of all RCs.

CLERK Represent a keyword for a batch history report name or a delayed RC file name.

clerkname Represents a batch history report specific to the RC clerk who entered the delayed release RCs. It is used to identify a delayed RC file (that is, used as file name). The value of clerk name is preceded by the keyword CLERK.

SUMMARY Represents the selection of a summary for a batch history report. The default format for batch history request is SUMMARY. The default for selection parameters are COMPLETE and ERROR.

DETAIL Represents the selection of a detail batch history report.

PENDING Select only the pending RC records for reporting.

COMPLETE
Select only the complete RC records for reporting.

ERROR Select only the RC records with error for reporting.

ALL Select all RC records for reporting. These RCs could have the following status: pending, complete, error or demand.

DEMAND Select only the demand RC records (that is, batch mode input of DEMAND type RCs) for reporting.

TIME Select the records that were/are to be released between the range of times as specified. The time fields are specified in month (mm), day (dd), year (yy), hour (hh), minute (mm) format. Leading zeros must be specified. The first time field must specify a time earlier than the second time field.

ACTIVITY Represent the activity of RCs entered into the delayed RC files. The activity includes RCs with various statuses, such as complete, error, or pending.

5.1.2.2 Message Data Format For OFR Capabilities

The message data for an Office Records request message is comprised of one or more input message (IM) commands shown in Table 5.1-3 in the PDS and MML formats. Any IM command may be used at any time to perform its intended function, and may be used any number of times during the session. The message ID is used as a unique identifier of the IM command and is used through the text to represent an OFR input line. Refer to document 235-600-700, **5ESS**-*2000 Switch Input Messages* for details.

Table 5.1-3 — Office Records Input Commands

MESSAGE ID	PDS FORMAT	MML FORMAT
IN:OFR-PARM	IN:OFR;PARM[,HR a][,LNG b] [,ONDAY c][,OFFDAY d][,ENAB e]	IN:OFR:PARM[,HR=a][,LNG=b] [,ONDAY=c][,OFFDAY=d][,ENAB=e]
OP:OFR-CAT	OP:OFR;CAT a[,OPT b][,DEVICE c]	OP:OFR:CAT=a[,OPT=b][,DEVICE=c]
OP:OFR-FORM	OP:OFR;FORM a[,b c[-d]] [,OPT e][,DEVICE f]	OP:OFR:FORM=a[,b=c[&&d]] [,OPT=e][,DEVICE=f]
OP:OFR-STATUS	OP:OFR;STATUS [[COMPLETED|PENDING|PROCESSING|SCHED|PARM]] [;REQID a]	OP:OFR:STATUS[=[COMPLETED|PENDING|PROCESSING|SCHED|PARM]] [,REQID=a]
STP:OFR	STP:OFR;REQID a	STP:OFR:REQID=a

These OFR input lines are interpreted as follows:

IN:OFR-PARM Inputs/changes the operating and scheduling parameters of the OFR control process. These parameters affect the print requests of the OP:OFR input messages.

a = Hour at which printing of automatically generated changed pages or delayed requests begin.

b = Length of time (in hours) that may be used at one time to print automatically generated changed pages or delayed requests.

c = Days on which automatic printing of changed pages may occur (MON, TUE, WED, THU, or FRI).

If more than one day is specified, a list format should be used. Example:

PDS ONDAY (MON,TUE,WED)
MML ONDAY=MON-TUE-WED

d = Days on which automatic printing of changed pages may not occur.

e = Changed page recording enable flag (YES or NO).

OP:OFR-CAT Request the printing of all categories or one category of OFR forms.

a = Category to be printed:

ALL - All categories, all forms
APPROC - Applications processor form
BRCSFD - BRCS (business and residence custom services) feature definition forms
CHG - Forms for which changed pages information is maintained
OSPS - 5400 series Operator Services Position System forms
CONFIG - 5500 series configuration forms
DARC - 5300 series and other digit analysis, screening, routing, and charging forms
EQUIP - 5700 series equipment forms
LINE - 5100 series line forms
TRUNK - 5200 series trunk forms

b = Printing option:

Both - Combines both of the following options.
DELAY - Delays printing until the time set by the IN:OFR-PARM input message
REPRINT - Reprints the last requested set of changed pages (applies to forms 5100, 5105A, 5105B, and 5171 only)

c = "FILE".

OP:OFR-FORM Request printing of an OFR form.

a = Form type

b = Key ID associated with form type

c = Number associated with the key ID, or the lower limit of a range of key ID numbers

d = Number associated with the key ID. d is an optional upper bound of a range between c and d.

e = Printing option:

DELAY - Delays printing until the time set by the "IN:OFR-PARM" input message
REPRINT - Reprints the last requested set of changed pages (applies to forms 5100, 5105A, 5105B, and 5171 only)
BOTH - Executes both the *DELAY* and the *REPRINT* options.

f = "FILE".

OP:OFR-STATUS Lists all output requests or all output requests of a specific type that are queued, and/or lists all the operating and scheduling parameters of the OFR capabilities.

COMPLETED =	List all completed output requests
PENDING =	List all pending output requests
PROCESSING =	List the processing output request
REQID =	List the specified output request, a is request identification number of print request
SCHED =	List all output requests
PARM =	List the OFR operating parameters

Note: Output defaults to SCHED and PARM.

STP:OFR Stops (or cancels) any OFR print command.

a = Request identification number of a print request.

5.1.3 LIMITATIONS ON RC AND OFR CAPABILITIES MESSAGES

The maximum number of lines in an RC or OFR capabilities request, which may consist of multiple input lines, is limited by the network file size limitations. That is, the total number of characters of the lines of a capabilities request must not exceed 1,000,000 characters. The actual limits would be somewhat less than this since RC adds service order information to the clerk file.

The maximum number of characters in one single field, which is delimited by separators (for example, comma), is 256 characters.

With the exception of <bell> (Octal 007), <ack> (Octal 006), the new line <nl> character (Octal 012), the carriage return line delimiter <cr> (Octal 015), the end of message delimiter (Octal 031), and the error codes for reply opcode RCMRCRTRERR (see “ERROR CODES FOR REPLY OPCODE RCMRCRTRERR”, Section 6.25.5), all characters in a message are ASCII non-control printable characters.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

5.2 RECENT CHANGE AND OFFICE RECORDS CAPABILITIES MESSAGE STRUCTURE

The remote processor accesses the switch data base via invoking the RC capabilities procedures. The message structures for these RC capabilities messages are defined in this section.

5.2.1 TEXT IMMEDIATE RECENT CHANGE MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum document Issue 3 (PUB.54001 and PUB.54001A). The Text Immediate Recent Change capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines.

5.2.1.1 (Immediate) Recent Change/Verify Request Message

This message is sent from the remote processor to the network for changing data in the switch data base, deleting data from the switch data base, or inserting data into the switch data base. This message is also used to query the switch data base for data verification. See Table 5.2-1.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-1 — (Immediate) Recent Change/Verify Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
MODE=IM<cr> /*optional input line*/
FORM=view&action,att=val,att=val,.....,action<cr>
.
.
.
FORM=view&action,att=val,att=val,.....,action<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=IM	Section 5.1.2
FORM	Section 5.1.2
view	Section 6.5
action	Section 6.6
att=val	Section 6.7

Notes:

1. The input line MODE=IM is optional. This is the default mode (immediate input mode) if mode was not previously set.
2. Comma is used as a separator between two pairs of attribute name and value. Equal sign (=) is used as a separator between attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.1.2 Recent Change Response Message

This message is sent from the network to respond to the remote processor request for changing data in the switch data base, deleting data from the switch data base, or inserting data into the switch data base. See Table 5.2-2.

Request Type: Text Recent Change
Direction: remote processor <- Network

Table 5.2-2 — Recent Change Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	OK

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Notes:

1. The message data in a Recent Change Response Message takes the first format if no error is encountered on the form action line of the request message, and takes the second format if an error is encountered on the form action line of the request.
2. The reserved field is filled with blanks.
3. The message data (format 1, 2, or 3) is terminated with a null character ('\0').
4. msg.# - message number identifies which input line in the request message that this response message is associated with and is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.
5. error code - length of one octet.
6. error input - the part(s) of the RC input line in a request message that cause the error.
7. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.
8. The error message can be a single message or multiple error messages separated by the new line character <nl>. In some instances, the remote processor will receive the following input line:

"?E,ERROR_LINES=2,WARNING_LINES=0"

as there first error message. This message will inform the remote processor that there are two error message lines that will follow this line. If the network has both error and warning messages to return to the remote processor, the error messages will appear first followed by the warning messages. In instances where there are no error messages to be returned to the remote processor but warning messages exist, the network will send the following message:

"?W,ERROR_LINES=0,WARNING_LINES=1"

as its first message. This message will inform the remote processor that there is one warning message that follows this line.

5.2.1.3 Verify Response Message

This message is sent from the network to respond to the remote processor request for the immediate data verification. See Table 5.2-3.

Request Type: Text Recent Change
Direction: remote processor <- Network

Table 5.2-3 — Verify Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:
msg.#<nl>FORM=view&actionatt=val<ack>att=val<ack>
.....att=val<ack><bell>
Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
FORM	Section 5.1.2
view	Section 6.5
action	Section 6.6
att val	Section 6.7

Notes:

1. The reserved field is filled with blanks.
2. The message data (format 1, 2, or 3) is terminated with a null character ('\0').
3. msg.# - message number identifies which input line in the request message that this response message is associated with and is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.
4. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.
5. Attribute name and value are separated by blank in a PDS format.
6. View and action are separated by comma.
7. action = VFY, MVFY or SHVFY
8. <ack> - control character, which is represented by Octal 006 in ASCII code.
9. <bell> - control character, which is represented by Octal 007 in ASCII code. A <bell> is present only after the last name value pair of every page break.
10. - end of message character, which is represented by Octal 031 in ASCII code.
11. error code - length of one octet.
12. error input - the part(s) of the RC input line in a request message that cause the error.

5.2.1.4 Finish Request Process Response Message

This message is sent from the network to the remote processor, indicating the finish of the request process. See Table 5.2-4.

Request Type: same as request message (See note 1)
Direction: remote processor <- Network

Table 5.2-4 — Finish Request Process Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See note 2	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See note 3	M	1

Notes:

1. For example:

   Request type = text recent change for Recent Change/Verify Request Message

   Request type = craft shell for Batch History For A Clerk Request Message

   Request type = craft shell for Office Records Processing Message

2. The reserved field is filled with blanks.
3. The message data is set to a null value (that is, '\0').

5.2.2 TEXT BATCH RECENT CHANGE MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum document Issue 3 (PUB.54001 and PUB.54001A). The Text Batch Recent Change capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines.

5.2.2.1 DEMAND Type Batch Create Request Message

This message is sent from the remote processor to the network for creating an RC DEMAND type batch file. See Table 5.2-5.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-5 — DEMAND Type Batch Create Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
MODE=BMI,DEMAND,CLERK=clerkid<cr>
FORM=view&action,att=val,att=val,.....,action<cr>

.
.
.
FORM=view&action,att=val,att=val,.....,action<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=BMI	Section 5.1.2
DEMAND	Section 5.1.2
CLERK	Section 5.1.2
clerkid	Section 6.10
view	Section 6.5
action	Section 6.6
att=val	Section 6.7

Notes:

1. The message data is comprised of the BMI input line and one or more form action lines with action = NEW, CHG, or OUT.
2. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.2 TIMEREL Type Batch Create Request Message

This message is sent from the remote processor to the network for creating an RC TIMEREL type batch file. See Table 5.2-6.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-6 — TIMEREL Type Batch Create Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:

```
MODE=BMI,TIMEREL,CLERK=clerkid<cr>
FORM=BV1&NEW,ORDNO=ordno,ITNO=itno,MSGNO=msgno,
RDATE=rdate,RTIME=rtime,NEW<cr>                          |
FORM=view&action,att=val,att=val,.....,action<cr>        |
                                                         }      repeated
                                                         |
                                                         |
FORM=view&action,att=val,att=val,.....,action<cr>
```

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=BMI	Section 5.1.2
TIMEREL	Section 5.1.2
CLERK	Section 5.1.2
clerkid	Section 6.10
ORDNO=ordno	Section 6.11
ITNO=itno	Section 6.12
MSGNO=msgno	Section 6.13
RDATE=rdate	Section 6.14
RTIME=rtime	Section 6.15
view	Section 6.5
action	Section 6.6
att=val	Section 6.7

Notes:

1. The message data is comprised of the BMI input line, the FORM BV1 input line and one or more form action lines with action = NEW, CHG, or OUT.
2. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.3 Batch Add Request Message

This message is sent from the remote processor to the network for adding RCs to an existing batch file. This request message is applicable only to an RC batch file of TIMEREL type. See Table 5.2-7.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-7 — Batch Add Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:

```
MODE=BMI,TIMEREL,CLERK=clerkid<cr>
FORM=BV1&NEW,ORDNO=ordno,ITNO=itno,MSGNO=msgno,
RDATE=rdate,RTIME=rtime,NEW<cr>                          |
FORM=view&action,att=val,att=val,.....,action<cr>        |
                                                         } repeated
                                                         |
                                                         |
FORM=view&action,att=val,att=val,.....,action<cr>
```

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=BMI	Section 5.1.2
TIMEREL	Section 5.1.2
CLERK	Section 5.1.2
clerkid	Section 6.10
ORDNO=ordno	Section 6.11
ITNO=itno	Section 6.12
MSGNO=msgno	Section 6.13
RDATE=rdate	Section 6.14
RTIME=rtime	Section 6.15
view	Section 6.5
action	Section 6.6
att=val	Section 6.7

Notes:

1. The group of the input lines (marked by "}") can be repeated one or more times.

   Format 1: If the marked group of lines is not repeated, the message data is comprised of the BMI input line, the FORM BV1 input line, and one or more form action lines with action = NEW, CHG, or OUT.

   Format 2: If the marked group of lines is repeated one or more times, the message data is comprised of the single BMI input line, followed by two or more

groups of lines, each group consisting of one FORM BV1 input line and one or more form action lines with action = NEW, CHG, or OUT.

2. The attribute value for MSGNO is automatically incremented by 10 for each RC input line associated with a given service order number and item number.
3. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
4. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.4 Batch Delete Request Message

This message is sent from the remote processor to the network for deleting RCs from an existing batch file. This request message is applicable only to an RC batch file of TIMEREL type. See Table 5.2-8.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-8 — Batch Delete Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
MODE=BMI,TIMEREL,CLERK=clerkid<cr>
FORM=BV1&OUT,ORDNO=ordno,ITNO=itno,MSGNO=msgno,OUT<cr>
FORM=BV1&OUT,ORDNO=ordno,ITNO=itno,MSGNO=msgno,OUT<cr>
.
.
.
FORM=BV1&OUT,ORDNO=ordno,ITNO=itno,MSGNO=msgno,OUT<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=BMI	Section 5.1.2
TIMEREL	Section 5.1.2
CLERK	Section 5.1.2
clerkid	Section 6.10
ORDNO=ordno	Section 6.11
ITNO=itno	Section 6.12
MSGNO=msgno	Section 6.13

Notes:

1. The message data is comprised of the BMI input line and one or more FORM BV1 input lines with action = OUT.
2. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.5 Batch Replacement Request Message

This message is sent from the remote processor to the network for replacing batched RCs in an existing batch file. This request message is applicable only to an RC batch file of TIMEREL type. See Table 5.2-9.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-9 — Batch Replacement Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
MODE=BMI,TIMEREL,CLERK=clerkid<cr>
FORM=BV1&OUT,ORDNO=ordno,ITNO=itno,MSGNO=msgno,OUT<cr>
FORM=BV1&NEW,ORDNO=ordno,ITNO=itno,MSGNO=msgno, } repeated
RDATE=rdate,RTIME=rtime,NEW<cr> |
FORM=view&action,att=val,att=val,.....,action<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
MODE=BMI	Section 5.1.2
TIMEREL	Section 5.1.2
CLERK	Section 5.1.2
clerkid	Section 6.10
ORDNO=ordno	Section 6.11
ITNO=itno	Section 6.12
MSGNO=msgno	Section 6.13
RDATE=rdate	Section 6.14
RTIME=rtime	Section 6.15
view	Section 6.5
action	Section 6.6
att=val	Section 6.7

Notes:

1. The message data is comprised of the BMI input line and the recurrence of a group of the lines, consisting of one FORM BV1 line (action = OUT), one FORM BV1 line (action = NEW), and one form action line.
2. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.6 Batch Change Time and Date Request Message

This message is sent from the remote processor to the network for changing the release date and time of RCs in an existing batch file. This request message is applicable only to an RC batch file of TIMEREL type. See Table 5.2-10.

Request Type: Text Recent Change
Direction: remote processor -> Network

Table 5.2-10 — Batch Change Time and Date Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
MODE=BMI,TIMEREL,CLERK=clerkid<cr>
or
MODE=BMI,DEMAND,CLERK=clerkid<cr>
FORM=BV1&CHG,ORDNO=ordno,ITNO=itno,MSGNO=msgno,

RDATE=rdate,RTIME=rtime,CHG<cr>
FORM=BV1&CHG,ORDNO=ordno,ITNO=itno,MSGNO=msgno,
RDATE=rdate,RTIME=rtime,CHG<cr>
.
.
.
FORM=BV1&CHG,ORDNO=ordno,ITNO=itno,MSGNO=msgno,
RDATE=rdate,RTIME=rtime,CHG<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
ORDNO=ordno	Section 6.11
ITNO=itno	Section 6.12
MSGNO=msgno	Section 6.13
RDATE=rdate	Section 6.14
RTIME=rtime	Section 6.15

Notes:

1. The message data is comprised of one or more FORM BV1 lines (action = CHG)
2. Comma is used as a separator between two pairs of attribute name and value. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.2.7 Text Batch Recent Change Response Message

This message is sent from the network to respond to the remote processor request for changing data in the switch data base, deleting data from the switch data base, or inserting data into the switch data base in the batch input mode. See Table 5.2-11.

Request Type: Text (Batch) Recent Change
Direction: remote processor <- Network

Table 5.2-11 — Text Batch Recent Change Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	OK

Message Data Format 2:

msg.#	<nl>	message text (See notes)

Message Data Format 3:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 4:

error code

Notes:

1. The reserved field is filled with blanks.
2. The message data (format 1, 2, 3, or 4) is terminated with a null character ('\0').
3. Format 1 or format 2 - If no errors are found in the RC input line of the request message, the message data takes format 1 or format 2.

   Format 3 - If errors are found in the RC input line of the request message, the message data takes format 3.

   Format 4 - If the message header is invalid, the message data takes format 4.
4. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.
5. Message text - see Table 4.2.3.1 Message Data for Text Batch Recent Change Response Messages in "Text Batch Recent Change Response Messages", Section 4.2.2.7.
6. msg.# - message number identifies which input line in the request message that this response message is associated with and is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.
7. error code - length of one octet.
8. error input - the part(s) of the RC input line in a request message that cause the error.

5.2.3 BATCH CRAFT SHELL MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum document Issue 3 (PUB.54001 and PUB.54001A). The Batch CRAFT SHELL input capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines.

5.2.3.1 Batch History for a Clerk Request Message

This message is sent from the remote processor to the network for requesting an RC batch history report for a given clerk. See Table 5.2-12.

Request Type: Craft Shell
Direction: remote processor -> Network

Table 5.2-12 — Batch History for a Clerk Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
REPT:RCHIST,CLERK=clerkname[,FORMAT={SUMMARY|DETAIL}]
[[,ALL]|[,PENDING][,COMPLETE][,ERROR][,DEMAND]]
[,TIME (mmddyyhhmm,mmddyyhhmm)]<cr>

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
REPT	Section 5.1.2
CLERK=clerkname	Section 6.10

Notes:

1. The message data is comprised of one REPT:RCHIST input line.
2. An input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.3.2 Batch History for All Clerks Request Message

This message is sent from the remote processor to the network for requesting an RC batch history report for all clerks. See Table 5.2-13.

Request Type: Craft Shell
Direction: remote processor -> Network

Table 5.2-13 — Batch History for All Clerks Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
REPT:RCHIST,ACTIVITY<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
REPT	Section 5.1.2
ACTIVITY	Section 5.1.2

Notes:

1. The message data is comprised of one REPT:RCHIST input line.
2. An input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.3.3 Batch Remove/Release Request Message

The Batch Remove request message is sent from the remote processor to the network for requesting a DEMAND type RC batch file to be removed. The Batch Release request message is sent from the remote processor to the network for requesting a DEMAND type RC batch file to be released on demand. See Table 5.2-14.

Request Type: Craft Shell
Direction: remote processor -> Network

Table 5.2-14 — Batch Remove/Release Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
EXC:RCRMV,CLERK=clerkid<cr>
or
EXC:RCRLS,CLERK=clerkid<cr>

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
EXC	Section 5.1.2
CLERK=clerkname	Section 6.10

Notes:

1. The message data is comprised of one EXC:RCRMV (or EXC:RCRLS) input line.
2. An input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.3.4 Batch History Response Messages

This message is sent from the network to respond to the remote processor request for an RC batch history report for a given clerk or all clerks. See Table 5.2-15.

Request Type: Craft Shell
Direction: remote processor <- Network

Table 5.2-15 — Batch History Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	batch history report

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Notes:

1. The reserved field is filled with blanks.

2. The message data (format 1, 2, or 3) is terminated with a null character ('\0').
3. msg.# - message number identifies which input line in the request message that this response message is associated with and is an integer of variable length.
4. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.

5.2.3.5 Batch Remove/Release Response Message

This message is sent from the network to respond to the remote processor request for removing or releasing an RC batch file for a given clerk. See Table 5.2-16.

Request Type: Craft Shell
Direction: remote processor <- Network

Table 5.2-16 — Batch Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	clerkid removed or clerkid released

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Notes:

1. The reserved field is filled with blanks.
2. The message data (format 1, 2, or 3) is terminated with a null character ('\0').
3. msg.# - message number identifies which input line in the request message that this response message is associated with and is an integer of variable length.

4. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.

5.2.4 OFFICE RECORDS, ACSR RC AND ASCII FILE RETRIEVAL MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum documents Issue 3 (PUB.54001 and PUB.54001A). The Office Records, ACSR RC and ASCII File Retrieval capabilities require messages, contained in the user portion of the BX.25 data SPDU.

5.2.4.1 Office Records Processing Request Message

This message is sent from the remote processor to the network for an office records processing request. See Table 5.2-17.

Request Type: Craft Shell
Direction: remote processor -> Network

Table 5.2-17 — Office Records Processing Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
IM command<cr>
IM command<cr>
.
.
.
IM command<cr>

Notes:

1. The message data is comprised of one or more OFR input lines.
2. Each OFR input line contains one input message (IM) command and is delimited by <cr>. The formats of the valid IM commands are described in "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2, and detailed in document 235-600-700, **5ESS®**-*2000 Switch Input Messages*.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.4.2 Office Records Processing Response Message

This message is sent from the network as a response to the remote processor for requesting office records processing. See Table 5.2-18.

Request Type: Craft Shell
Direction: remote processor <- Network

Table 5.2-18 — Office Records Processing Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	system response

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Notes:

1. Message format 1 - No error is encountered on an input line of the request message.

   Message format 2 - An error is encountered on an input line of the request message or during processing.

   Message format 3 - The message header is invalid.

2. The reserved field is filled with blanks.

3. The message data (format 1, 2, or 3) is terminated with a null character ('\0').

4. msg.# - message number identifies which input line that is associated with this response message. It is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.

5. system response - the format of system responses to individual input messages are detailed in documents 235-600-700, **5ESS**-*2000 Switch Input Messages*, and 235-600-750, **5ESS**-*2000 Switch Output Messages*.

6. error code - length of one octet.

7. error input - the part(s) of the input line in a request message that cause the error, or the entire input line.

8. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.

5.2.4.3 Office Records/ACSR RC Output Request Message

This message is sent from the remote processor to the network for an office records or ACSR RC data output request. See Table 5.2-19.

Request Type: Office Records Output or ACSR RC Output
Direction: remote processor -> Network

Table 5.2-19 — OFR/ACSR RC Output Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
REQID<sep>reqid,OUTBYTES<sep>outbytes,ERRBYTES<sep>errbytes<cr>
REQID<sep>reqid,OUTBYTES<sep>outbytes,ERRBYTES<sep>errbytes<cr>
.
.
.
REQID<sep>reqid,OUTBYTES<sep>outbytes,ERRBYTES<sep>errbytes<cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
reqid (request ID number)	Section 6.16
outbytes (output bytes)	Section 6.17
errbytes (error bytes)	Section 6.18

Note:

1. sep - A separator is used between attribute name and attribute value (for example, between REQID and reqid). The separator is " " (space) for the PDS format and "=" for the MML format.
2. Each input line in the message data is delimited by <cr>.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.4.3.1 ASCII File Retrieval Request Message

This message is sent from the remote processor to the network for retrieving an ASCII file output request. See Table 5.2-20.

Request Type: ASCII File Retrieval Output
Direction: remote processor -> Network

Table 5.2-20 — ASCII File Retrieval Output Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:
DIR=<path>,FILE=<filename><cr>
Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Explanation
DIR=<path>	The full path to the directory containing the file to be retrieved.
FILE=<filename>	the name of the file to be sent back to the remote processor

Note:

1. Each input line in the message data is delimited by <cr>.
2. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.2.4.4 Office Records, ACSR RC, and ASCII File Retrieval Output Response Message

This message is sent from the network as a response to the remote processor request for retrieving the output of OFR processing, ASCII File Retrieval, or the ACSR RC information on daily ACSR activities (See note 8). See Table 5.2-21.

Request Type: Office Records Output, ACSR RC Output, or ASCII File Retrieval Output
Direction: remote processor <- Network

Table 5.2-21 — OFR/ACSR RC Output Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1(a):

msg.#	<nl>	output file

Message Data Format 1(b):

msg.#	<nl>	error file

Message Data Format 1(c):

msg.#	<nl>	input line	<nl>	x'mission message

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Message Data Format 4:

complete message

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
Error Code	Section 6.25
Error Message	Section 6.26.1

Notes:

1. Message format 1(a), 1(b), and 1(c) - No error is encountered on an input line of the request message.

   Message format 2 - An error is encountered on an input line of the request message or during processing.

   Message format 3 - The message header is invalid.

   Message format 4 - Finish Request Process Response

2. The reserved field is filled with blanks.
3. The message data [format 1(a), 1(b), 1(c), 2, 3, or 4] is terminated with a null character ('\0').
4. msg.# - message number identifies which input line that is associated with this response message. It is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.
5. error code - length of one octet.

6. error input - the part(s) of the input line in a request message that cause the error, or the entire input line.
7. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.
8. For ACSR RC data retrieval, the format of the output file is as follows:

   DN <ack> LCEN <ack> DN <ack> LCEN <ack> ... DN <ack> LCEN <ack>

   Refer to the RC Text Interface document for the format and length of the primary directory number (DN) and line card equipment number (LCEN) fields. See "REFERENCES", Section 7, for the document number.
9. <ack> - control character, which is represented by Octal 006 in ASCII code.

5.2.4.5 Text Recent Change Interface Termination Request Message

This message is sent from the remote processor to the network for terminating text RC activities. This message is applicable to immediate and batch text RC activities. See Table 5.2-22.

Request Type: Text Recent Change Termination
Direction: remote processor -> Network

Table 5.2-22 — Text Recent Change Interface Termination Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	1

Note:

1. The message data is a null field (that is, '\0').

5.2.4.6 Text Recent Change Interface Termination Response Message

This message is sent from the network to respond to the remote processor for requesting the termination of text RC activities. This message is applicable to immediate and batch text RC activities. See Table 5.2-23.

Request Type: Text Recent Change Termination
Direction: remote processor <- Network

Table 5.2-23 — Text Recent Change Interface Termination Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	?

Notes:

1. The reserved field is filled with blanks.
2. The message data is terminated with a null character ('\0').
3. If the request is processed successfully, the message data is a null field (that is, '\0').
4. If the request is processed unsuccessfully, the message data is one octet of error code. (See "MISCELLANEOUS ERROR CODES", Section 6.25.1.)

5.2.4.7 Craft Shell Interface Termination Request Message

This message is sent from the remote processor to the network for terminating craft shell activities. See Table 5.2-24.

Request Type: Craft Shell Termination
Direction: remote processor -> Network

Table 5.2-24 — Craft Shell Interface Termination Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	1

Note:

1. The message data is a null field (that is, '\0').

5.2.4.8 Craft Shell Interface Termination Response Message

This message is sent from the network to respond to the remote processor for requesting the termination of craft shell activities. See Table 5.2-25.

Request Type: Craft Shell Termination
Direction: remote processor <- Network

Table 5.2-25 — Craft Shell Interface Termination Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	?

Notes:

1. The reserved field is filled with blanks.
2. The message data is terminated with a null character ('\0').
3. If the request is processed successfully, the message data is a null field (that is, '\0').
4. If the request is processed unsuccessfully, the message data is one octet of error code. (See "MISCELLANEOUS ERROR CODES", Section 6.25.1.)

5.2.5 RECENT CHANGE VIEW TIME-OUT MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum documents Issue 3 (PUB.54001 and PUB.54001A). The Recent Change View Time-Out capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines.

The messages defined in this section are applicable during the entire session.

5.2.5.1 Recent Change View Time-Out Request Message

This message is sent from the remote processor to the network for requesting the time-out values for RC view operations. See Table 5.2-26.

Request Type: RC Time-Out
Direction: remote processor -> Network

Table 5.2-26 — Recent Change View Time-Out Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	1

Note:

1. The message data is a null field ('\0').

5.2.5.2 Recent Change View Time-Out Response Message

This message is sent from the network to respond to the remote processor request for requesting the time-out values of RC view operations. See Table 5.2-27.

Request Type: RC Time-out
Direction: remote processor<- Network

Table 5.2-27 — Recent Change View Time-Out Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:
Read_Time<ack>DF_Time<ack>view_name<del>timeout<ack>
view_name<del>timeout<ack>.....view_name<del>timeout<ack>

Message Data Format 2:

error code

Keywords/Parameters Cross-Reference Table:

Keyword/Parameter in Message Data	Reference
Read_Time	Section 6.19
DF_Time (default time)	Section 6.20
view_name	Section 6.5
timeout	Section 6.21

Notes:

1. The reserved field is filled with blanks.
2. The message data (format 1 or 2) is terminated with a null character ('\0').
3. msg.# - message number identifies which input line is associated with this response message. It is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.
4. <nl> - new line character, which is represented by Octal 012 in ASCII code.
5. <ack> - control character, which is represented by Octal 006 in ASCII code.

6. <del> - delimiter, which is a blank in the PDS format and "=" in the MML format.
7. <bell> - control character, which is represented by Octal 007 in ASCII code.
8. error code - length of one octet.

5.2.6 STATUS CHECK MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs (Session Protocol Data Units) are defined in the BX.25 Issue 3 and addendum documents Issue 3 (PUB.54001 and PUB.54001A). The Status Check capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines.

The messages defined in this section are applicable during the entire session.

5.2.6.1 Status Check Request Message

This message is sent from the remote processor to the network for checking the progress or status of the request, or for checking the status of the network. See Table 5.2-28.

Request Type: Status Check
Direction: remote processor -> Network

Table 5.2-28 — Status Check Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	1

Note:

1. The message data is set to the null character (that is, '\0').

5.2.6.2 Status Check Response Message

This message is sent from the network to respond to the remote processor request for checking the progress or status of the request or for checking the status of the network. See Table 5.2-29.

Request Type: Status Check
Direction: remote processor <- Network

Table 5.2-29 — Status Check Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Subsystem	Section 6.22	M/NA	1
Activity	Section 6.23	M/NA	1
State	Section 6.24	M/NA	1

Notes:

1. The reserved field is filled with blanks.
2. The message data is terminated with a null character (\0').
3. NA - not applicable. That is, the field is not applicable when the network is idle.
4. The message data is set to the null character (that is, '\0') if the network is idle.
5. The message data format is as follows, if the network is busy:

<———Message Data———->		
Subsystem	Activity	State

5.2.7 ABORT MESSAGES

The BX.25 packets required for transport services and BX.25 data SPDUs are defined in the BX.25 Issue 3 and addendum documents Issue 3 (PUB.54001 and PUB.54001A). The Abort capabilities require messages, contained in the user portion of the BX.25 data SPDU, which this section defines. The messages defined in this section are applicable during the entire session.

5.2.7.1 Abort Request Message

This message is sent from the remote processor to the network for aborting a request or an activity. See Table 5.2-30.

Request Type: Abort
Direction: remote processor -> Network

Table 5.2-30 — Abort Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	See notes	M	1

Note:

1. The message data is a null field (that is, '\0').

5.2.7.2 Abort Response Message

This message is sent from the network to respond to the remote processor request for aborting a request or an activity. See Table 5.2-31.

Request Type: Abort
Direction: remote processor <- Network

Table 5.2-31 — Abort Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Subsystem	Section 6.22	M/NA	1
Activity	Section 6.23	M/NA	1
State	Section 6.24	M/NA	1
Error Code	Section 6.25	M/NA	1

Notes:

1. The reserved field is filled with blanks.
2. The message data is terminated with a null character ('\0').
3. If the abort request is successful, the message data format is as follows:

<———Message Data ———>		
Subsystem	Activity	State

4. If the abort request is not successful, the message data format is as follows:

<–––––––––––––– Message Data ––––––––––––––>			
Subsystem	Activity	State	Error Code

5. If the network is idle, the message data format is as follows:

<––Message Data––>
error code

6. The error code field is applicable if the abort request is unsuccessful.
7. NA - not applicable

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

LIST OF TABLES

5.3 GLOBAL RECENT CHANGE (GRC) CAPABILITY MESSAGE

5.3.1 Message Data Format For GRC Capabilities

The message data for a GRC request message is comprised of one or more IM commands shown in Table 5.3-1 in the MML format. Any IM command may be used at any time to perform its intended function, and may be used any number of times during the session. The message ID is used as a unique identifier of the IM command and is used through the text to represent a GRC input line. Refer to document 235-600-700, **5ESS®**-*2000 Switch Input Messages*for details.

The GRC is not supported by the RCOS DC4I/AFT capability. See “RCOS DC/AFT FEATURE”, Section 4.3.1, for a complete description of DC/AFT capability.

Table 5.3-1 — Global Recent Change Input Commands

MESSAGE ID	MML FORMAT
GRC:BACKOUT	GRC:BACKOUT[,NAME=a[,SECT=b]];
GRC:CANCEL	GRC:CANCEL[,NAME=a[,SECT=b]];
GRC:CONTINUE	GRC:CONTINUE,MODE=a[,NAME=b[,SECT=c]];
GRC:HALT	GRC:HALT;
GRC:REPORT	GRC:REPORT[,NAME=a[,SECT=b]];
GRC:RETRY	GRC:RETRY[,NAME=a[,SECT=b]];
GRC:RMV	GRC:RMV,NAME=a;
GRC:SCHED	GRC:SCHED,NAME=a,RDATE=b,RTIME=c[,SECT=d][,TEST=e];

These GRC input lines are interpreted as follows:

GRC:BACKOUT Request that the changes made by a GRC update be backed out. This initiates the restoral of the line data to its original, pre-updated condition.

a = GRC name (up to 10 characters)

b = GRC section number.

GRC:CANCEL Request that a GRC that is waiting in queue be canceled. This allows the user to unschedule a GRC.

a = GRC name (up to 10 characters)

b = GRC section number.

GRC:CONTINUE Request that a GRC be restarted from the point of suspension. This restarts GRC updates, backouts, or retries.

a = GRC mode
BACKOUT - restore line attributes to values before GRC
RETRY - restart to correct failed recent changes
UPDATE - make changes to line attributes

b = GRC name (up to 10 characters)

c = GRC section number.

GRC:HALT Suspend the execution of the current GRC job.

GRC:REPORT Request the current schedule of all timed release batch and global recent change activity.

a = GRC name (up to 10 characters)

b = GRC section number.

GRC:RETRY Request that the failed GRC updates be tried again. This initiates a job that tries to apply GRC updates that previously failed.

a = GRC name (up to 10 characters)

b = GRC section number.

GRC:RMV Request that the specified GRC job be removed from the switch. This deletes every file and directory associated with the job.

a = GRC name (up to 10 characters)

GRC:SCHED Request that a GRC job be rescheduled. This allows the user to change the run date and time of a job.

a = GRC name (up to 10 characters)

b = Release date, in the form month-day-year

Month of the year (01-12)
Day of the month (01-31)
Last two digits of year (00-99).

c = Release time, in the form hour-minute

Hour (00-23)
Minute (00-59)

d = GRC section number

e = Number of RCs that will be performed as a test to verify the GRC validity (1-65535).

5.3.2 GRC Processing Request Message

This message is sent from the remote processor to the network for a GRC processing request. See Table 5.3-2.

Request Type: Craft Shell
Direction: remote processor -> Network

Table 5.3-2 — GRC Processing Request Message Content

Information Field	Reference	Type	Length
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format:

IM command<cr>
IM command<cr>
.

.
.
IM command<cr>

Notes:

1. The message data is comprised of one or more GRC input lines.
2. Each GRC input line contains one IM command and is delimited by <cr>. The formats of the valid IM commands are described in "GENERAL FORMAT FOR MESSAGE DATA", Section 5.1.2, and detailed in document 235-600-700, **5ESS**-*2000 Switch Input Messages*.
3. <cr> - control character for carriage return, which is represented by Octal 015 in ASCII code.

5.3.3 GRC Processing Response Message

This message is sent from the network as a response to the remote processor for requesting GRC processing. See Table 5.3-3.

Request Type: Craft Shell
Direction: remote processor <- Network

Table 5.3-3 — GRC Processing Response Message Content

Information Field	Reference	Type	Length
Reply	Section 6.8	M	1
Reply Opcode	Section 6.9	M	1
Reserved	See notes	M	2
Opcode	Section 6.1	M	1
Request Type	Section 6.2	M	1
Sequence Number	Section 6.3	M	1
Continuation Flag	Section 6.4	M	1
Message Data	as follows	M	?

Message Data Format 1:

msg.#	<nl>	system response

Message Data Format 2:

error code	msg.#	<nl>	error input	<nl>	error message

Message Data Format 3:

error code

Notes:

1. Message format 1 - No error is encountered on an input line of the request message.

   Message format 2 - An error is encountered on an input line of the request message or during processing.

   Message format 3 - The message header is invalid.

2. The reserved field is filled with blanks.

3. The message data (format 1, 2, or 3) is terminated with a null character ('\0').

4. msg.# - message number identifies which input line that is associated with this response message. It is an integer of variable length. For example, Message Number 2 indicates that this response message corresponds to the second input line of the request message.

5. system response - the format of system responses to individual input messages are detailed in documents 235-600-700, **5ESS**-*2000 Switch Input Messages*, and 235-600-750, **5ESS**-*2000 Switch Output Messages*.

6. error code - length of one octet.

7. error input - the part(s) of the input line in a request message which cause the error, or the entire input line.

8. <nl> - new line character, which is used as a separator and represented by Octal 012 in ASCII code.

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

6. MESSAGE FIELD DEFINITIONS

This section defines the use and coding of individual information fields in the messages defined in “MESSAGE STRUCTURE DEFINITIONS”, Section 5.

The message field definitions for session layer, packet layer, and link layer can be found in BX.25 Issue 3 and addendum documents Issue 3 (PUB.54001 and PUB.54001A).

The four information fields contained in the standard header for all request messages are as follows:

- Opcode
- Request Type
- Sequence Number
- Continuation Flag.

The six information fields contained in the standard header for all response messages are as follows:

- Reply
- Reply Opcode
- Opcode
- Request Type
- Sequence Number
- Continuation Flag.

6.1 OPCODE

The opcode information field identifies the major system that will handle a request message. The valid codes for this field are as follows:

Bit								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Opcode[a]**
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.
				.				.
0	0	1	1	0	0	0	0	Reserved
0	0	1	1	0	0	0	1	RCMRC (’1’)
0	0	1	1	0	0	1	0	RCM5E (’2’)
0	0	1	1	0	0	1	1	RCMOFR (’3’)
0	0	1	1	0	1	0	0	RCMCSH (’4’)
0	0	1	1	0	1	0	1	RCMACSR (’5’)
0	0	1	1	0	1	1	0	Reserved
0	0	1	1	0	1	1	1	Reserved

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Opcode[a]
				.				.
				.				.
				.				.
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.2 REQUEST TYPE

The request type information field identifies the type of functionality that a message requests. The valid codes for this field are as follows:

Bit								
7	6	5	4	3	2	1	0	Request [a]Type
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.
				.				.
0	0	1	1	0	0	0	0	RCMNOTAPP ('0')
0	0	1	1	0	0	0	1	RCMRCTXT ('1')
0	0	1	1	0	0	1	0	RCMTERMTXT ('2')
0	0	1	1	0	0	1	1	Reserved
0	0	1	1	0	1	0	0	Reserved
0	0	1	1	0	1	0	1	RCMCFTMSG ('5')
0	0	1	1	0	1	1	0	RCMTERMCS ('6')
0	0	1	1	0	1	1	1	Reserved
0	0	1	1	1	0	0	0	Reserved
0	0	1	1	1	0	0	1	Reserved
				.				.
				.				.
				.				.
0	1	0	0	0	0	0	0	Reserved
0	1	0	0	0	0	0	1	RCM5ESTAT ('A')
0	1	0	0	0	0	1	0	RCM5EABORT ('B')
0	1	0	0	0	0	1	1	RCM5ETIMERS ('C')
0	1	0	0	0	1	0	0	Reserved
0	1	0	0	0	1	0	1	Reserved
0	1	0	0	0	1	1	0	Reserved
0	1	0	0	0	1	1	1	Reserved

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Request [a]Type
0	1	0	0	1	0	0	0	Reserved
0	1	0	0	1	0	0	1	Reserved
0	1	0	0	1	0	1	0	Reserved
0	1	0	0	1	0	1	1	RCMOFROUT ('K')
			.					.
			.					.
			.					.
0	1	0	1	0	0	0	1	Reserved
0	1	0	1	0	0	1	0	Reserved
0	1	0	1	0	0	1	1	RCMACSROUT ('S')
0	1	0	1	0	1	0	0	Reserved
			.					.
			.					.
			.					.
0	1	0	1	0	1	1	1	RCMFLOUT ('W')
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.3 SEQUENCE NUMBER

The sequence number information field is used by the remote processor to identify the sequence of the messages sent to the network. The valid codes for this field are any ASCII characters. The sequence number in the message responding to the remote processor request will be same as that in the request message.

6.4 CONTINUATION FLAG

The continuation flag information field is used to indicate the continuation of information.

The capabilities messages can be logically related in a sequence by using the continuation flag. If a message is not the last one in the sequence, the continuation flag is set to RCMCONTIN (flag on). If a message is the last one or the only one in the sequence, the continuation flag is set to RCMNOCONT (flag off).

The valid codes for this field are as follows:

Bit								
7	6	5	4	3	2	1	0	Continuation [a]Flag
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Continuation [a]Flag
				.				.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	RCMNOCONT ('0')
0	0	1	1	0	0	0	1	RCMCONTIN ('1')
0	0	1	1	0	0	1	0	Reserved
0	0	1	1	0	0	1	1	Reserved
0	0	1	1	0	1	0	0	Reserved
				.				.
				.				.
				.				.
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.5 VIEW

View defines a collection of data items that are parts of the switch data base and can be updated or verified by specifying a view. It is represented in the PDS format of FORM (bVc) or in the MML format of FORM=bVc.

Where:

b = Decimal digits that specify the class of the RC/V view that is to be referenced. See the RC menu mode or RC text interface document for a list of existing classes. (See "REFERENCES", Section 7, for the document numbers.)

c = Decimal digits that specify the view within a class of RC/V views that is to be referenced. See the RC menu mode or RC text interface document for a list of views assigned to each existing class. (See "REFERENCES", Section 7, for the document numbers.)

As an alternative to the bVc format, the view may also be represented by the view name. Refer to the RC menu mode or RC text interface document for a list of view names. (See "REFERENCES", Section 7, for the document numbers.)

6.6 ACTION

Action specifies a desired operation mode for each recent change form. The six valid operations are:

- NEW - operation mode to insert new data.
- CHG - operation mode to update existing data.
- OUT - operation mode to delete existing data.
- ATTR - operation mode to list all the attributes on a specified form.
- VFY - operation mode to initiate a verify of all existing data.
- MVFY - operation mode to initiate a verify of existing data. Name and value pairs are output only when the value is non-default, and non-null.

- SHVFY - operation mode to initiate a verify of existing data. Name and value pairs are output only when the value is non-default, non-null, and changeable.

6.7 ATTRIBUTE NAME AND VALUE PAIRS

An “attribute name” identifies a data field in a Recent Change and Verify View.

Note: Each Recent Change and Verify View consists of one or more data fields.

The value corresponding to each “attribute name” is called an “attribute value.” Refer to the RC text interface document. (See “REFERENCES”, Section 7, for the document number.)

6.8 REPLY

The reply information field indicates whether a request is successful or not. It does not indicate the successful completion of the request. That is, an accepted request may result in an error message during processing.

In response to a request message, the reply field indicates whether the opcode field in the request message is valid or invalid.

The valid codes for this field are as follows:

Bit								
7	6	5	4	3	2	1	0	Reply[a]
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.
				.				.
0	0	1	1	0	0	0	0	Reserved
0	0	1	1	0	0	0	1	RCMREQSUCC ('1')
0	0	1	1	0	0	1	0	RCMREQFAIL ('2')
0	0	1	1	0	0	1	1	Reserved
0	0	1	1	0	1	0	0	Reserved
0	0	1	1	0	1	0	1	Reserved
				.				.
				.				.
				.				.
1	1	1	1	1	1	1	1	Reserved

Note(s):
a. The ASCII characters that the values represent are shown in parentheses.

6.9 REPLY OPCODE

The reply opcode information field indicates the result of a request after the completion of processing. The valid codes for this field are as follows:

Bit								
7	6	5	4	3	2	1	0	Reply[a] Opcode
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	RCMNOROC ('0') no reply opcode recent change request success
0	0	1	1	0	0	1	0	RCMRCERROR ('2') recent change request error
0	0	1	1	0	0	1	1	RCMRCRTRERR ('3') RTR error
0	0	1	1	0	1	0	0	RCMRCINTERR ('4') internal interface controller error
0	0	1	1	0	1	0	1	RCMRCBUSY ('5') recent change is busy with a request
0	0	1	1	0	1	1	0	RCMODERROR ('6') incorrect RC mode
0	0	1	1	0	1	1	1	Reserved
0	0	1	1	1	0	0	0	Reserved
				.				.
				.				.
0	1	0	1	0	0	0	1	Reserved

Note(s):
a. The ASCII characters that the values represent are shown in parentheses.

Bit								
7	6	5	4	3	2	1	0	Reply[a] Opcode
0	1	0	1	0	0	1	0	RCM5EIDLE ('R') interface controller is idle
0	1	0	1	0	0	1	1	RCM5EBUSY ('S') interface controller is working on previous request
0	1	0	1	0	1	0	0	RCM5EAOK ('T') abort request successful
0	1	0	1	0	1	0	1	RCM5EAERROR ('U') abort request fails
0	1	0	1	0	1	1	0	RCM5EFAIL ('V') fails to process an interface controller request
0	1	0	1	0	1	1	1	RCMCNTLRTO ('W') interface controller inactivity time-out

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Reply[a] Opcode
0	1	0	1	1	0	0	0	RCM5EOK ('X') process an interface controller request successfully
0	1	0	1	1	0	0	1	Reserved
0	1	0	1	1	0	1	0	Reserved
			.					.
			.					.
0	1	1	0	0	0	0	0	Reserved
0	1	1	0	0	0	0	1	RCMSRVFAIL ('a') fails OFR/ACSR/ASCII file output request
0	1	1	0	0	0	1	0	RCMSRVSUCC ('b') successful OFR/ACSR/ASCII file output request
0	1	1	0	0	0	1	1	RCMSRVSYNC ('c') transmission is complete for OFR/ACSR/ASCII file output/error file
0	1	1	0	0	1	0	0	Reserved
0	1	1	0	0	1	0	1	Reserved
			.					.
			.					.
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.10 CLERK IDENTIFICATION OR CLERK NAME

Clerk Identification (clerkid) indicates the file name under which the batched recent changes are stored. Clerk name (clerkname) indicates the file name under which the batched recent changes are stored and the batch history report extracts the information from this file. CLERK is the key word, which is followed by the parameter value clerkid or clerkname. The valid codes are one to nine ASCII characters.

6.11 SERVICE ORDER NUMBER

ORDNO is the attribute name for service order number; ordno represents the service order number and should be replaced by an actual value.

Service order number is an ASCII alphanumeric string and must be less than 10 characters long. There must be no blanks in the service order number.

Refer to the RC text interface document (See “REFERENCES”, Section 7, for the document number).

6.12 ITEM NUMBER

The item number information field provides for sequencing of individual requests within a service order. It represents a batch input recent change item number.

"ITNO" is the attribute name for item number; "itno" represents the value of item number. Item number is ASCII numeric. Refer to the RC Text Interface document (See "REFERENCES", Section 7, for the document number) for the valid range of item number in a service order form (BV1).

6.13 MESSAGE NUMBER

The message number is used when there are several dependent recent changes within an item; otherwise a zero should be input. It represents a batch input recent change number.

"MSGNO" is the attribute name for message number; "msgno" represents the value of message number. Message number is ASCII numeric. Refer to the RC Text Interface document (See "REFERENCES", Section 7, for the document number) for the valid range of message number in a service order form (BV1).

6.14 RELEASE DATE

The release date field specifies the date on which the batched recent changes are to be released and applied to the switch data base. The date is in the format of mmddyy, where mm (01-12) is month, dd (01-31) is day, and yy is the last two digits of calendar year. Leading zeros must be specified.

"RDATE" is the attribute name for release date; "rdate" represents the value of release date and must be replaced by an actual value. Examples of release date are 040188, and 102987.

Bit								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Release Date**
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	'0'
0	0	1	1	0	0	0	1	'1'
0	0	1	1	0	0	1	0	'2'
			.					.
			.					.
			.					.
0	0	1	1	1	0	0	1	'9'
0	0	1	1	1	0	1	0	Reserved
0	0	1	1	1	0	1	1	Reserved
0	0	1	1	1	1	0	0	Reserved
			.					.
			.					.
			.					.

Bit								
7	6	5	4	3	2	1	0	Release Date
1	1	1	1	1	1	1	1	Reserved

6.15 RELEASE TIME

The release time field specifies the time at which the batched recent changes are to be released and applied to the switch data base. The time is expressed as military time of hours and minutes (0000 to 2359). Leading zeros must be specified.

“RTIME” is the attribute name for release time; “rtime” represents the value of release time and must be replaced by an actual value. Examples of release time are 0400, 0905, 1400, and 2315.

Bit								
7	6	5	4	3	2	1	0	Release Time
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
				.				.
				.				.
				.				.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	’0’
0	0	1	1	0	0	0	1	’1’
0	0	1	1	0	0	1	0	’2’
				.				.
				.				.
				.				.
0	0	1	1	1	0	0	1	’9’
0	0	1	1	1	0	1	0	Reserved
0	0	1	1	1	0	1	1	Reserved
0	0	1	1	1	1	0	0	Reserved
				.				.
				.				.
				.				.
1	1	1	1	1	1	1	1	Reserved

6.16 REQUEST IDENTIFICATION NUMBER

The request identification number indicates the request number assigned to an office records processing request by the network and used by a remote processor to request office records output or cancel an office records processing.

The network assigns request numbers in ascending order. The upper bound is 999. Request number resets to one after 999.

“REQID” is the attribute name for the request identification number; “reqid” represents the value of the request identification number.

For ACSR RC data retrieval, the request identification number indicates the age of the output file and the error file that store information about the current and previous ACSR RC activities, successes, and errors, respectively.

reqid=1 identifies the output and error files that store the current ACSR RC data.

reqid=2 identifies the output and error files that store yesterday ACSR RC data.

reqid=3 identifies the output and error files that store the ACSR RC data the day before yesterday.

6.17 OUTPUT BYTES

The “output bytes” parameter is ASCII numeric (in unit of decimal) that specifies the starting point from which the OFR or ACSR RC output file should be transmitted to the remote processor after the network receives an OFR/ACSR RC output request message.

This parameter is normally set to zero, which designates to send the entire output file back to the remote processor. This parameter is set to a positive non-zero number in cases where the remote processor desires retransmission of a file from a given octet offset from the beginning of the file.

This parameter is set to '-' (dash), which designates to skip the transmission of the entire output file. If both “output bytes” and “error bytes” parameters are set to '-' (dash), an error message will be sent back to the remote processor.

“OUTBYTES” is the attribute name for the “output bytes” parameter; “outbytes” represents the value of the “output bytes” parameter.

6.18 ERROR BYTES

The “error bytes” parameter is ASCII numeric (in unit of decimal) that specifies the starting point from which the OFR or ACSR RC error file should be transmitted to the remote processor after the network receives an OFR/ACSR RC output request message.

This parameter is normally set to zero, which designates to send the entire error file back to the remote processor. This parameter is set to a positive non-zero number in cases where the remote processor desires retransmission of a file from a given octet offset from the beginning of the file.

This parameter is set to '-' (dash), which designates to skip the transmission of the entire error file. If both “output bytes” and “error bytes” parameters are set to '-' (dash), an error message will be sent back to the remote processor.

“ERRBYTES” is the attribute name for the “error bytes” parameter; “errbytes” represents the value of the “error bytes” parameter.

6.19 RC VIEW READ TIME

The read time for a recent change view indicates the required time (in units of seconds) to query the switch data base for data verification. Each recent change views VFY (verify), MVFY (intermediate verify) and SHVFY (short verify) operation will take the same amount of time.

6.20 RC VIEW DEFAULT TIME

The default time for a recent change view indicates the time (in units of seconds) to update the switch data base, insert data into or delete data from the switch data base. Each recent change view for CHG, NEW, or OUT operation takes the default time unless it is specified in the view name and time-out pairs in the Recent Change Time-Out Response Message. (See "Recent Change View Time-Out Response Message", Section 5.2.5.2.)

6.21 RC VIEW TIME-OUT

The time-out for a recent change view indicates the time (in units of seconds) to update the switch data base, insert data into or delete data from the switch data base for a given recent change view.

6.22 SUBSYSTEM

The subsystem information field appears in the Status Check Response Message and the Abort Response Message. It identifies which system is currently working on the previous request. This field has the same value as the opcode field from the previous request message.

The valid codes for this field are as follows:

Bit								
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Subsystem[a]**
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	0	1	1	0	0	0	0	Reserved
0	0	1	1	0	0	0	1	RCMRC ('1')
0	0	1	1	0	0	1	0	RCM5E ('2')
0	0	1	1	0	0	1	1	RCMOFR ('3')
0	0	1	1	0	1	0	0	RCMCSH ('4')
0	0	1	1	0	1	0	1	RCMACSR ('5')
0	0	1	1	0	1	1	0	Reserved
0	0	1	1	0	1	1	1	Reserved
			.					.
			.					.
			.					.
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.23 ACTIVITY

The activity information field appears in the Status Check Response Message and the Abort Response Message. It identifies the type of subsystem activity that was requested by the previous request message. This field has the same value as the request type field from the previous request message. The valid codes for this field are shown in the following example.

Bit								
7	6	5	4	3	2	1	0	Activity[a]
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	0	1	1	0	0	0	0	Reserved
0	0	1	1	0	0	0	1	RCMRCTXT ('1')
0	0	1	1	0	0	1	0	RCMTERMTXT ('2')
0	0	1	1	0	0	1	1	Reserved
0	0	1	1	0	1	0	0	Reserved
0	0	1	1	0	1	0	1	RCMCFTMSG ('5')
0	0	1	1	0	1	1	0	RCMTERMCS ('6')
0	0	1	1	0	1	1	1	Reserved
0	0	1	1	1	0	0	0	Reserved
			.					.
			.					.
			.					.
0	1	0	0	0	0	0	0	Reserved
0	1	0	0	0	0	0	1	RCM5ESTAT ('A')
0	1	0	0	0	0	1	0	RCM5EABORT ('B')
0	1	0	0	0	0	1	1	RCM5ETIMERS ('C')
0	1	0	0	0	1	0	0	Reserved
0	1	0	0	0	1	0	1	Reserved
0	1	0	0	0	1	1	0	Reserved
0	1	0	0	0	1	1	1	Reserved
0	1	0	0	1	0	0	0	Reserved
0	1	0	0	1	0	0	1	Reserved
0	1	0	0	1	0	1	0	Reserved
0	1	0	0	1	0	1	1	RCMOFROUT ('K')
	.							.
	.							.
	.							.
0	1	0	1	0	0	0	1	Reserved

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Activity[a]
0	1	0	1	0	0	1	0	Reserved
0	1	0	1	0	0	1	1	RCMACSROUT ('S')
0	1	0	1	0	1	0	0	Reserved
			.					.
			.					.
			.					.
0	1	0	1	0	1	1	1	RCMFLOUT ('W')
1	1	1	1	1	1	1	1	Reserved

Note(s):
a. The ASCII characters that the values represent are shown in parentheses.

6.24 STATE

The state information field appears in the message data portion of the Status Check Response Message and in the message data portion of the Abort Response Message. It indicates the state of the previous request.

The valid codes for this field are as follows:

Bit								
7	6	5	4	3	2	1	0	State[a]
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	Reserved
0	0	1	1	0	0	0	1	RCM5ESEND ('1') Interface controller is in the process of sending the request to the subsystem
0	0	1	1	0	0	1	0	RCM5EWAIT ('2') Interface controller is waiting for data from the subsystem
0	0	1	1	0	0	1	1	RCM5ERECV ('3') Interface controller is receiving data from the subsystem
0	0	1	1	0	1	0	0	RCMIDLE ('4') Interface controller is waiting for a request from a remote processor

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	State[a]
0	0	1	1	0	1	0	1	RCMACRCV ('5') Interface controller is receiving a request from a remote processor
0	0	1	1	0	1	1	0	RCMACSND ('6') Interface controller is sending a response to a remote processor
0	0	1	1	0	1	1	1	RCMWORKING ('7') Interface controller is doing work internally
0	0	1	1	1	0	0	0	Reserved
0	0	1	1	1	0	0	1	Reserved
0	0	1	1	1	0	1	0	Reserved
				.				.
				.				.
				.				.
1	1	1	1	1	1	1	1	Reserved

Note(s):
a. The ASCII characters that the values represent are shown in parentheses.

6.25 ERROR CODE

The error code field specifies the reason for the error. Certain reply opcodes each have a unique set of error codes.

The length of error code is one octet. The value of error code is ASCII numeric with the exception of the error codes for reply opcode RCMRCRTRERR whose values are binary number.

6.25.1 MISCELLANEOUS ERROR CODES

The following error codes may be sent with any reply opcode.

Bit								
7	6	5	4	3	2	1	0	Error[a] Code
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	0	1	0	1	1	1	1	Reserved
0	0	1	1	0	0	0	0	RCMNOERRC ('0') (no error code for this failure)
0	0	1	1	0	0	0	1	RCMREQINV ('1') (invalid request type in header)

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Error[a] Code
0	0	1	1	0	0	1	0	RCMOPCINV ('2') (invalid opcode in header)
0	0	1	1	0	0	1	1	Reserved
0	0	1	1	0	1	0	0	Reserved
0	0	1	1	0	1	0	1	Reserved
			.					.
			.					.
			.					.
1	1	1	1	1	1	1	1	Reserved

Note(s):
a. The ASCII characters which the values represent are shown in parentheses.

6.25.2 ERROR CODES FOR REPLY OPCODE RCMRCERROR

The following error codes will be sent only if the reply opcode is set to "RCMRCERROR."

Bit								
7	6	5	4	3	2	1	0	Error [a]Code
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	1	0	0	0	0	0	0	Reserved
0	1	0	0	0	0	0	1	RCM_DERR ('A') "?D" - invalid data field
0	1	0	0	0	0	1	0	RCM_EERR ('B') "?E" with no "# of error messages"
0	1	0	0	0	0	1	1	RCMRC_EMSG ('C') "?E" with "# of error messages" (multiple errors detected)
0	1	0	0	0	1	0	0	RCMRC_TERR ('D') "?T" - time-out (inactivity timeout error)
0	1	0	0	0	1	0	1	RCMRC_LERR ('E') (length of form field is greater than 256 octets)
0	1	0	0	0	1	1	0	RCM_IERR ('F') "?I" - invalid ID field

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Error [a]Code
0	1	0	0	0	1	1	1	RCM_AERR ('G') "?A" - invalid action field
0	1	0	0	1	0	0	0	RCM_NGERR ('H') "NG" - no good, conflict with current system status
0	1	0	0	1	0	0	1	RCM_RLERR ('I') "RL" - retry later
0	1	0	0	1	0	1	0	RCM_UNKNERR ('J') unknown error response
0	1	0	0	1	0	1	1	Reserved
0	1	0	0	1	1	1	1	Reserved
0	1	0	1	0	0	0	0	Reserved
	.							.
	.							.
	.							.
1	1	1	1	1	1	1	1	Reserved

Note(s):
a. The ASCII characters that the values represent are shown in parentheses.

6.25.3 ERROR CODES FOR REPLY OPCODE RCM5EAERROR

The following error codes will be sent if only the reply opcode is set to "RCM5EAERROR."

Bit								
7	6	5	4	3	2	1	0	Error [a]Code
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Reserved
0	0	0	0	0	0	1	0	Reserved
			.					.
			.					.
			.					.
0	1	1	0	1	0	1	0	Reserved
0	1	1	0	1	0	1	1	RCM5E_CSERR ('k') The remote processor sent an abort request message to abort the request previously sent to the network, but the network is not working currently on the request with the same "opcode" that the remote processor wants to abort
0	1	1	0	1	1	0	1	Reserved

See note(s) at end of table.

Bit								
7	6	5	4	3	2	1	0	Error [a]Code
0	1	1	0	1	1	1	0	Reserved
0	1	1	0	1	1	1	1	Reserved
			.					.
			.					.
			.					.
1	1	1	1	1	1	1	1	Reserved

Note(s):

a. The ASCII characters that the values represent are shown in parentheses.

6.25.4 ERROR CODES FOR REPLY OPCODE RCMODERROR OR RCMRCINTERR

If the reply opcode is set to “RCMODERROR” or “RCMRCINTERR”, the message data is set to the null value (’[0]’).

6.25.5 ERROR CODES FOR REPLY OPCODE RCMRCRTRERR

The following error codes will be sent only if the reply opcode is set to “RCMRCRTRERR.”

Bit								
7	6	5	4	3	2	1	0	Error Code
0	0	0	0	0	0	0	0	Reserved
0	0	0	0	0	0	0	1	Not super-user
0	0	0	0	0	0	1	0	No such file or directory
0	0	0	0	0	0	1	1	No such process
0	0	0	0	0	1	0	0	Interrupted system call
0	0	0	0	0	1	0	1	I/O error
0	0	0	0	0	1	1	0	No such device or address
0	0	0	0	0	1	1	1	Argument list too long
0	0	0	0	1	0	0	0	Execution format error
0	0	0	0	1	0	0	1	Bad file number
0	0	0	0	1	0	1	0	No children
0	0	0	0	1	0	1	1	No more processes
0	0	0	0	1	1	0	0	Not enough core
0	0	0	0	1	1	0	1	Permission denied
0	0	0	0	1	1	1	0	Bad address
0	0	0	0	1	1	1	1	Block device required
0	0	0	1	0	0	0	0	Mount device busy
0	0	0	1	0	0	0	1	File exists
0	0	0	1	0	0	1	0	Cross-device link
0	0	0	1	0	0	1	1	No such device
0	0	0	1	0	1	0	0	Not a directory
0	0	0	1	0	1	0	1	Is a directory

Bit								
7	6	5	4	3	2	1	0	Error Code
0	0	0	1	0	1	1	0	Invalid argument
0	0	0	1	0	1	1	1	File table overflow
0	0	0	1	1	0	0	0	Too many open files
0	0	0	1	1	0	0	1	Not a typewriter
0	0	0	1	1	0	1	0	Text file busy
0	0	0	1	1	0	1	1	File too large
0	0	0	1	1	1	0	0	No space on device
0	0	0	1	1	1	0	1	Illegal seek
0	0	0	1	1	1	1	0	Read-only file system
0	0	0	1	1	1	1	1	Too many links
0	0	1	0	0	0	0	0	Broken pipe
0	0	1	0	0	0	0	1	Temped file
0	0	1	0	0	0	1	0	Reserved
0	0	1	0	0	0	1	1	Reserved
0	0	1	0	0	1	0	0	Reserved
			.					.
			.					.
			.					.
1	1	1	1	1	1	1	1	Reserved

6.26 ERROR MESSAGE

The error message describes the reason for the error or the result of an event.

6.26.1 ERROR MESSAGES FOR OFR OUTPUT RESPONSE MESSAGE

The following error messages may be sent to respond to an OFR Output Request Message.

	Error Message
1	AN INVALID ERROR FILE BYTES PARAMETER WAS FOUND FOR REQID: xxx
2	THE ERROR FILE BYTES SPECIFIED EXCEEDS THE FILE SIZE FOR REQID: xxx
3	AN INVALID OUTPUT FILE BYTES PARAMETER WAS FOUND FOR REQID: xxx
4	THE OUTPUT FILE BYTES SPECIFIED EXCEEDS THE FILE SIZE FOR REQID: xxx
5	THE REQID IS NOT WITHIN THE VALID RANGE OF REQID'S
6	THE PROCESSING OF THE REQUEST(S) HAS BEEN COMPLETED
7	THE TRANSMISSION OF THE RC/OS ERROR FILE HAS FINISHED FOR REQID: xxx
8	THE TRANSMISSION OF THE RC/OS OUTPUT FILE HAS FINISHED OP REQID: xxx
9	INTERNAL OFR ERROR - CAN'T CREATE TEMPORARY ERROR FILE FOR REQID: xxxx
10	INTERNAL OFR ERROR - CAN'T CREATE TEMPORARY OUTPUT FILE FOR REQID: xxx
11	INTERNAL OFR ERROR - CAN'T OPEN TEMPORARY ERROR FILE FOR REQID: xxx
12	INTERNAL OFR ERROR - CAN'T OPEN TEMPORARY OUTPUT FILE FOR REQID: xxx
13	INTERNAL OFR ERROR - CAN'T WRITE TO TEMPORARY ERROR FILE FOR REQID: xxx
14	INTERNAL OFR ERROR - CAN'T WRITE TO TEMPORARY OUTPUT FILE FOR REQID: xxx
15	MESSAGE CONTAINING REQID, OUTBYTES AND ERRBYTES IS INVALID
16	SKIPPING BOTH OUTPUT AND ERROR FILES RETURNS NOTHING FOR REQID: xxx
17	AN INVALID OFR REQUEST TYPE WAS SPECIFIED
18	SYSTEM ERROR WHILE ATTEMPTING TO LINK THE ERROR FILE FOR REQID: xxx
19	SYSTEM ERROR WHILE ATTEMPTING TO LINK THE OUTPUT FILE FOR REQID: xxx
20	LSEEK ERROR IN OFR ERROR FILE FOR REQID: xxx
21	LSEEK ERROR IN OFR OUTPUT FILE FOR REQID: xxx

	Error Message
22	THE INPUT MESSAGE FOR OFR OUTPUT REQUIRES AN ENDING CARRIAGE RETURN
23	THE STARTING ERROR FILE BYTE POSITION WAS NOT FOUND FOR REQID: xxx
24	REQUESTED OFR ERROR FILE WAS BLANK FOR REQID: xxx
25	THE STARTING OUTPUT FILE BYTE POSITION WAS NOT FOUND FOR REQID: xxx
26	INTERNAL OFR ERROR - NO OUTPUT FILE WAS FOUND. THE ERROR FILE WILL FOLLOW FOR REQID: xxx
27	REQUESTED OFR OUTPUT WAS BLANK FOR REQID: xxx
28	THE OFFICE RECORD PROCESSING HAS NOT STARTED OR DOES NOT EXIST FOR REQID: xxx
29	THE REQUESTED OFFICE RECORD PROCESSING IS IN PROGRESS FOR REQID: xxx
30	ERROR IN DETERMINING THE SIZE OF THE OFR ERROR FILE FOR REQID: xxx
31	ERROR IN DETERMINING THE SIZE OF THE OFR OUTPUT FOR REQID: xxx

Recent Change Operations Systems Interface Specification

CONTENTS PAGE

7. REFERENCES

The following documents are referenced in this document:

- PUB.54001, *Operations Systems Network Communications Protocol Specification BX.25*, Issue 3
- PUB.54001A, *Operations Systems Network Communications Protocol Specification BX.25*, Issue 3, Addendum A
- Document 235-118-251, **5ESS®**-*2000 Switch Recent Change Procedures, 5E11 and Later Software Releases*
- Document 235-600-700, **5ESS**-*2000 Switch Input Messages*
- Document 235-600-750, **5ESS**-*2000 Switch Output Messages*.

ABBREVIATIONS AND ACRONYMS

ACSR	Automatic Customer Station Rearrangement
AFT	APPTEXT Forward Translator
APPROC	Applications Processor Form
BH	Batch History
BMI	Batch Mode Input
BMR	Batch Mode Release
BRC	Batch Recent Change
BRCSFD	Business and Residence Custom Services Feature Definition Forms
CD	Carrier Detect
CDF	Control Data Flag
CDC	Cyclic Redundancy Check
CTS	Clear To Send
D-bit	Delivery Confirmation Bit
DC	Downward Compatibility
DSR	Data Set Ready
DTR	Data Terminal Ready
EIA	Electronics Industry Association
IM	Immediate Mode
IOP	Input/Output Port
IRC	Immediate Recent Change
ISDN	Integrated Services Digital Network
LAPB	Balanced Link Access Procedure
M	Mandatory
M-bit	More Data Bit
MDF	More Data Flag
MML	Man-Machine Language
O	Optional
OFR	Office Records
OSPS	Operator Services Position System
PDS	Program Documentation Standards
PVC	Permanent Virtual Circuits
Q-bit	Qualified Data Bit
RC/OS	Recent Change/Operations Systems
RD	Received Data
RNR	Receive Not Ready

RR	Receive Ready
RTS	Request To Send
SADM	Session Abort Disconnect Message
SAM	Session Accept Message
SCM	Session Connect Message
SNDM	Session Normal Disconnect Message
SNDMRM	Session Normal Disconnect Message Reject Message
SPDU	Session Protocol Data Unit
SRM	Session Reject Message
SVC	Switched Virtual Circuits
TD	Transmitted Data
VFY	Verify

A

B

C

D

N

O

P

Q

R

S

T

U

V

W