

**3270 Information Display System**  
**Data Stream Programmer's Reference**  
**GA23-0059-07**

Front matter

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Chapter 9

Chapter 10

Chapter 11

Chapter 12

Chapter 13

Appendix A

Appendix B

Appendix C

Appendix D

Appendix E

Appendix F

End matter

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## COVER Book Cover

---

3270 Information Display System

Data Stream Programmer's Reference

Document Number GA23-0059-07

File Number 36/38/370/4300/8100/3174-30

---

## Notices

### 3174 Licensed Internal Code

Licensed Internal Code remains the property of IBM and is to be used only in accordance with the Agreement for Purchase of IBM Machines or the Agreement for Lease or Rental of IBM Machines, whichever Agreement is applicable.

In this document, the term **3174 microcode** refers to Licensed Internal Code.

---

## EDITION Edition Notice

### Eighth Edition (June 1992)

This major revision obsoletes and replaces GA23-0059-06. See "Summary of Changes" in topic PREFACE.3 for the changes made to this book. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the changes.

Changes are made periodically to the information herein. Before using this publication in connection with the operation of IBM systems or equipment, consult your IBM sales representative or the latest *IBM System/370, 30xx, and 4300 Processors Bibliography*, GC20-0001, for editions that are applicable and current.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A Reader's Comment Form is provided at the back of this publication. If the form has been removed, address comments to IBM Corporation, Communication Products Information Development, Department E02, PO Box 12195, Research Triangle Park, North Carolina, U.S.A. 27709. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever.

© Copyright International Business Machines Corporation 1981, 1992.  
All rights reserved.

Note to U.S. Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

---

# CONTENTS Table of Contents

## [Summarize]

COVER	<u>Book Cover</u>
NOTICES	<u>Notices</u>
EDITION	<u>Edition Notice</u>
CONTENTS	<u>Table of Contents</u>
FIGURES	<u>Figures</u>
TABLES	<u>Tables</u>
FRONT_1	<u>Notices</u>
PREFACE	<u>Preface</u>
PREFACE.1	<u>Who Should Use This Book</u>
PREFACE.2	<u>How This Book Is Organized</u>
PREFACE.3	<u>Summary of Changes</u>
PREFACE.4	<u>Where To Find More Information</u>
1.0	<u>Chapter 1. Introduction to the 3270 Data Stream</u>
1.1	<u>Introduction</u>
1.2	<u>Format of the 3270 Data Stream</u>
1.3	<u>The Device Buffer</u>
1.4	<u>Attributes</u>
1.5	<u>Commands</u>
1.6	<u>Orders</u>
1.7	<u>Structured Fields</u>
2.0	<u>Chapter 2. Partitions</u>
2.1	<u>Introduction</u>
2.2	<u>Presentation Space, Partitions, Windows, and Viewports</u>
2.3	<u>Implicit and Explicit Partitions</u>
2.4	<u>Using Multiple Partitions</u>

2.5	<u>Activate Partition</u>
2.6	<u>Managing the Presentation Space</u>
2.7	<u>Partition Wait Condition (PWAIT)</u>
2.8	<u>System Lock Condition</u>
2.9	<u>Terminal Wait (TWAIT) Condition</u>
3.0	<u>Chapter 3. 3270 Data Stream Commands</u>
3.1	<u>Introduction</u>
3.2	<u>Commands within Structured Fields</u>
3.3	<u>Command Codes</u>
3.4	<u>Write Control Character (WCC) Byte</u>
3.5	<u>Write Operation</u>
3.6	<u>Read Operations</u>
3.7	<u>Read Operations from Partitions</u>
3.8	<u>Read Operations for SNA</u>
3.9	<u>Inbound Operation (INOP)</u>
3.10	<u>Inbound Partition Identifier (INPID)</u>
3.11	<u>Enter Actions</u>
3.12	<u>Processing of Read Commands (Alphanumeric)</u>
4.0	<u>Chapter 4. Character Sets, Orders, and Attributes</u>
4.1	<u>Introduction</u>
4.2	<u>Character Sets</u>
4.3	<u>Orders</u>
4.4	<u>Attributes</u>
4.5	<u>Processing of Character Attributes</u>
4.6	<u>Defaults for Attributes</u>
5.0	<u>Chapter 5. Outbound/Inbound and Outbound Structured Fields</u>
5.1	<u>Introduction</u>
5.2	<u>Structured Field Grouping</u>
5.3	<u>Structured Field Self-Defining Parameters</u>
5.4	<u>Outbound Structured Fields</u>
5.5	<u>Activate Partition</u>
5.6	<u>Begin/End of File</u>
5.7	<u>Create Partition</u>
5.8	<u>Destroy Partition</u>
5.9	<u>Erase/Reset</u>
5.10	<u>Load Color Table</u>
5.11	<u>Load Format Storage</u>
5.12	<u>Load Line Type</u>
5.13	<u>Load Programmed Symbols (Load PS)</u>
5.14	<u>Modify Partition</u>
5.15	<u>Outbound Text Header</u>
5.16	<u>Outbound 3270DS</u>
5.17	<u>Present Absolute Format</u>
5.18	<u>Present Relative Format</u>
5.19	<u>Read Partition</u>
5.20	<u>Request Recovery Data</u>
5.21	<u>Reset Partition</u>
5.22	<u>Restart</u>
5.23	<u>SCS Data</u>
5.24	<u>Select Color Table</u>
5.25	<u>Select Format Group</u>
5.26	<u>Set Checkpoint Interval</u>
5.27	<u>Set MSR Control</u>
5.28	<u>Set Partition Characteristics</u>
5.29	<u>Set Printer Characteristics</u>
5.30	<u>Set Reply Mode</u>
5.31	<u>Set Window Origin</u>
5.32	<u>Type 1 Text Outbound</u>
5.33	<u>Outbound/Inbound Structured Fields</u>
5.34	<u>Data Chain</u>
5.35	<u>Destination/Origin</u>
5.36	<u>Object Control</u>

5.37 Object Data  
5.38 Object Picture  
5.39 OEM Data  
5.40 Save/Restore Format  
5.41 Select Intelligent Printer Data Stream (IPDS) Mode

6.0 Chapter 6. Inbound Structured Fields  
6.1 Introduction  
6.2 Exception/Status  
6.3 Inbound Text Header  
6.4 Inbound 3270DS  
6.5 Recovery Data  
6.6 Type 1 Text Inbound  
6.7 Query Replies  
6.8 Query Reply  
6.9 Query Reply (Alphanumeric Partitions)  
6.10 Query Reply (Auxiliary Device)  
6.11 Query Reply (Begin/End of File)  
6.12 Query Reply (Character Sets)  
6.13 Query Reply (Color)  
6.14 Query Reply (Cooperative Processing Requestor)  
6.15 Query Reply (Data Chaining)  
6.16 Query Reply (Data Streams)  
6.17 Query Reply (DBCS-Asia)  
6.18 Query Reply (Device Characteristics)  
6.19 Query Reply (Distributed Data Management)  
6.20 Query Reply (Document Interchange Architecture)  
6.21 Query Reply (Extended Drawing Routine)  
6.22 Query Reply (Field Outlining)  
6.23 Query Reply (Field Validation)  
6.24 Query Reply (Format Presentation)  
6.25 Query Reply (Format Storage Auxiliary Device)  
6.26 Query Reply (Graphic Color)  
6.27 Query Reply (Graphic Symbol Sets)  
6.28 Query Reply (Highlighting)  
6.29 Query Reply (IBM Auxiliary Device)  
6.30 Query Reply (Image)  
6.31 Query Reply (Implicit Partition)  
6.32 Query Reply (IOCA Auxiliary Device)  
6.33 Query Reply (Line Type)  
6.34 Query Reply (MSR Control)  
6.35 Query Reply (Null)  
6.36 Query Reply (OEM Auxiliary Device)  
6.37 Query Reply (Paper Feed Techniques)  
6.38 Query Reply (Partition Characteristics)  
6.39 Query Reply (Port)  
6.40 Query Reply (Procedure)  
6.41 Query Reply (Product Defined Data Stream)  
6.42 Query Reply (Reply Modes)  
6.43 Query Reply (RPQ Names)  
6.44 Query Reply (Save/Restore Format)  
6.45 Query Reply (Segment)  
6.46 Query Reply (Settable Printer Characteristics)  
6.47 Query Reply (Storage Pools)  
6.48 Query Reply (Summary)  
6.49 Query Reply (Text Partitions)  
6.50 Query Reply (Transparency)  
6.51 Query Reply (Usable Area)  
6.52 Query Reply (3270 IPDS)

7.0 Chapter 7. Magnetic Slot Reader, Keyboard, and Selector Pen Operations  
7.1 Introduction  
7.2 Magnetic Slot Reader Operations  
7.3 Keyboard Operations  
7.4 Selector Pen Operation

8.0	<u>Chapter 8. Printer Operations</u>
8.1	<u>Introduction</u>
8.2	<u>Local Copy Function in an SNA Environment</u>
8.3	<u>Copy Initiation</u>
8.4	<u>Printer Availability</u>
8.5	<u>Display/Printer Compatibility</u>
8.6	<u>Partition Mode Considerations</u>
8.7	<u>Local Copy Command in the BSC Environment</u>
8.8	<u>Valid LU 1 SCS Control Codes</u>
8.9	<u>Early Print Complete (EPC) Operations</u>
8.10	<u>IPDS Data/Non-SNA</u>
9.0	<u>Chapter 9. Binary Synchronous Communications (BSC) Environment</u>
9.1	<u>Introduction</u>
9.2	<u>Transparent Mode</u>
9.3	<u>Write Commands</u>
9.4	<u>Read Commands</u>
9.5	<u>Inbound Transmissions</u>
9.6	<u>Inbound Operation (INOP)</u>
9.7	<u>Read States</u>
9.8	<u>Indicators</u>
9.9	<u>Read Acknowledgments</u>
9.10	<u>Processing of Read Commands</u>
9.11	<u>Processing of Read Partition Query Structured Fields</u>
9.12	<u>BSC Copy Command</u>
10.0	<u>Chapter 10. Non-SNA Environment (Locally Attached Devices--3272 Version)</u>
10.1	<u>Introduction</u>
10.2	<u>Commands</u>
10.3	<u>Inbound Transmissions</u>
10.4	<u>Inbound Operation (INOP)</u>
10.5	<u>Read States</u>
10.6	<u>Indicators</u>
10.7	<u>Read Acknowledgments - Non-SNA</u>
10.8	<u>Processing of Read Commands</u>
10.9	<u>Processing of Read Partition Query Structured Fields</u>
11.0	<u>Chapter 11. Auxiliary Devices and Workstations</u>
11.1	<u>Introduction</u>
11.2	<u>Data Routing</u>
11.3	<u>Query Reply</u>
11.4	<u>Input Control</u>
11.5	<u>Auxiliary Device and Display Interaction</u>
11.6	<u>Exception Handling</u>
12.0	<u>Chapter 12. Double-Byte Coded Character Set (DBCS)-Asia</u>
12.1	<u>Introduction</u>
12.2	<u>Codepoints</u>
12.3	<u>DBCS Fields</u>
12.4	<u>DBCS Character Attribute (SA)</u>
12.5	<u>Shift Out (SO)/Shift In (SI)</u>
12.6	<u>Set Attribute (SA) Order and SO/SI Interaction</u>
12.7	<u>Orders</u>
13.0	<u>Chapter 13. Local Format Storage</u>
13.1	<u>Introduction</u>
13.2	<u>Architecture Requirements</u>
13.3	<u>Format Distribution</u>
13.4	<u>Sample Sequences</u>
13.5	<u>Format Selection</u>
13.6	<u>User Selected Formats</u>
A.0	<u>Appendix A. SNA Sense Codes</u>

B.0	<u>Appendix B. Reset Actions</u>
C.0	<u>Appendix C. 12-, 14-, and 16-Bit Addressing</u>
D.0	<u>Appendix D. Special Applications</u>
D.1	<u>Introduction</u>
D.2	<u>Query Reply (Anomaly Implementation)</u>
E.0	<u>Appendix E. Functions Required for Systems Application Architecture (SAA) S</u>
E.1	<u>Introduction</u>
E.2	<u>Query Replies</u>
E.3	<u>Structured Fields</u>
E.4	<u>Basic 3270 Commands</u>
E.5	<u>Basic 3270 Orders</u>
E.6	<u>3270 Format Control Orders</u>
F.0	<u>Appendix F. Hexadecimal Index</u>
F.1	<u>Introduction</u>
F.2	<u>Commands</u>
F.3	<u>Orders</u>
F.4	<u>Format Control Orders</u>
F.5	<u>Structured Fields (One-Byte ID)</u>
F.6	<u>Structured Fields (Two-Byte ID)</u>
F.7	<u>SCS Control Codes</u>
ABBREVIATIONS	<u>List of Abbreviations</u>
GLOSSARY	<u>Glossary</u>
INDEX	<u>Index</u>
COMMENTS	<u>Readers' Comments</u>

---

## Figures

1-1.	<u>Mapping the Display to the Character Buffer</u>	1.3
1-2.	<u>Example of Formatted Fields</u>	1.3
1-3.	<u>Field, Extended Field, and Character Attributes</u>	1.4
2-1.	<u>Presentation Space and Viewport (without Scrolling)</u>	2.2
2-2.	<u>The Presentation Plane--A Conceptual View</u>	2.2
2-3.	<u>Presentation Space, Window, and Viewport (with Scrolling)</u>	2.2.1
2-4.	<u>Management of Presentation Spaces</u>	2.6
5-1.	<u>Structured-Field Format</u>	5.1
5-2.	<u>Vertical and Horizontal Slicing of a Character Cell</u>	5.13.8
5-3.	<u>Type 1 Data Format -- An Example Dot Pattern Encoded</u>	5.13.8
5-4.	<u>Example of Compression Algorithm Using Comparison Rule 1</u>	5.13.12
5-5.	<u>Example of Compression Algorithm Using Comparison Rule 2</u>	5.13.12
5-6.	<u>Example of Compression Algorithm Using Comparison Rule 3</u>	5.13.12
6-1.	<u>Steps Used in Interpreting Load PS Data Stream</u>	6.12.4.3
7-1.	<u>Display Screen Activity Before and After Magnetic-Stripe Input (Unformatted Display)</u>	7.2.1
7-2.	<u>Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Unprotected Field Attribute)</u>	7.2.1
7-3.	<u>Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Protected Field Attribute)</u>	7.2.1
8-1.	<u>The Copy Data Stream</u>	8.7
C-1.	<u>Conversion of Binary Values to Hexadecimal Values that Obtain Graphic Symbols</u>	C.0

---

## Tables

1-1.	Attribute Pairs	1.4.3
3-1.	Command Codes and Abbreviations	3.3
3-2.	Write Control Character (WCC) Bit Definitions for Displays	3.4
3-3.	Write Control Character (WCC) Reset Actions for Displays	3.4
3-4.	Attention Identification (AID) Bytes Sent from the Display to the Application Program	3.5.6
3-5.	Normal Read State Process	3.8.5.1
3-6.	Retry Enter State Process	3.8.5.2
3-7.	Retry Read State Process	3.8.5.3
3-8.	Read-State Transitions	3.8.5.4
4-1.	Order Codes	4.3
4-2.	Example of Attribute Type-Value Pairs	4.3.4
4-3.	Format Control Orders	4.3.11
4-4.	Bit Definitions for 3270 Field Attributes	4.4.1
4-5.	Rules for Resolving Attribute Conflicts	4.4.4
4-6.	Attribute Types	4.4.5
4-7.	Color Identifications	4.4.6.4
4-8.	Attribute Default Conditions	4.6
6-1.	Query Responses	6.7.3
8-1.	Copy Control Character (CCC)	8.7
8-2.	Valid LU 1 SCS Control Codes	8.8
8-3.	LU 1 BIND Byte 18 (Data Stream Flags)	8.8
9-1.	Read State Transitions for BSC	9.7.3
10-1.	Read-State Transitions for Non-SNA Locally Attached Devices--3272 Version	10.5.3
A-1.	SNA Sense Codes	A.0
B-1.	Reset Actions - Part 1	B.0
B-2.	Reset Actions - Part 2	B.0

---

## FRONT\_1 Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

---

## Preface

This book describes 3270 data stream operations; however, your particular device may not implement the described function. Consult your product library for information about the 3270 data stream functions your product uses.

Subtopics:

- PREFACE.1 Who Should Use This Book
- PREFACE.2 How This Book Is Organized
- PREFACE.3 Summary of Changes
- PREFACE.4 Where To Find More Information

---

## PREFACE.1 Who Should Use This Book

This book is for the programmers who need to know what is involved in using the 3270 data stream to produce panels or information.

Programmers who write the access method macro instructions or other input/output (I/O) instructions should use this book in conjunction with the appropriate access method or IBM program product publications.

This book is also for those programmers who plan and code for support of Systems Application Architecture (SAA). This architecture is supported by the 3270 data stream and this book contains information about the architecture and its relationship to the 3270 data stream.

---

## **PREFACE.2 How This Book Is Organized**

This book has thirteen chapters and six appendixes:

*Chapter 1:* Introduces the 3270 data stream, gives an overview of it, and discusses 3270 data stream concepts.

*Chapter 2:* Describes partitions and functions related to partitions.

*Chapter 3:* Describes the commands used in the 3270 data stream and their operations. It also designates the specific commands that are necessary for SAA support.

*Chapter 4:* Describes the orders and attributes used in the 3270 data stream and how they function. It also explains character set properties and identifies specific orders that are necessary for SAA support.

*Chapter 5:* Describes the outbound and outbound/inbound structured fields in alphabetic order, gives their syntax, and describes how they function. The chapter also designates outbound structured fields that are necessary for SAA support.

*Chapter 6:* Describes the inbound structured fields, including Query Replies, in alphabetic order, gives their syntax, and describes how they function. This chapter also designates inbound structured fields that are necessary for SAA support.

*Chapter 7:* Describes how magnetic readers work with the 3270 data stream, the keyboard functions that affect data stream operation, and the use of the selector pen.

*Chapter 8:* Describes printer functions with the 3270 data stream and the Local Copy function for Systems Network Architecture (SNA) and binary synchronous communications (BSC).

*Chapter 9:* Discusses a BSC environment and describes the differences in

operation between a BSC environment and an SNA environment for the 3270 data stream.

*Chapter 10:* Discusses a non-SNA environment of locally attached devices (3272 version) and describes the differences in operation in an SNA or BSC environment.

*Chapter 11:* Describes the enhancements of the 3270 data stream that allow support of 3270 data stream workstations.

*Chapter 12:* Describes the 2-byte coded character set defined for use in South East Asia Region (SEAR) countries, Japan, Korea, and Taiwan.

*Chapter 13:* Describes the Local Format Storage function.

*Appendix A:* Summarizes the sense codes returned for data stream command, order, and structured field errors.

*Appendix B:* Summarizes the reset actions for the 3270 data stream.

*Appendix C:* Explains the addressing used in the 3270 data stream.

*Appendix D:* Describes how applications that do not implement 3270 data stream operations report those deviations to the host.

*Appendix E:* Describes the functions that the 3270 data stream must support for SAA.

*Appendix F:* Indexes all commands, orders, format control orders, structured fields, and SCS control codes by hexadecimal value.

---

## **PREFACE.3 Summary of Changes**

Subtopics:

- PREFACE.3.1 Eighth Edition (June 1992)

---

### **| PREFACE.3.1 Eighth Edition (June 1992)**

| Information has been changed or added in this edition to reflect

| architectural changes in the 3270 data stream for Read operations, Zero Length Structured Fields, Set Reply Mode Structured Fields, Early Print Complete operations, and the Delete Key.

| Changes are indicated by a vertical line to the left of the changed text.

---

## **PREFACE.4 Where To Find More Information**

Subtopics:

- PREFACE.4.1 3270 General Information
- PREFACE.4.2 Systems Application Architecture
- PREFACE.4.3 Coded Character Sets
- PREFACE.4.4 Document Interchange Architecture
- PREFACE.4.5 3174 Establishment Controller

---

### **| PREFACE.4.1 3270 General Information**

| The following publications provide a general introduction to the 3270 Information Display System:

| For access methods, refer to the appropriate access method library used in your environment.

- | ° *IBM 3270 Information Display System  
3174 Establishment Controller Introduction, GA27-3850*
  
- | ° *IBM 3270 Information Display System Color and Programmed Symbols,  
GA33-3056*
  
- | ° *IBM 3270 Information Display System  
3271 Control Unit  
3272 Control Unit  
3275 Display Station  
Description and Programmer's Guide, GA23-0060*
  
- | ° *IBM 3270 Information Display System  
3274 Control Unit  
Description and Programmer's Guide, GA23-0061*
  
- | ° *IBM 3270 Information Display System  
3276 Control Unit Display Station  
Description and Programmer's Guide, GA18-2081*
  
- | ° *IBM 5080 Graphics System Principles of Operation, GA23-2012*
  
- | ° *IBM AIX Workstation Host Interface Program  
User's Guide and Reference, Version 1.1, SC23-2060*
  
- | ° *IBM Systems Reference Library*

| *General Information - Binary Synchronous Communications*, GA27-3004

| ° *IBM 3179 Color Display Station Description*, GA18-2177

| ° *IBM 3192 Display Station Description*, GA18-2535

| ° *IBM 3193 Display Station Description*, GA18-2364

| ° *IBM 3194 Device Functional Interface Programming Guide*, SA23-0314

| ° *IBM 3278 Display Station Description*, GA18-2127

| ° *IBM 3290 Information Panel Description and Reference*, GA23-0021

| ° *IBM Office Information Architectures: Concepts*, GC23-0765.

---

## | **PREFACE.4.2 Systems Application Architecture**

| The following publications provide a general introduction to Systems  
| Application Architecture (SAA):

| ° *An Overview*, GC26-4341

| ° *Common User Access: Panel Design and User Interaction*, SC26-4351

| ° *Writing Applications: A Design Guide*, SC26-4362

| ° *Application Generator Reference*, SC26-4355

| ° *C Reference*, SC26-4353

| ° *COBOL Reference*, SC26-4354

| ° *Communications Reference*, SC26-4399

| ° *Database Reference*, SC26-4348

| ° *Dialog Reference*, SC26-4356

| ° *FORTTRAN Reference*, SC26-4357

| ° *Presentation Reference*, SC26-4359

| ° *Procedures Language Reference*, SC25-4358

- | ° *Query Reference, SC26-4349*
  
  - | ° *Systems Application Architecture: Common Communications Support Summary, GC31-6810*
  
  - | ° *Font Object Content Architecture Reference, S544-3285*
  
  - | ° *Intelligent Printer Data Stream Reference, S544-3417.*
- 

## **| PREFACE.4.3 Coded Character Sets**

| The following publications contain information on valid Coded Character Set Identifier parameter values and other aspects of Character Data Representation Architecture:

- | ° *Character Data Representation Architecture - Level 1, Reference, SC90-1390*
  
  - | ° *Character Data Representation Architecture - Level 1, Executive Overview, SC90-1392*
  
  - | ° *Character Data Representation Architecture - Level 1, Registry, SC90-1391.*
- 

## **| PREFACE.4.4 Document Interchange Architecture**

| The following publications provide a description of the Document Interchange Architecture functions:

- | ° *Document Interchange Architecture: Technical Reference, SC23-0781*
  
  - | ° *Document Interchange Architecture: Interchange Document Profile Reference, SC23-0764*
  
  - | ° *Document Interchange Architecture: Transaction Programmer's Guide, SC23-0763.*
- 

## **| PREFACE.4.5 3174 Establishment Controller**

| The 3174 library contains information for installing, customizing, operating, maintaining, and programming the data stream for the 3174 Establishment Controller. The list below shows the manuals you need to perform these tasks.

| **To Find Translations of Safety Notices:**

- | *Safety Notices, GA27-3824*

**To Organize Library Materials:**

Binders and Inserts, SBOF-0089  
Binder, SX23-0331  
Inserts, SX23-0332

**To Become Familiar with the 3174:**

*Master Index*, GC30-3515  
*3174 Introduction*, GA27-3850

**To Prepare Your Site for the 3174:**

*Site Planning*, GA23-0213  
*Physical Planning Template*, GX27-2999

**To Set Up and Operate the 3174:**

*Models 1L, 1R, 2R, 3R, 11L, 11R, 12L, 12R, and 13R User's Guide*,  
GA23-0337  
*Models 21L, 21R, 22L, and 23R User's Guide*, GA27-3874  
*Models 51R, 52R, 53R, 61R, 62R, and 63R User's Guide*, GA23-0333  
*Models 81R, 82R, 90R, 91R, and 92R User's Guide*, GA23-0313

**To Plan for and Customize the 3174:**

Configuration Support A and S

*Planning Guide*, GA27-3844  
*Utilities Guide*, GA27-3853  
*Central Site Customizing User's Guide*, GA23-0342  
*Asynchronous Emulation Adapter Description and Reference*, GA27-3872

Configuration Support B

*Planning Guide*, GA27-3862  
*Model 90R Tokenway Planning*, GD21-0036  
*Utilities Guide*, GA27-3863  
*Central Site Customizing User's Guide*, GA27-3868  
*Asynchronous Emulation Adapter Description and Reference*, GA27-3872

Configuration Support C

*Planning Guide*, GA27-3918  
*Utilities Guide*, GA27-3920  
*Central Site Customizing User's Guide*, GA27-3919  
*Asynchronous Emulation Adapter Description and Reference*, GA27-3872  
*Configuration Support C Release 2.0 User's Guide*, GA27-3966

**To Perform Problem Determination:**

*Customer Problem Determination, GA23-0217*  
*Status Codes, GA27-3832*  
*Configuration Support C Release 2.0 User's Guide, GA27-3966*

**To Install Features or Convert Models on the 3174:**

*Fixed Disk Installation and Removal Instructions, GA27-3864*  
*Diskette Drive Installation and Removal Instructions, GA23-0263*  
*Device Control Adapters Installation and Removal Instructions, GA23-0265*  
*Model Conversion Instructions, GA23-0295*  
*Token-Ring Network Feature Installation and Removal Instructions, GA23-0329*  
*Storage Expansion Feature Installation and Removal Instructions, GA23-0330*  
*Communication Adapter Installation and Removal Instructions, GA27-3830*  
*Asynchronous Emulation Adapter Installation and Removal Instructions, GA23-0341*  
*Concurrent Communication Adapter and Integrated Services Digital Network Adapter Installation and Removal Instructions, GA27-3851*  
*Models 21L, 21R, 22L, and 23R Feature Installation and Removal Instructions, GA27-3875*

**To Use the Asynchronous Emulation Adapter Feature:**

*Asynchronous Emulation Adapter Description and Reference, GA27-3872*  
*Terminal User's Reference for Expanded Functions, GA23-0332*

**To Use the Multiple Logical Terminals Function:**

*Terminal User's Reference for Expanded Functions, GA23-0332*

**To Obtain Data Stream Programming and Reference Information:**

*Functional Description, GA23-0218*  
*Data Stream Programmer's Reference, GA23-0059*  
*Asynchronous Emulation Adapter Description and Reference, GA27-3872*  
*3174 Reference Summary, GX27-3872*  
*3174 Character Set Reference, GA27-3831*  
*3270 X.25 Operation, GA23-0204*  
*Configuration Support C Release 2.0 User's Guide, GA27-3966*

**To Perform Maintenance (Service Personnel):**

*Models 1L, 1R, 2R, 3R, 11L, 11R, 12L, 12R, 13R Maintenance Information, SY27-2572*  
*Models 21L, 21R, 22L, and 23R Maintenance Information, SY27-0323*  
*Models 51R, 52R, 53R, 61R, 62R, and 63R Maintenance Information, SY27-2573*  
*Models 81R, 82R, 90R, 91R, and 92R Maintenance Information, SY27-2584*  
*CE Reference Summary, SX27-3873*  
*Status Codes, GA27-3832*

**Other Publications**

| The following publications are available for the 3174. They are developed  
| by the International Technical Support Center. The intended audience for  
| these books are IBM System Engineers and Customer Network Planners.

| *IBM 3174 Establishment Controller Installation Guide*, GG24-3061.

| *NetView Distribution Manager Release 2 and 3174 Central Site Change  
| Management Implementation Guide*, GG24-3424.

| *IBM 3174 CECP Migration Issues*, GG24-3380.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 1.0 Chapter 1. Introduction to the 3270 Data Stream

Subtopics:

- 1.1 Introduction
- 1.2 Format of the 3270 Data Stream
- 1.3 The Device Buffer
- 1.4 Attributes
- 1.5 Commands
- 1.6 Orders
- 1.7 Structured Fields

---

### 1.1 Introduction

The 3270 data stream controls the processing and formatting of data with commands, orders, control characters, attributes, and structured fields. This chapter describes the format and content of the 3270 data stream. It also describes the importance of the device buffer in sending and receiving data.

The 3270 data stream operations are designed primarily for transmitting data between an application program and a display with a keyboard. This book describes the 3270 data stream operations as they apply to displays. With certain exceptions, the operations described in this book also apply to printers. These exceptions are described in Chapter 8, "Printer Operations."

This book describes how the 3270 data stream operates in a Systems Network Architecture (SNA) environment. With certain exceptions, the operations described in this book also apply to Binary Synchronous Communications (BSC) and other non-SNA environments. These exceptions are described in Chapter 9, "Binary Synchronous Communications (BSC) Environment," and Chapter 10, "Non-SNA Environment (Locally Attached Devices--3272 Version)."

---

### 1.2 Format of the 3270 Data Stream

The data stream transmitted between an application program and a terminal

has the following format for SNA:

LH	TH	RH	3270 Data Stream	LT
----	----	----	------------------	----

The LH, TH, RH, and LT are the link header, transmission header, request/response header, and link trailer, respectively. The data stream can be *outbound* or *inbound*. An *outbound* data stream is a data stream sent from the application program to the device and has the following format:

Command	Write Control Character	Data
---------	-------------------------	------

OR

Write Structured Field	Structured Field	....	Structured Field
------------------------	------------------	------	------------------

An *inbound* data stream is sent from the device to the application program and consists of an attention identifier (AID) followed by data or an AID (X'88') followed by structured fields. The AID describes the action that caused the transmission of the inbound data stream.

The inbound data stream has the following format:

AID	Cursor Address (2 bytes)	Data
-----	-----------------------------	------

OR

AID (X'88')	Structured Field	....	Structured Field
----------------	------------------	------	------------------

The *data* shown in this format is the information transferred between the application program and the display. It can be used or modified by either the application program, the operator, or both. In many cases it is optional.

---

## 1.3 The Device Buffer

The 3270 data stream depends on the presence of a mapped character buffer in the device. Data received from the application program and data to be transmitted to the application program are stored in a device buffer and

displayed on the screen in the form of alphanumeric characters and symbols. The displayed data is updated when the operator modifies the buffer data and when new data is received from the application program.

Each character storage location in the buffer maps to a character position on the display. A character entered at the keyboard is stored in the display buffer (and is displayed) at the cursor position. Then the cursor advances one position and is ready for the next character to be entered. Before a character can be entered, the cursor must be positioned in an unprotected field. If a character already exists at the current cursor (and device buffer) position, that character is overwritten by the new character (except in insert mode).

For example, if the display screen has 12 rows and 80 columns, row 1 maps to the first 80 character storage positions in the character buffer, row 2 maps to the second 80 character storage positions, and so on. The sequence is the same regardless of the size of the display. Figure 1-1 shows the character positions for a 12-row, 80-column screen on the display and in the buffer.

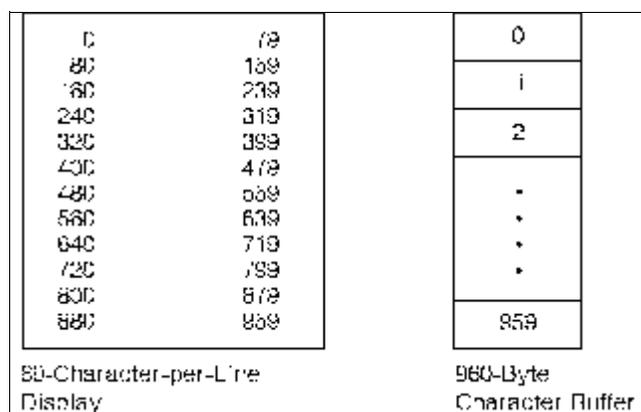


Figure 1-1. Mapping the Display to the Character Buffer

The character buffer can contain codes for graphic characters or field attributes. *Field attributes* define the start of a field and control the characteristics of the field. Because each storage location in the character buffer is mapped to a position on the display screen, the field attribute takes up a character position on the display screen and appears as a blank. The *field* is defined as the field attribute position plus the character positions up to, but not including, the next field attribute in the character buffer.

If the application program uses field attributes to define fields on the screen, then the screen is *formatted*. If there are no fields defined on the screen, then the screen is *unformatted*, and the operator uses it in free-form manner.

Figure 1-2 shows an example of the versatility of formatted fields. The solid characters represent the displayed form of characters stored in the character buffer. The squares represent buffer locations occupied by the field attribute characters (which are displayed as blanks). The dotted characters represent a field of data stored in the buffer but defined by the program as nondisplayable (not visible to the operator).

```
NAME : JOHN B DOE
SALARY 12523
JOB TITLE : WRITER
PHONE #: 383-7628
```

Figure 1-2. Example of Formatted Fields

---

## 1.4 Attributes

The display uses three kinds of attributes: field attributes, extended field attributes, and character attributes. Field and extended field attributes define the start of a field and control the characteristics of the field. Character attributes control the characteristics of a character.

The field attribute occupies a character location in the character buffer and is displayed as a blank. The extended field attributes and character attributes do *not* occupy positions in the character buffer but control the characteristics of the field and characters, respectively. The extended field attributes are extensions of the field attribute, as shown in [Figure 1-3](#).

```
PICTURE 3
```

Figure 1-3. Field, Extended Field, and Character Attributes

Subtopics:

- [1.4.1 Field Attributes](#)
- [1.4.2 Extended Field Attributes](#)
- [1.4.3 Character Attributes](#)

---

### 1.4.1 Field Attributes

The field attribute defines the start of a field and the characteristics of the field.

A field can wrap from the end of one row to the beginning of the next row on the screen. A field can also wrap from the last location on the screen to the first location. In any case, the field is terminated by the next field attribute. There is no limit to the number of fields that can be defined, other than that imposed by the screen size.

Field attribute defines the following field characteristics:

- *Protected or unprotected.* A protected field cannot be modified by the

operator. The operator can enter data or modify the contents of an unprotected field. Unprotected fields are classified as input fields.

- *Alphanumeric or numeric.* Unprotected alphanumeric fields are fields into which an operator enters data normally, using the shift keys (uppercase/lowercase or numeric/alphabetic) as required.

Fields defined as numeric accept all uppercase symbols and numerics from a data entry keyboard. On a typewriter keyboard, numeric has no meaning and all entries are accepted.

- *Autoskip.* The cursor skips over fields that are defined as protected and numeric.
- *Nondisplay or display/intensified display.* The selected characteristics apply to the entire field. *Nondisplay* means that any characters entered from the keyboard are placed in the buffer for possible subsequent transmission to the application program, but they are not displayed. *Intensified display* means the intensified characters appear on the screen brighter than the nonintensified characters. Some devices cannot intensify characters on the screen and highlight characters in a different manner.
- *Detectable or nondetectable.* A field defined as detectable can be detected by the selector pen or the cursor select key, subject to the use of a designator character.

---

## 1.4.2 Extended Field Attributes

The extended field attribute defines field characteristics such as color, character set, field validation, field outlining, and extended highlighting. The extended field attribute is always associated with a field attribute.

---

## 1.4.3 Character Attributes

A character attribute is associated with an individual character to define characteristics such as character color, character highlighting, or character set. The extended field attributes of any single character are superseded by the character attributes associated with it. However, characters in nondisplay fields are never displayed. The attribute structure used for character attributes is the same as for extended field attributes.

The attribute structure used for extended field attributes defines all characteristics with attribute type-value pairs, as shown in Table 1-1. Each attribute type is associated with a set of attribute values.

Attribute Pair -	Attribute Type	Attribute Value
------------------	----------------	-----------------

Attribute Type	Attribute Value
Extended highlighting	<p>Default--If used as a character attribute, the default is the set of characteristics defined by the extended field attribute. If used as an extended field attribute, the default becomes those characteristics indicated by the Query Reply (Highlight) structured field.</p> <p>Underscore--Each character or field is underlined.</p> <p>Blink--Each character or field affected flashes on and off.</p> <p>Reverse Video--In each character cell affected, the on/off illumination state for every display point is reversed. The effect is analogous to white on black becoming black on white.</p>
Color	<p>Default--If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, it is indicated by the Query Reply (Color) structured field.</p> <p>Multicolor--Indicates that the color is defined by a triple-plane Programmed Symbol (PS) set.</p> <p>All others--Assigned to the color identifications as indicated by the Query Reply (Color) structured field.</p>
Character set	<p>Default--If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, it is the nonloadable character set that has the Logical Channel Identifier (LCID) of X'00' in the Query Reply (character sets) structured field.</p> <p>Local Character Set ID--For the loadable or nonloadable character set.</p>
Field validation	<p>Mandatory Entry--A field that must be modified by the operator before the operator can transmit any data from the display.</p> <p>Mandatory Fill--A field that, if modified by the operator, must be filled with characters other than the null character before the operator can move the cursor out of the field or transmit any data from the display.</p> <p>Trigger--A field that, if modified by the operator, is transmitted inbound as soon as the operator tries to move the cursor out of the field. This allows the application program to receive and to validate fields one by one.</p>
Field outlining	<p>Default--No fields outlined.</p> <p>Outlining--Sixteen kinds of outlining can be defined by the combinations of the four horizontal and vertical lines.</p>
Transparency	<p>Default--If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, the default is determined by whatever the device supports on its inbound response to a Query Reply.</p>

Transparent--Picture elements (pels) of character background are ignored. Underlying presentation space can be viewed.

Opaque--Pels are set as indicated. Underlying presentation space cannot be viewed.

---

## 1.5 Commands

Commands define the function to be performed by the display. They control such things as whether the application program writes to or reads from a display and whether the screen is erased before new data is written.

In the 3270 data stream, some write commands are followed by the write control character (WCC). The WCC is used to perform such functions as sounding the audible alarm, starting the print operation, and enabling the keyboard.

The commands and the data that can follow the command are:

<b>Command</b>	<b>Data</b>
Write	A WCC, orders and data
Erase/Write	A WCC, orders and data
Erase/Write Alternate	A WCC, orders and data
Erase All Unprotected	None
Read Modified	None
Read Modified All	None
Read Buffer	None
Write Structured Field	Structured fields.

---

## 1.6 Orders

Orders are instructions in the 3270 data stream that provide control information. The orders that can be sent with the write commands are:

- Set Buffer Address (SBA)
- Start Field (SF)
- Start Field Extended (SFE)
- Modify Field (MF)
- Set Attribute (SA)
- Insert Cursor (IC)
- Program Tab (PT)
- Repeat to Address (RA)
- Erase Unprotected to Address (EUA)
- Graphic Escape (GE).

The orders that can be included in the inbound data stream are:

- Set Buffer Address
  - Start Field
  - Start Field Extended
  - Set Attribute
  - Graphic Escape.
-

## 1.7 Structured Fields

Structured fields are used to send and receive additional control functions and data to or from the terminal. There are three types of structured fields: outbound, outbound/inbound, and inbound. Outbound structured fields are introduced with the Write Structured Field (WSF) command. Inbound structured fields are preceded by an Attention Identifier (AID) X'88'.

The outbound and outbound/inbound structured fields are defined in Chapter 5, "Outbound/Inbound and Outbound Structured Fields." The inbound structured field functions that can be sent by the display are defined in Chapter 6, "Inbound Structured Fields."

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 2.0 Chapter 2. Partitions

Subtopics:

- 2.1 Introduction
- 2.2 Presentation Space, Partitions, Windows, and Viewports
- 2.3 Implicit and Explicit Partitions
- 2.4 Using Multiple Partitions
- 2.5 Activate Partition
- 2.6 Managing the Presentation Space
- 2.7 Partition Wait Condition (PWAIT)
- 2.8 System Lock Condition
- 2.9 Terminal Wait (TWAIT) Condition

---

## 2.1 Introduction

This chapter describes how the data stream allows the application program to divide the display surface into several areas, one of which is an active area. Each area is associated with a *partition*, and the partition that is active is the one that the operator is using to enter data or requests.

If the display supports partitioning, the application program can define a logical area called a *partition* that can differ in both size and shape from the physical display screen. The partition is defined with the Create Partition structured field. (Structured fields are discussed in Chapter 5, "Outbound/Inbound and Outbound Structured Fields" and Chapter 6, "Inbound Structured Fields.") Once a partition has been created, data is transmitted to and from the partition as if it were a physical screen with the geometrical characteristics specified in the Create Partition structured field.

Multiple partitions can be defined for the display that divide the display screen into several rectangular areas called *viewports*, where data from multiple partitions can be displayed on the same physical display screen. The operator can enter, delete, or modify data in any selected partition (except in protected fields) by positioning the cursor appropriately within the partition's viewport. (See Figure 2-1 in topic 2.2.)

The application program can organize a partition as either formatted or unformatted.

Each partition has a unique partition identifier (PID) assigned at creation time. The PID identifies the partition so that the application program can send data to, or receive data from, individual partitions.

If the application program does not define any partitions, the device assumes a single partition of default size with the PID equal to 0. This is referred to as an *implicit partition*.

---

## 2.2 Presentation Space, Partitions, Windows, and Viewports

The mapping of the presentation space to the physical screen is transparent to the application program.

The application program defines the amount of storage in the character buffer that is used by the partition. The application program and the operator can interact with the data in this buffer. The character buffer provides storage for characters to be shown on the display screen. This character buffer is simply addressable storage that contains as many locations as there are character positions in the partition. Each buffer location contains one character and is separately addressable.

PICTURE 4

Figure 2-1. Presentation Space and Viewport (without Scrolling)

Conceptually, however, you could think of a partition as a two-dimensional presentation space, the size of which is defined in terms of its depth H (number of rows) and its width W (number of columns). Thus, the character buffer associated with this partition is defined to be  $W \times H$  bytes. The addresses of the character buffer locations range from 0 to  $(W \times H) - 1$ .

Each partition is associated with a conceptual two-dimensional surface, called the *presentation space*. You could think of data as being presented on this two-dimensional surface, although the surface does not exist physically as such on the device. (See Figure 2-2 in topic 2.2.) A *window* on the presentation space identifies that part of the presentation space available for viewing on the display surface.

PICTURE 5

Figure 2-2. The Presentation Plane--A Conceptual View

The viewport is that area on the display surface where the operator sees the area of the presentation space bounded by the window. Each viewport is related to a window so that the area of the presentation space within the window appears on the screen within the viewport. The combination of the viewport and associated presentation space is a partition. (See Figure 2-2.)

For processing the 3270 data stream, a coordinate system must be defined on the presentation space. Rather than formatting data on the presentation space in row/column coordinates, 3270 compatibility requires lineal addressing of the presentation space (the character buffer) with the 3270 Set Buffer Address order. Orders are discussed in Chapter 4, "Character Sets, Orders, and Attributes."

Subtopics:

- 2.2.1 Relationship Between Presentation Space and Viewport
- 2.2.2 Scrolling
- 2.2.3 The Cursor in Partitions

---

## 2.2.1 Relationship Between Presentation Space and Viewport

When the display function scrolling is not implemented, a partition's presentation space and viewport have the same dimensions. (See Figure 2-1 in topic 2.2.) Assuming it is not in a nondisplay field, each data character in the presentation space appears in the corresponding row and column of the viewport.

When the presentation space is larger than the viewport, the viewport displays the data from the window. (See Figure 2-3 in topic 2.2.1.) The window and the viewport have the same dimensions (rows and columns). Assuming it is not in a nondisplay field, each data character in the window appears in the corresponding row and column of the viewport. The position of the window in relation to the data on the presentation space can be altered by *scrolling*.



Figure 2-3. Presentation Space, Window, and Viewport (with Scrolling)

---

## 2.2.2 Scrolling

In a scrollable partition, the presentation space data can be viewed by:

- Using the keys for scrolling
- Using the application program.

The application program positions the window by specifying the number of rows (see "Set Window Origin" in Chapter 5) by which the top of the window is to be offset from the top of the presentation space.

---

## 2.2.3 The Cursor in Partitions

As on the nonpartitioned screen, the cursor can be moved by operator keystrokes or by instructions from the application program, but the range of cursor movement on the screen is always constrained by the bounds of a

viewport. The cursor can be moved out of a given viewport only into another viewport; this can be done, for example, by using the Jump Partition key or by an Activate Partition structured field sent from the application program. The cursor can never appear at a screen position that is outside the bounds of a viewport. The partition associated with the viewport that contains the cursor is the *active partition*. The operator can enter data only in the active partition.

Each partition has a current cursor position that determines where alphanumeric data is placed in the presentation space during operator keystroking. The cursor on the display screen is displayed at the current cursor position of the active partition's presentation space. Data entry or cursor movement causes the current cursor position of the partition to be changed.

---

## 2.3 Implicit and Explicit Partitions

Displays using the 3270 data stream can operate in one of two states. When operating without partitions (operating with implicit partition 0), the device is said to be in implicit partition state. When partitions have been created explicitly, with the Create Partition structured field, the device is said to be in the explicit partitioned state. In each state, all the orders and commands described in this book are valid. The distinction between the two states relates to the way in which the usable area is managed. In implicit partition state, the size of the viewport equals the screen size. In explicit partitioned state, there can be more than one viewport on the usable area. For more information, see the Create Partition structured field in Chapter 5, "Outbound/Inbound and Outbound Structured Fields."

The 3270 data stream supports displays in both explicit partitioned and implicit partition states. The initial state of a display supporting partitions is the implicit partition state (for example, power-on reset). Interaction with implicit partition 0 on a display that supports partitioning is the same as on a display that does not support partitioning.

The application program can replace the implicitly created partition by *explicitly* creating one or more partitions of its own, thereby placing the display in explicit partitioned state.

The display can be returned to implicit partition state from explicit partitioned state by sending an Erase/Write or Erase/Write Alternate command with bit 1 of the WCC set equal to 1.

The transmission of data from a partition can be initiated either by the application program or by the display operator in the same way as for a nonpartitioned display surface. The display operator, however, can initiate a transmission only from an active partition. The application program can initiate a transmission from any partition by using the Read Partition structured field.

The Read Partition structured field provides the same read functions for partitions as the read commands do for nonpartitioned screens (for example, Read Modified and Read Buffer).

Subtopics:

- 2.3.1 The Implicit Partition
- 2.3.2 Explicit Partitions

---

## 2.3.1 The Implicit Partition

The display is placed in implicit partition state at BIND time (SNA) or when the display is powered on (non-SNA). A single implicitly defined partition is created automatically and assigned a PID of 0 with the default screen size. In implicit partition state, the size of the implicit partition is controlled by the Erase/Write (EW) and Erase/Write Alternate (EWA) commands in the data stream. It is also controlled by the Erase/Reset structured field.

EW redefines implicit partition 0 with the default screen size. EWA redefines implicit partition 0 with the alternate screen size. The Erase/Reset structured field allows you to specify either the default screen size or the alternate size. The default and the alternate size are specified in BIND SESSION.

The characteristics of the implicit partition are as follows:

- Partition parameters expressed in row/column coordinate system
- Partition size = screen size
- Window size = partition size
- Viewport size = window size
- Viewport origin = screen origin
- No scrolling permitted
- Unprotected (operator interaction allowed).

**Note:** For non-SNA environments, the default size is 1920 characters (24 x 80) on the display screen. The alternate size is implementation defined.

---

## 2.3.2 Explicit Partitions

The Create Partition structured field is used to replace the implicitly created partition 0 with a partition that is explicitly defined. The first Create Partition causes the implicit partition to be destroyed and the display to be placed in an explicit partitioned state.

The difference between the implicit partition, hereafter called *implicit partition 0*, and explicit partition 0 is that implicit partition 0 is assigned partition characteristics by default, whereas the application program can specify partition characteristics for explicit partition 0 using the Create Partition structured field. In addition, implicit partition 0 operates in implicit partition state, while explicit partition 0 operates in explicit partitioned state.

No matter which partition 0 is used, commands can be sent to the partition in the first byte of the data stream or enclosed in a structured field. When data is returned from either implicit or explicit partition 0, it is transmitted without use of the Inbound 3270DS structured field.

---

## 2.4 Using Multiple Partitions

The physical screen can be divided into several viewports, allowing data from multiple partitions to be displayed on the same physical screen. The partition with which the operator interacts is called the *active* partition, and only one partition can be active at a time. Each partition is identified by a PID.

The operator can move the screen cursor from one partition to the next by pressing the designated Jump Partition key. Pressing this key moves the cursor to the current cursor position of the next partition and makes that partition the active partition. If the partition ID has the value N, the next active partition is that partition with the smallest PID greater than N. If no such partition exists, the next active partition is that partition with the lowest PID.

---

## 2.5 Activate Partition

The application program can activate any partition by using the Activate Partition structured field and can deactivate any partition by activating another partition. The host application program can destroy any partition by using the Destroy Partition structured field. These structured fields are described in Chapter 5, "Outbound/Inbound and Outbound Structured Fields."

---

## 2.6 Managing the Presentation Space

The presentation space controls the value at which buffer addresses wrap. For those devices that permit scrolling, the presentation space can be larger than the viewport. For devices that do not permit scrolling, the presentation space is the same size as the viewport.

Figure 2-4 in topic 2.6 summarizes the management of presentation spaces. The commands, reset function, and structured fields shown in this figure are described in detail in this book.

When in implicit partition state, EW establishes a default screen size and EWA establishes an alternate screen size. The Erase/Reset structured field resets the device to implicit partition state, destroying all existing implicit or explicit partitions. It creates an implicit partition 0 with default partition characteristics and a default size if IPZ=B'0' or an alternate size if IPZ=B'1'.

When the Clear key is pressed, the device may either be set to the default size or remain the same, depending on the customization value selected or the option that has been designed into the product.

The functions of the EW, EWA, Write, and EAU commands on a partition with a nonzero PID are achieved by transmitting a WSF command containing an Outbound 3270DS structured field. This structured field contains the corresponding partition command; that is, it indicates whether the operation to be performed is to be an EW, EWA, Write, or EAU operation.

**Notes:**

1. Any local action that resets the screen, such as device return from local-test or SSCP use, may also reset the screen size to the default value.
2. If a structured field cannot be processed because of a data stream error, no state transition occurs.

Figure 2-4. Management of Presentation Spaces

---

## 2.7 Partition Wait Condition (PWAIT)

The partition wait condition (PWAIT) is a partition-related input-inhibit condition that, when activated, prevents operator keystroking into that partition. PWAIT is set by any operator enter action except a trigger action. The following rules apply:

- The host resets PWAIT by acknowledging the INOP.
- While the PWAIT condition is active for the active partition, the appropriate indicator is displayed. The operator can use the Jump Partition key to jump to another partition and keystroke into that partition. The PWAIT indicator is removed from the screen, but the PWAIT condition is remembered, and when the original partition becomes active again the PWAIT indicator reappears on the screen.
- The PWAIT condition can only be applied against INPID.

---

## 2.8 System Lock Condition

System Lock is a partition-related input-inhibit condition that, when activated, prevents operator keystroking into that partition. The following rules apply:

- The host or the operator can reset System Lock.
- While System Lock is active for the active partition, an appropriate indicator is displayed (provided a higher-priority condition, such as PWAIT, does not exist). The operator can use the Jump Partition key to jump to another partition and keystroke into that partition. The System Lock indicator is removed from the screen, but the System Lock condition is remembered, and when the original partition becomes active again the System Lock indicator reappears on the screen.
- At any one time, several partitions can have a system lock condition.
- Any operator enter action, except a trigger action, activates System Lock with PWAIT. If the display is in Contention state, Bid or begin bracket (BB) activates System Lock on partition 0 if partition 0 is

active.

- System Lock is removed by any of the following:
  - A write with keyboard restore removes System Lock from the partition addressed by the write.
  - A Reset key pressed by the operator removes System Lock from the active partition.
  - An end bracket (EB) indicator in the RU chain removes System Lock from the inbound partition whose PID is defined by INPID.

---

## 2.9 Terminal Wait (TWAIT) Condition

TWAIT is a terminal-related input-inhibit condition that prevents the operator from keystroking. TWAIT is activated when there is only one partition and the operator performs any enter action. Then TWAIT is a special case of the PWAIT condition, and the rules defined for PWAIT apply to TWAIT. If the display is in Contention state, Bid or BB will activate TWAIT. TWAIT is removed by a write with keyboard restore to any partition or by any transmission that after processing leaves the display in Send or Contention state.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 3.0 Chapter 3. 3270 Data Stream Commands

Subtopics:

- 3.1 Introduction
- 3.2 Commands within Structured Fields
- 3.3 Command Codes
- 3.4 Write Control Character (WCC) Byte
- 3.5 Write Operation
- 3.6 Read Operations
- 3.7 Read Operations from Partitions
- 3.8 Read Operations for SNA
- 3.9 Inbound Operation (INOP)
- 3.10 Inbound Partition Identifier (INPID)
- 3.11 Enter Actions
- 3.12 Processing of Read Commands (Alphanumeric)

---

### 3.1 Introduction

This chapter describes the commands and how they function in the 3270 data stream. The 3270 data stream commands and user-provided data are transmitted between the application program and the display.

The outbound data stream usually consists of write commands and a write control character (WCC) followed by orders and data. If the write command is a Write Structured Field (WSF), however, no WCC byte follows this write command in the data stream. The format of the write type command is as follows:

Write Command	WCC	Orders and Data
---------------	-----	-----------------

or

WSF	Structured Field(s)
-----	---------------------

Commands are sent to a display to initiate the total or partial writing, reading, or erasing of data in a selected character buffer. Commands are

sent as a command code in the first byte of a request/response unit (RU) chain (referred to throughout this book as RU chain or chain), or they may be sent in structured fields.

---

## 3.2 Commands within Structured Fields

Structured fields are used to extend the function provided by the commands. When structured fields are used, the entire RU chain must be made up of structured fields. Therefore, certain structured fields have been defined to allow sending command functions, orders, attributes, and so on, in the same RU chain with other types of structured fields. (See Chapter 5, "Outbound/Inbound and Outbound Structured Fields," and Chapter 6, "Inbound Structured Fields," for a description of the structured fields.)

The Outbound 3270DS structured field provides the write command functions (Write, Erase/Write, Erase/Write Alternate, Erase All Unprotected), and the Read Partition structured field provides the read command functions (Read Buffer, Read Modified, Read Modified All).

In general, the command protocol is the same whether the function is initialized by a command code (first byte of the 3270 data stream) or by an Outbound 3270DS, Write Structured Field, or Read Partition structured field. There are some differences, however, which are detailed in this chapter.

---

## 3.3 Command Codes

The command codes are not unique code points; they rely on position to resolve ambiguity. Only one command is allowed per RU chain. The command must be the first byte of the 3270 data stream.

Table 3-1 contains codes and command abbreviations.

Command	Abbreviation	HEBCDIC	ASCII
Write	W	X'F1'	X'31'
Erase/Write	EW	X'F5'	X'35'
Erase/Write Alternate	EWA	X'7E'	X'3D'
Read Buffer	RB	X'F2'	X'32'
Read Modified	RM	X'F6'	X'36'
Read Modified All	RMA	X'6E'	X'3E'
Erase All Unprotected	EAU	X'6F'	X'3F'
Write Structured	WSF	X'F3'	(Note)

Field			

**Note:** The use of structured fields requires that the full 8 bits of a byte be used; therefore, WSF is not supported in an American National Standard Code for Information Interchange (ASCII) environment.

### 3.4 Write Control Character (WCC) Byte

The WCC byte is not a unique code but is identified by position; that is, it is the byte following the write type command. If the WCC is omitted, whatever follows the write type command is interpreted as the WCC. The data stream is normally a minimum of a Write, EW, or EWA command and the WCC. If any write command (except EAU) is sent with no WCC or data, it is treated as a no operation (NOP).

Although no WCC follows the WSF command, there may be a WCC in the Outbound 3270DS structured field. When the WCC specifies an operation that does not apply to the display, the specified operation is not performed. For example, the Sound Alarm is an NOP if the display does not have an Audible-Alarm feature.

All WCC functions except for Reset MDT are deferred until data is written and orders are performed. See Table 3-2 in topic 3.4 for a description of each WCC bit and Table 3-3 in topic 3.4 for a summary of the reset actions.

When a data stream contains multiple WCCs (because they may appear in structured fields), the WCC functions are performed as follows:

Reset	Performed in each structured field as it is encountered.
Start print	Performed at the end of the RU chain, after the write operation has been completed. Only the last structured field can have a WCC that specified Start Print; otherwise, the chain is rejected (sense code X'1001' or sense code X'1005').
Sound alarm	Performed for each structured field, at the end of the operation specified for the structured field.
Keyboard restore	Performed for each structured field, at the end of the operation specified for the structured field.
Reset Modified Data Tag (MDT)	Performed for each structured field, prior to the writing of any data or the performing of any orders in the data stream.

Table 3-2. Write Control Character (WCC) Bit Definitions for Displays	
<b>Bit</b>	<b>Explanation</b>
0	If the Reset function is not supported, the only function of

	<p>bits 0 and 1 is to make the WCC byte an EBCDIC/ASCII-translatable character. Bits 0 and 1 are set in accordance with <u>Figure C-1 in topic C.0</u>.</p> <p>If the Reset function is supported, bit 1 controls reset/no reset and bit 0 has no function. When bit 1 is used for the Reset function the WCC byte is no longer always EBCDIC/ASCII-translatable; therefore, the Reset function cannot be supported in an ASCII environment.</p>
1	WCC reset bit. When set to 1, it resets partition characteristics to their system-defined defaults. When set to 0, the current characteristics remain unchanged (no reset operations are performed).
2 and 3	For printers. (See <u>Chapter 8, "Printer Operations."</u> )
4	Start-printer bit. When set to 1, it initiates a local copy operation of the display surface at the completion of the write operation. When no printer is available, a negative response (0801) is returned. (See <u>Chapter 8, "Printer Operations"</u> for details of local copy operation.)
5	Sound-alarm bit. When set to 1, it sounds the audible alarm at the end of the operation if that device has an audible alarm.
6	Keyboard restore bit. When set to 1, this bit unlocks the keyboard. It also resets the AID byte.
7	Bit 7 resets MDT bits in the field attributes. When set to 1, all MDT bits in the device's existing character buffer are reset before any data is written or orders are performed.

Table 3-3. Write Control Character (WCC) Reset Actions for Displays

Reset Condition	Partitions Not Supported	Partitions Supported, but Display in Implicit Partition State	Partitions Supported, but Display in Explicit Partitioned State
1. WCC following an Erase/Write or an Erase/Write Alternate command.			
a. WCC = Reset.	Perform the command; reset the inbound reply mode to field (if applicable).	Perform the command; reset the inbound reply mode to field.	Reset the display to the implicit partition state; perform the command.
b. WCC = No reset.	Perform the command.	Perform the command.	Perform the command against explicit partition 0; if

			not explicit partition 0, reject the command.
2. WCC following a Write command.			
WCC = Reset or no reset.	Perform the command.	Perform the command.	Perform the command against explicit partition 0; if not explicit partition 0, reject the command.
3. WCC in Outbound 3270DS, and the function is Erase/Write or Erase/Write Alternate.			
a. WCC = Reset.	If the PID equals 0, perform the function and reset the inbound reply mode (if applicable).	If the PID equals 0, perform the function and reset the inbound reply mode.	Reset the designated (PID) partition, and perform the function against the designated partition.
	If the PID does not equal 0, reject with -RSP.	If the PID does not equal 0, reject with -RSP.	If the designated partition does not exist, reject with -RSP.
b. WCC = No reset.	If the PID equals 0, perform the function.	If the PID equals 0, perform the function.	Perform the function against the designated partition.

	If the PID does not equal 0, reject with -RSP.	If the PID does not equal 0, reject with -RSP.	If the designated partition does not exist, reject with -RSP.
4. WCC in Outbound 3270DS, and the function is Write.			
WCC = Reset or no reset.	Perform the function if the PID equals 0; otherwise, reject with -RSP.	Perform the function if the PID equals 0; otherwise, reject with -RSP.	Perform the function against the designated partition. If the designated partition does not exist, reject with -RSP.

## 3.5 Write Operation

The process of sending a write type command and performing that command is called a *write operation*. Five write commands are initiated by the application program and performed by the display:

- Write (W)
- Erase/Write (EW)
- Erase/Write Alternate (EWA)
- Erase All Unprotected (EAU)
- Write Structured Field (WSF).

The Write, EW, and EWA commands are used by the application program to load, format, and selectively erase a character buffer or presentation space at the display. These commands can also initiate certain display operations, such as copying the contents of the display screen, restoring the keyboard, and sounding the audible alarm.

Write and EW operations are identical except that EW causes complete erasure of the character buffer before the write operation is started. Thus, EW is used to load the buffer with completely new data, whereas Write can be used to add to or modify existing buffer data. EWA and EW are identical except that EW sets and uses the default display screen size while EWA sets and uses the alternate display screen size.

Write, EW, and EWA, when sent in the first byte of the data stream, are used for write operations in partition 0. They must be encoded in a structured field if used for partitions with nonzero IDs.

The WSF must be used for any write operation to partitions with nonzero IDs. The command is followed by one or more structured fields, which are interpreted and performed by the display. The structured field identifies the specific partition by its partition identifier.

Subtopics:

- [3.5.1 Write Command](#)
- [3.5.2 Erase/Write Command](#)
- [3.5.3 Erase/Write Alternate Command](#)
- [3.5.4 Write Structured Field Command](#)
- [3.5.5 Erase All Unprotected Command](#)
- [3.5.6 Attention Identification \(AID\)](#)

---

## 3.5.1 Write Command

The Write command writes data into specified locations of the character buffer of partition 0 without erasing or modifying data in the other locations. Data is stored in successive buffer locations until an order is encountered in the data stream that alters the buffer address, or until all the data has been stored. During the write operation, the buffer address is advanced one location as each character is stored.

The Write command is required for Systems Application Architecture (SAA) support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

The buffer location where the entry of data starts depends on the starting location specified by the SBA order that follows the WCC. If an SBA does not follow the WCC, the starting location is the buffer address where the cursor is positioned. The formatting and placement of write data and the modifying of existing buffer data are described under "Orders" in Chapter 4.

---

## 3.5.2 Erase/Write Command

The EW command does the following:

- Sets the implicit partition size to the default size, if in implicit state.
- Resets a Program Check Indication, if one exists.
- Erases the character buffer by writing null characters into all buffer locations.
- Sets all the associated character attributes and extended field attributes to their default value (X'00').
- Erases all field validation attributes.

- Sets the current cursor position to 0. If directed to a partition, autoscroll is performed, if necessary, to position the window at offset (0,0).
- If bit 1 of the WCC is set to B'1', EW does the following:
  - Resets the inbound reply mode to Field.
  - Resets to implicit partition state, if currently in explicit partitioned state. It destroys all partitions, creates implicit partition 0 with default screen size, and sets inbound PID to 0 and INOP to Read Modified.
- Provides an acknowledgment of any outstanding read or enter if the keyboard restore bit in the WCC is set to B'1'.
- Provides a negative trigger reply.
- Performs a write operation.

The EW command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

Bit 1 in the WCC carries reset information used by EW. If no WCC is defined following an EW, the command is considered a NOP. Therefore, no erasing or resetting occurs and any outstanding read or enter operation is not acknowledged. However, it is treated as a negative reply to a trigger field AID.

To perform the EW function in a named partition, other than partition 0, a WSF command must be used. An Outbound 3270DS structured field contains the PID and the partition command (EW).

---

### 3.5.3 Erase/Write Alternate Command

The EWA command performs the same operation as described for EW, but EWA uses the alternate implicit partition size. If there is no alternate screen size, the EWA is treated the same as EW.

EWA is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

---

### 3.5.4 Write Structured Field Command

WSF is used to send structured fields from the application program to the display. On the application-to-display flow (outbound), structured fields can be sent only with the WSF command.

Devices not supporting structured field data streams must reject this

command (sense code X'1003').

The format of a WSF data stream is as follows:

WSF Command	Structured Field	Structured Field ...
-------------	------------------	----------------------

The data stream can contain one or more structured fields. Each structured field contains a length count that enables the receiver to calculate where the current structured field ends and the next one begins. Some structured fields may have a length field that equals 0, but only when they are sent as the last structured field in the RU chain.

The Outbound 3270DS is an example of a structured field. It allows one of four operations to be performed within the named partition:

- Write (W)
- Erase/Write (EW)
- Erase/Write Alternate (EWA)
- Erase All Unprotected (EAU).

W can either be sent as the first byte of the data stream to write to partition 0 or be enclosed in the Outbound 3270DS structured field to write to any explicitly created partition. See "Create Partition" in Chapter 5 for information on creating a partition.

---

### 3.5.5 Erase All Unprotected Command

EAU does the following:

- Clears all the unprotected character locations of the partition to nulls and sets any character attributes affected to their default values
- Resets to 0 the MDT bit in the field attribute for each unprotected field
- Unlocks the keyboard
- Resets the AID
- Repositions the cursor to the first character location, after the field attribute, in the first unprotected field of the partition's character buffer.

EAU is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

If the entire buffer is protected, buffer data is not cleared and MDT bits are not reset. However, the keyboard is unlocked, the AID is reset, and

the cursor is repositioned to the first buffer address in the partition.

---

## 3.5.6 Attention Identification (AID)

The AID byte appears only in the inbound (display to application program) data stream and must be the first byte of the inbound data stream. The AID indicates the source or type of data that follows in the data stream. Usually, there is only one AID byte in an RU chain. The exception is an RU chain containing an Inbound 3270DS structured field that itself can contain an AID byte.

When the operator initiates an enter operation, the display includes an AID byte in the input to the application program indicating the operator action. Operator actions that initiate an enter operation include the following:

- Pressing a program function or program attention key
- Pressing the Enter, Clear, or Clear Partition key
- Reading a magnetic stripe with a magnetic reader
- Detecting on an attention field with the selector pen.

The possible AID bytes are shown in Table 3-4 in topic 3.5.6. All AID bytes transmitted by the display are a result of operator actions except for the following:

- No AID generated
- No AID generated (printer)
- Structured field
- Read partition.

The display sends the no-AID-generated AIDs for unsolicited reads, errors, and unusual conditions. It sends the structured field AID whenever a structured field is sent inbound. The structured field can be sent as a result of some operator action, or it can be a reply to a previous application program request. The display sends the structured field AID (X'88') and the Read Partition AID when replying to a Read Partition structured field requesting a read operation. (Table 3-4 in topic 3.5.6 lists the AID bytes sent from the display.)

When data is transmitted to the application program, the most recent AID value is transmitted. The display records the most recent AID byte value. This value, initially set to *no AID generated*, can be set to another value by operator action. (See Table 3-4 in topic 3.5.6.) The application program can reset the AID value to *no AID generated* by sending a write command with the keyboard restore bit set on in the WCC.

Once the AID is set, it remains set and input is inhibited until one of the following commands is issued:

- Any write command that has the keyboard-restore bit on in the WCC
- EAU.

Table 3-4. Attention Identification (AID) Bytes Sent from the Display to the Application Program
--

AID	EBCDIC (Hex)	ASCII (Hex)	EBCDIC (Graphic)
No AID generated	60	2D	
No AID generated (printer only)	E8	59	Y
Structured field	88(1)		h
Read partition	61(2)		/
Trigger action	7F		"
Test Req and Sys Req	F0	30	0
PF1 key	F1	31	1
PF2 key	F2	32	2
PF3 key	F3	33	3
PF4 key	F4	34	4
PF5 key	F5	35	5
PF6 key	F6	36	6
PF7 key	F7	37	7
PF8 key	F8	38	8
PF9 key	F9	39	9
PF10 key	7A	3A	:
PF11 key	7B	23	#
PF12 key	7C	40	@
PF13 key	C1	41	A
PF14 key	C2	42	B
PF15 key	C3	43	C
PF16 key	C4	44	D
PF17 key	C5	45	E
PF18 key	C6	46	F
PF19 key	C7	47	G
PF20 key	C8	48	H
PF21 key	C9	49	I
PF22 key	4A	5B	ç
PF23 key	4B	2E	.
PF24 key	4C	3C	<
PA1 key	6C	25	%
PA2 key (Cncl)	6E	3E	>

PA3 key	6B	2C	,
Clear key	6D	5F	-
Clear Partition key	6A		
Enter key	7D	27	,
Selector pen attention	7E	3D	=
Magnetic Readers: Operator ID reader	E6 E7	57 58	W X
Mag Reader Number			

(1) When structured fields are sent inbound, the first byte of the inbound data stream is always X'88'. Some structured fields also contain an AID, so the inbound data stream that contains structured fields can have more than one AID present. For example, the Inbound 3270DS contains an AID of X'61' if it is sent inbound as a result of a Read Partition structured field. It contains an AID of X'7D' if sent inbound as a result of the Enter key.

(2) Can never be the first byte in an inbound data stream.

## 3.6 Read Operations

The process of sending data inbound is called a *read operation*. A read operation can be initiated by the following:

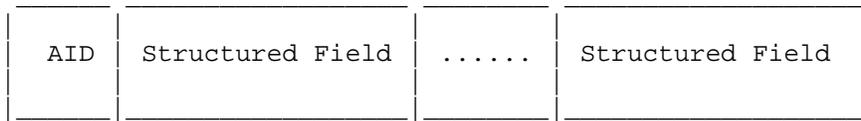
- The host application sending an explicit read command
- The host application program sending a Read Partition structured field specifying Read Buffer, Read Modified, or Read Modified All
- An operator action, for example, pressing the Enter key.

A read operation sends an inbound data stream (from the terminal to the application program) with an AID byte as the first byte of the inbound data stream. The inbound data stream usually consists of an AID followed by the cursor address (2 bytes). These 3 bytes of the read data stream, the AID, and cursor address are known as the *read heading*. The inbound data stream format is as follows:

AID	Cursor address (2 bytes)	Data
-----	-----------------------------	------

An inbound data stream can also consist of an AID (X'88') followed by

structured fields as follows:



Subtopics:

- 3.6.1 Read Commands
- 3.6.2 Read Modified Command
- 3.6.3 Read Commands in Structured Fields

---

## 3.6.1 Read Commands

Three read commands can be sent by the application program: Read Buffer, Read Modified, and Read Modified All.

When the display receives the Read Buffer command, the entire contents of the character buffer are sent to the application program. The Read Modified and Read Modified All commands start read operations that transfer only those character buffer fields that have the MDT bit set on in the field attribute.

The Read Partition structured field can initiate a read buffer, read modified, or read modified all operation, or it can initiate a query operation.

For both implicit and explicit partition 0, the read command can be the first byte of the data stream or it can be encoded in the Read Partition structured field.

The information that the display transfers to the application program in reply to Read Partition is either read data from one partition or a reply to a query. The read data is returned in the Inbound 3270DS structured field. The reply to the query is returned in one or more Query Reply structured fields.

The contents of an inbound data stream depend on the inbound reply mode. The inbound reply mode is set when the application program sends a Set Reply mode structured field. The inbound reply modes that can be set are Field, Extended Field, and Character; the default is Field mode. The three modes function as follows:

- Field mode supports inbound data that contains field attributes. SF and SBA orders, field attributes, characters, and Graphic Escape (GE) code X'08' can be included in the inbound transmission.
- Extended Field mode supports field attributes and extended field attributes. SFE and SBA orders, field attributes, extended field attribute, characters, and the GE code X'08' can be included in the inbound transmission.

- Character mode supports the field attributes, extended field attributes, characters, and character attributes. SFE, SBA, and SA orders, field attributes, extended field attributes, character attributes, and the GE code (X'08') can be included in the inbound transmission.

Subtopics:

- 3.6.1.1 Read Buffer Command
- 3.6.1.2 Read Buffer Field Mode
- 3.6.1.3 Read Buffer Extended Field Mode
- 3.6.1.4 Read Buffer Character Mode

### 3.6.1.1 Read Buffer Command

Operation of the Read Buffer command causes all data in the addressed display buffer, from the buffer location at which reading starts through the last buffer location, to be transmitted to the host. For displays, the transfer of data begins from buffer address 0.

The Read Buffer command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

**Note:** Because of the large quantities of data that are transferred and processed, using the Read Buffer command significantly increases the 3270 subsystem and teleprocessing network response times. The Read Buffer command is intended primarily for diagnostic purposes.

### 3.6.1.2 Read Buffer Field Mode

In read buffer field mode, no extended field or character attribute information is returned. Only SF orders occur in the data stream. The format of the Read Buffer Field data stream is as follows:

AID	Cursor Address (2 bytes)	SF Order	Attribute	Data
-----	-----------------------------	----------	-----------	------

### 3.6.1.3 Read Buffer Extended Field Mode

In read buffer extended field mode, SFE orders are generated in place of SF orders, and no character attributes are transmitted inbound. The form of the Read Buffer Extended Field data stream is as follows:

AID	Cursor Address	SFE Order	No. of Attribute Type Value Pairs	Attribute Type	Attribute Value	..... Data
-----	----------------	-----------	-----------------------------------	----------------	-----------------	------------

Attributes with default values are not transmitted inbound in the SFE order.

---

### 3.6.1.4 Read Buffer Character Mode

In read buffer character mode, the inbound data stream is as defined for extended field mode above, except that SA orders can be inserted. SA orders are generated only when the attribute value of an attribute type listed in the Set Reply Mode changes.

All attributes are assumed to have their default values at the beginning of the inbound transmission. Therefore, the first SA order generated is for the first attribute type not equal to its default value. However, SA orders are not generated for attribute types not listed in the Set Reply Mode structured field. The format of the Read Buffer Character data stream is as follows:

---

AID	Cursor Address	SA	Attr Type	Attr Value	Data	SFE	No. of Attr Type-Value Pairs	Attr Type	Attr Value	...	Data
-----	----------------	----	-----------	------------	------	-----	------------------------------	-----------	------------	-----	------

---

### 3.6.2 Read Modified Command

Read Modified initiates either a read modified or a short read operation, depending on the operator actions at the display station.

The Read Modified command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

Subtopics:

- 3.6.2.1 Read-Modified Operation
- 3.6.2.2 Short-Read Operation
- 3.6.2.3 Read Modified Field and Extended Field Modes
- 3.6.2.4 Read Modified Character Mode
- 3.6.2.5 Read Modified All Command
- 3.6.2.6 Read Modified All Modes

---

#### 3.6.2.1 Read-Modified Operation

During a read modified operation, if an AID other than selector pen attention, cursor select key, PA key, or Clear key is generated, all fields that have been modified by keyboard, selector pen, or magnetic-reader activity are transferred to the application program. A major feature of the read modified operation is null suppression. Only non-null character data and corresponding character attribute data (in Character mode) are transmitted. All null character data and all extended attributes for null character data are suppressed.

If a space or null selector pen AID is generated, fields are not transferred to main storage during the read modified operation. Instead, when a set MDT bit is found (indicating selector pen and/or keyboard activity), only the read heading, the SBA order code, and the attribute address +1 are transferred.

If fields are modified by the keyboard, but completion of the modification is signaled by a selector pen attention operation on other than ampersand character designator fields, a resulting read modified operation reads only the address of the modified fields, not the modified data. A Read Modified command can be used to obtain both the address of, and the data in, each field that has the MDT bit set to 1.

For SNA, the buffer location at which the search for attribute locations begins is location 0. For non-SNA, the buffer location is determined as follows:

- If the Read Modified command is unchained or is chained from a Copy, Select, Sense, or No Operation command, the search begins at buffer address 0.
- If the Read Modified command is chained from a Write, Erase/Write, Read Modified All, or Read Buffer command, the search begins at the current address.

The search for modified fields ends when the last buffer location is checked.

If the buffer is formatted (contains fields) but none of the fields have been modified, the read data stream consists of the 3-byte read heading only.

If the buffer is unformatted (contains no fields), the read data stream consists of the 3-byte read heading followed by all alphanumeric data in the buffer (nulls are suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no attribute bytes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of data cannot be determined. Data transfer starts at address 0 and continues to the end of the buffer. At the end of the operation, the buffer address is set to 0.

---

### **3.6.2.2 Short-Read Operation**

The Read Modified command causes a short read operation if the Clear, CNCL (Cancel), or a PA key has been pressed at the selected device. During the short-read operation, only an AID byte is transferred to the application program. This AID byte identifies the key that was pressed.

---

### **3.6.2.3 Read Modified Field and Extended Field Modes**

In read modified field and extended field modes, no character attribute information is returned. Only SBA orders occur in the data stream. The format of the Read Modified Field and Extended Field data streams is as follows:

AID	Cursor Address (2 bytes)	SBA	Attribute Address + 1	Text
-----	-----------------------------	-----	-----------------------	------

---

### 3.6.2.4 Read Modified Character Mode

This data stream for read modified character mode is identical with that defined for the read modified field mode above except that an SA order is inserted into the data stream as required to define the change in the attribute value for the graphic characters transmitted. SA orders are generated only when the attribute value of an attribute type listed in the Set Reply mode changes. All attributes are assumed to have their default values at the beginning of the inbound transmission. Therefore, the first SA generated is for the first attribute type not equal to its default. However, SA orders are not generated for attribute types not listed in the Set Reply Mode structured field. The form of the data stream is as follows:

AID	Cursor Address (2 bytes)	SA	Attr Type	Attr Value	Data	SBA	Attr Address + 1	SA	Attr Type	Attr Value
-----	-----------------------------	----	--------------	---------------	------	-----	------------------------	----	--------------	---------------

---

### 3.6.2.5 Read Modified All Command

The Read Modified All command operates exactly like the Read Modified command, except that it always causes both the addresses and the data from all modified fields to be sent to the application program independent of the AID code. This includes those AID codes that would normally result in a short read, as well as for trigger action (X'7F'), Clear Partition (X'6A'), and Read Partition (X'61').

The Read Modified All command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

---

### 3.6.2.6 Read Modified All Modes

The Read Modified All Field, Extended Field, and Character Mode commands operate exactly like the Read Modified Field, Extended Field, and Character Mode commands.

---

## 3.6.3 Read Commands in Structured Fields

For an application program to initiate a read operation from a partition with a nonzero PID, the Read Partition structured field must be used. The Read Partition structured field can also be used in addition to the Read Buffer, Read Modified, and Read Modified All commands to initiate a read from implicit or explicit partition 0. The Read Partition structured field allows the application program to specify a Read Buffer, Read Modified, or Read Modified All operation. The inbound reply to a Read

Partition structured field (directed to a nonzero partition) specifying a Read Buffer, Read Modified, or Read Modified All operation is sent in an Inbound 3270DS structured field.

---

## 3.7 Read Operations from Partitions

In addition to an operator enter action or an application program initiated (unsolicited) read, a read operation can result from an application program retry. During the processing of read and enter operations, the display goes into retry state from the time that the data is transferred until the application program acknowledges that the transfer of data was successful. (The retry state is discussed under "Read States" in topic 3.8.5.)

The addition of partitions to the 3270 data stream made it necessary to provide a formalized host application retry/acknowledgment mechanism for certain read operations. These read operations, called *inbound operations* (INOP), are defined as read operations caused by one of the following:

- An operator enter action (for example, the Enter key)
- A host application issuing a partition command (Read Partition structured field = Read Modified, Read Modified All, Read Buffer, Query, or Query List).

INOP here refers to the *current inbound operation* and INPID refers to the *partition associated with the current INOP*.

| INOP defines the type of operations to be performed on a retry.  
| See "Inbound Operation (INOP)" in topic 3.9.

| INPID is the inbound partition identifier and it defines the partition  
| from which data is to be transmitted. See "Inbound Partition Identifier  
| (INPID)" in topic 3.10. The INPID can be either implicit or explicit  
| partition 0.

In an SNA environment, an inbound operation causes the display to go into a retry state. Once in the retry state, another inbound operation cannot be initiated until the host application acknowledges the INOP. Once the acknowledgment is received, the display reverts to the *normal* state and ends the current inbound operation (INOP).

What constitutes an acknowledgment of an inbound operation depends on the inbound operation and the environment (SNA or non-SNA). The acknowledgment requirements are described in detail later in this section.

When the display is in a retry state, the host application can cause a repeat of the INOP by sending an RM command ( **not** a partition Read Modified command).

The Read Modified command is the only mechanism for a complete retry of the INOP. When in a retry state, a Read Modified All or Read Buffer command results in performing the Read Modified All or Read Buffer on the INPID. Although the use of INPID is repeated, the inbound operation performed is not necessarily the same as INOP.

Read operations that are not considered to be inbound operations are those operations initiated by a Read Modified, Read Modified All or Read Buffer command when the display is in the normal read state. The Read Modified, Read Modified All or Read Buffer is performed against either implicit or explicit partition 0 and does not cause a transition from the normal state. There is no retry/acknowledgment associated with these commands when the display is in the normal state. INOP and INPID are not relevant to this situation.

---

## 3.8 Read Operations for SNA

This section describes read operations for SNA. For non-SNA read operations, see Chapter 9, "Binary Synchronous Communications (BSC) Environment" and Chapter 10, "Non-SNA Environment (Locally Attached Devices--3272 Version)."

Subtopics:

- 3.8.1 Operator Enter Actions
- 3.8.2 Application Initiated Reads
- 3.8.3 Read Partition Structured Field
- 3.8.4 Host Retry
- 3.8.5 Read States

---

### 3.8.1 Operator Enter Actions

An operator enter action causes an AID and, optionally, data to be sent to the application program. For example, pressing a PA key or the Clear key causes only an AID to be sent, while the Enter key causes an AID and data from all fields with the MDT bit set to be sent.

An enter action does the following:

- Sets INOP and INPID so that the operation and partition identifier are known if a retry is needed.
  - Raises the enter-inhibit condition so that the operator cannot generate an AID from any partition. To ensure that the operator cannot change the value of INPID or INOP until the application program acknowledges that the transfer of data was successful, an input-inhibit condition, called *enter-inhibit*, prohibits the operator from generating an AID from any partition.
  - Raises the PWAIT and system lock conditions if the enter action is not a trigger action (AID = X'7F'). See "Trigger" in topic 4.4.9. To ensure data integrity for retry of operator enter actions, an input-inhibit condition, called PWAIT, prohibits the operator from working in the partition until the application program acknowledges that the transfer of data was successful.
  - Places the device in retry enter state.
-

## 3.8.2 Application Initiated Reads

An application-initiated read is one that is unsolicited by the device. It can be one of the following:

- A Read Partition structured field
- A Read Buffer, Read Modified, or Read Modified All command received for implicit or explicit partition 0.

---

## 3.8.3 Read Partition Structured Field

A Read Partition structured field does the following:

- Sets INPID if the operation is Read Buffer, Read Modified, or Read Modified All.
- Sets INOP to Read Modified, Read Modified All, Read Buffer, Query, or Query List.
- Raises the enter-inhibit condition
- Sends the data to the application program and places the device in retry-read state.

Explicit partition 0 can be read with a read command and does not require a Read Partition structured field operation. The Read Partition structured field identifies a partition through its PID. The partition is assigned a PID when it is created by the Create Partition structured field.

The display's reply (except from PID 0) is an inbound data stream containing one or more structured fields of the following form:

AID = X'88'	Structured Fields
-------------	-------------------

where the inbound control information is encoded into structured fields as defined in [Chapter 5, "Outbound/Inbound and Outbound Structured Fields"](#) and [Chapter 6, "Inbound Structured Fields."](#)

The display begins the session with no partitions explicitly defined or, thought of another way, begins with the entire display surface as one partition. This partition, called *implicit partition 0*, occurs in the following situations:

- After session activation but before the first Create Partition structured field

- After all partitions created by Create Partition are deleted
- After the Clear key is pressed
- When an EW command, EWA command or an Erase/Reset structured field is received as the first byte of an outbound data stream and the WCC Reset bit is on.

Since some application programs cannot send the Create Partition, those sessions use implicit partition 0 for the duration of their session.

**Note:** The Create Partition structured field can define explicit partition 0 by setting PID = X'00'. The difference between implicit partition 0 and explicit partition 0 is that implicit partition 0 is assigned partition characteristics by default, while the application program can specify partition characteristics for explicit partition 0 by using the Create Partition structured field.

No matter which partition 0 is used, commands can be sent to the partition as the first byte of the data stream or enclosed in a structured field.

In the event that an error causes the application program to retry the Read Partition, the display must remember the read operation that was performed and the partition on which the operation was performed.

| INOP records the type of operation (Read Buffer, Read Modified, Read Modified All, Query or Query List) that is being performed so that it can be repeated if a retry of a Read Partition is required.

INPID defines the partition from which the data is retransmitted.

The inbound data stream contains an Inbound 3270DS structured field with an AID byte of X'61' to identify that the data is a result of Read Partition:

X'88'	Length	X'80'	PID	X'61'	Cursor Address	Data
-------	--------	-------	-----	-------	----------------	------

**Notes:**

1. The X'80' identifies the structured field as an Inbound 3270DS.
2. From the AID X'61' to the end of the structured field, the data stream is the same as that sent by a corresponding Read Buffer, Read Modified, or Read Modified All command in implicit partition state.

A Read Buffer, Read Modified, or Read Modified All command received for implicit or explicit partition 0 does the following:

- Transmits data from partition 0
  - Leaves the device in its current read state.
-

## 3.8.4 Host Retry

When the device is in any of the retry states, the Read Modified command performs a retry of the inbound operation on INPID.

The Read Buffer or Read Modified All commands perform their respective read operations in the inbound partition. The Read Buffer and Read Modified All commands do not necessarily retry the inbound operation.

---

## 3.8.5 Read States

The device has one of two states with respect to read operations: normal read or retry:

- Normal read state. Prior to initiation of a read operation, the device is in normal read state. An operator enter action or a Read Partition structured field causes the device to transmit the data and to go into a retry state.
- Retry state. There are two forms of retry state:
  - Retry enter, in which the entered data has been transmitted to the application program
  - Retry read, in which the data read has been transmitted to the application program.

While in retry state, the last INOP can be retried by using a Read Modified command. The application program must acknowledge receipt of the read data, which causes the device to revert from a retry state to the normal read state. There is no retry state for any read command sent as the first byte of the data stream.

Subtopics:

- [3.8.5.1 Normal Read State](#)
- [3.8.5.2 Retry Enter State](#)
- [3.8.5.3 Retry Read State](#)
- [3.8.5.4 Read State Transitions](#)

---

### 3.8.5.1 Normal Read State

In normal read state, an operator enter action or a host initiated read is accepted and processed. [Table 3-5](#) shows the normal read state process.

Table 3-5. Normal Read State Process		
Transition Event	Next State	Operations

Enter action	Retry enter	Transmit read modified data stream.
Any read command(1)	Normal read(2)	Transmit data stream defined by command.
Read Partition(3)	Retry read	Transmit data stream defined by Read Partition.
<b>Notes:</b>		
(1) Partition 0, whether implicit or explicit, must exist or the read command sent as the first byte of the data stream is rejected.		
(2) There is no retry state.		
(3) The addressed partition must exist.		

### 3.8.5.2 Retry Enter State

In retry enter state, an operator enter action or a Read Partition is not processed and is rejected. When the inbound transmission is acknowledged, the device returns to normal read state. In retry enter state, a Read Modified command sent as the first byte of the data stream is interpreted as a retry of INOP. Table 3-6 shows the retry enter state process.

Transition Event	Next State	Operations
Read Modified command	Retry enter	Retransmit data from partition identified by INPID.
Host acknowledge	Normal read	Terminate read operation.

### 3.8.5.3 Retry Read State

In retry read state, an operator enter action or a Read Partition is not processed and is rejected or queued. In this state, a Read command sent as the first byte of a data stream is interpreted as a retry.

When the inbound transmission is acknowledged, the device returns to normal read state. Table 3-7 shows the retry read state process.

Transition Event	Next State	Operations
Read Modified command	Retry read	Retransmit data from command INPID.

Host acknowledge	Normal read	Terminate read operation.
------------------	-------------	---------------------------

### 3.8.5.4 Read State Transitions

Table 3-8 summarizes the read state transitions.

Table 3-8. Read-State Transitions			
Read States	Normal Read	Retry	
		Enter	Read
Enter Action	2	R	R
Any Read Command	1	G	G
Read Partition	3	R	R
Host Acknowledge	1	1	--

**Key:**

R Reject, no state transition  
G Retry, no state transition  
-- No state change, no action  
1 Normal read state  
2 Retry enter state  
3 Retry read state

**Note:** Read Partition is rejected (sense code X'0871') when the device is in retry state.

## 3.9 Inbound Operation (INOP)

The display records the INOP so that it knows the operation to perform when it transmits data inbound. The display also uses INOP to know which operation to retry when a host initiated retry is received.

INOP is set as follows:

- With the exception of a Read Partition structured field directed to partition 0, whenever INPID is set to 0, INOP is set to Read Modified.
- An operator enter action, including a trigger action, sets INOP to Read Modified.
- A Read Partition structured field sets INOP to the specified

operation, specifically, Read Buffer, Read Modified, Read Modified All, Query or Query List.

- Acknowledgment of an inbound transmission sets INOP to Read Modified.

---

## 3.10 Inbound Partition Identifier (INPID)

The display records the INPID so that it determines which partition to use when it transmits data inbound. The display also uses INPID to know which partition to use when an application program-initiated retry is received.

INPID is set as follows:

- When an implicit partition is created, INPID is set to 0.
- An operator enter action, including a trigger action, sets INPID to the PID of the active partition.
- A Read Partition structured field causes INPID to be set to the PID value specified in the Read Partition structured field unless the PID value is X'FF' (Query operation). If the Read Partition structured field PID equals X'FF', INPID is left unchanged.
- Destruction of the inbound partition sets INPID to 0.

---

## 3.11 Enter Actions

If the enter action is other than a trigger action, the following conditions are set when the device (logical terminal) goes into the retry enter state:

- Either the TWAIT or PWAIT input-inhibit condition
- The System Lock input-inhibit condition
- The enter-inhibit condition.

Subtopics:

- 3.11.1 TWAIT/PWAIT Input-Inhibit
- 3.11.2 System Lock Input Inhibit
- 3.11.3 Enter Inhibit Condition
- 3.11.4 Processing of Enter Actions

---

### 3.11.1 TWAIT/PWAIT Input-Inhibit

TWAIT is used during an implicit partition state or when there is only one explicit partition. PWAIT is used when there is more than one partition. PWAIT applies only to INPID. Only one PWAIT condition can exist for a device (logical terminal). The indicator for TWAIT is "X Clock"; for PWAIT it is "X <-Clock->." The arrows of the PWAIT indicator indicate

that the operator can "go elsewhere," that is, jump to another partition. When the TWAIT condition exists, the Jump key does nothing; it is a No-op. Both TWAIT and PWAIT are input-inhibit conditions; no keystroking is allowed. TWAIT or PWAIT is set for the duration of the retry enter state. TWAIT and PWAIT *cannot* be cleared by the operator (for example, with the Reset Key).

Independent of any enter action, the TWAIT condition is set whenever the device (logical terminal) is in the SNA contention state and receives a Bid or Begin Brackets (BB). When set as a result of a Bid or BB, TWAIT is cleared by any of the following:

- A partition Write, EW, or EWA command (with keyboard restore) to any existing partition
- A partition EAU command to any existing partition
- A Write, EW, or EWA command, with keyboard restore (implicit or explicit partition 0 must exist)
- An EAU command (implicit or explicit partition 0 must exist)
- Receipt of a transmission by the device (logical terminal) that leaves it in the send or contention state (for example, CD or EB).

---

### 3.11.2 System Lock Input Inhibit

System Lock, like TWAIT and PWAIT, is an input-inhibit condition. However, unlike TWAIT and PWAIT, System Lock can be cleared by the Reset key as well as by the host.

The System Lock is raised for any enter action that raises TWAIT or PWAIT. However, the System Lock condition is overridden as long as a TWAIT or PWAIT condition exists. The System Lock indicator ("X System") is not displayed while TWAIT or PWAIT exists.

With one exception, System Lock is cleared by any Read Acknowledgment that changes the retry enter state back to the normal read state. The exception is a transmission in which the only valid Read Acknowledgment is (SNA) Change Direction (CD). The CD causes a transition to the normal read state and clears TWAIT or PWAIT. The System Lock condition remains set, however, and the "X System" indicator is displayed. The operator must use the Reset key to clear the System Lock condition before resuming keystroking. This provides a means for the host application to attract the attention of the operator, for example, to read a host message.

---

### 3.11.3 Enter Inhibit Condition

Any enter action, *including a trigger action*, sets the enter-inhibit condition for the device (logical terminal). The enter-inhibit condition allows keystroking but does not allow an enter action.

The enter-inhibit condition has significance only for the partitions other than INPID because the input-inhibit conditions associated with INPID (for

example, TWAIT, PWAIT or System Lock) override the enter-inhibit condition.

The operator can jump from INPID to one of the other partitions and enter data in that partition as long as no attempt is made to initiate an enter action. There is no indicator associated with the enter-inhibit condition. However, if an attempt is made to initiate an enter action, it is inhibited and the appropriate indicator is activated.

The Enter Inhibit remains up for the duration of the retry enter state.

---

### 3.11.4 Processing of Enter Actions

An operator enter action is processed as follows:

1. If the device has an enter-inhibit condition, the enter action is rejected.
2. If the display is in receive (RCV) state, the enter action is rejected.
3. If an input-inhibit condition exists, the action is ignored.
4. If none of the preceding conditions pertains, steps 5 through 11 are performed.
5. Enter Inhibit is activated for the device.
6. INPID is set to the PID of the active partition. For the selector pen, the INPID is set to the PID of the partition containing the detected field.
7. INOP is set to Read Modified.
8. If the enter action is anything other than a trigger action, then PWAIT and SYSTEM LOCK are activated for the inbound partition.
9. The device is placed in retry enter state.
10. The data is transmitted inbound.
11. The display is placed in RCV state.

---

### 3.12 Processing of Read Commands (Alphanumeric)

A read command (Read Buffer, Read Modified, or Read Modified All as the first byte of the data stream) is processed as follows:

1. If any of the following conditions occur, the read command is

rejected:

- The display is not in RCV or contention (CONT) state.
- The RU does not specify CD.
- The RU specifies EB.

Otherwise step 2 or 3 is performed.

2. If the device is in normal read state, the command performs the read. Data from partition 0 is transmitted inbound as defined by the following:
  - a. The command (Read Buffer, Read Modified, and Read Modified All)
  - b. The AID (RM)
  - c. The inbound reply mode of partition 0.

If partition 0 does not exist, the command is rejected. The device remains in normal read state.

3. If the device is in retry state, then the command performs a retry. Data is transmitted inbound as defined by the following:
  - a. If the command is Read Modified, and INOP specifies Query or Query List, then the appropriate Query Replies are transmitted inbound.
  - b. If the command is Read Modified, and INOP specifies Read Buffer, Read Modified, or Read Modified All, then data is transmitted from the inbound partition (INPID) as defined by:
    - 1) The INOP (Read Buffer, Read Modified, or Read Modified All)
    - 2) The AID (only when INOP=RM)
    - 3) The inbound reply mode of the inbound partition.
  - c. If the command is Read Buffer or Read Modified All, and INOP specifies Query or Query List, then the Read Buffer or Read Modified All is performed on partition 0 if it exists. Otherwise, the command is rejected. Also, the enter-inhibit condition remains in effect, and the read state does not change.
  - d. If the command is Read Buffer or Read Modified All, and INOP specifies Read Modified, Read Modified All, or Read Buffer, then data is transmitted inbound from the inbound partition (INPID) as defined by:
    - 1) The command RB
    - 2) The inbound reply mode of the inbound partition.

Subtopics:

- 3.12.1 Read Acknowledgment -- SNA

---

### **3.12.1 Read Acknowledgment -- SNA**

An inbound operation (defined as a read operation caused by a Read Partition structured field that equals Read Modified, Read Modified All, Read Buffer, Query or Query List, or by an operator enter action) is acknowledged by any transmission that, after completion, leaves the display in a send or contention state. Although the transmission can put the display in the send state, if the transmission also causes the display to send input, it is not an acknowledgment because after completing the transmission, the display is not left in a send or contention state.

In addition, the following inbound operations constitute a Read Acknowledgement or ACK:

- Query or Query List

A Query or Query List operation is acknowledged by any outbound transmission except one with a read command. The following will acknowledge a Query or Query List:

- A WSF command with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields as long as the WSF is accepted.
- An EW, EWA, Write, or EAU command with or without a WCC or data. If data is present and an error is detected in the data, the transmission is not an acknowledgment.

- An Operator Enter, or a Read Modified, Read Modified All, or Read Buffer Partition command

A Read Modified, Read Modified All, or Read Buffer Partition command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission that restores the keyboard or destroying the inbound partition. Any of the following constitutes a Read Acknowledgment (assuming the acknowledging function is supported):

- If the inbound partition is implicit or explicit partition 0, either of the following:
  - An EW, EWA, or Write command with WCC = Keyboard Restore or Reset (see note 1)
  - An EAU command.
- If in the explicit partition state, an EW or EWA command with the WCC = reset (see note 1)
- A WSF command followed by an Outbound 3270DS structured field to the inbound partition with either of the following (see note 2):
  - An EW, EWA or Write partition command with WCC = Keyboard Restore or Reset

- An EAU partition command.
  
- A WSF command followed by a Destroy Partition structured field to the inbound partition including explicit and implicit partition 0 (see note 2)
  
- A WSF command followed by a Create Partition to the inbound partition (see note 2)
  
- If in implicit partition state, a WSF command followed by a Create Partition structured field (see note 2)
  
- A WSF command followed by an Erase/Reset structured field. This applies to both implicit and explicit partition state (see note 2).

**Notes:**

1. If data follows the WCC and an error is detected in the data, the transmission is not a Read Acknowledgment.
  
  2. If there is a detected error prior to, or within the structured field providing the Read Acknowledgment, the transmission is not a Read Acknowledgment.
-

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 4.0 Chapter 4. Character Sets, Orders, and Attributes

Subtopics:

- [4.1 Introduction](#)
- [4.2 Character Sets](#)
- [4.3 Orders](#)
- [4.4 Attributes](#)
- [4.5 Processing of Character Attributes](#)
- [4.6 Defaults for Attributes](#)

---

### 4.1 Introduction

This chapter describes the character sets, orders, and attributes used by the 3270 data stream.

---

### 4.2 Character Sets

There is a maximum of 190 language-dependent graphics in any character set. Typically, there are fewer than 128 unique graphics in most languages. Those codes not corresponding to defined graphics in a character set are displayed as undefined graphics in that character set (what is displayed is an implementation option). Any data stream attributes associated with an undefined graphic still apply. If a GE code (X'08') is encountered, then the next character defined is from an alternate character set. The alternate character set, if used, is applied as the data stream is interpreted.

Graphic code point integrity must be supported. The integrity of the undefined code points (X'40' through X'FE') must be maintained. That is, if a read operation follows a write operation, assuming no intervening operator alterations, the code points that are returned are the same as those that were transmitted.

Two types of character sets are used by the 3270 data stream: nonloadable and loadable.

Subtopics:

- [4.2.1 Nonloadable Character Sets](#)
- [4.2.2 Loadable Character Sets](#)

---

## 4.2.1 Nonloadable Character Sets

These character sets are provided with the device. They are assigned an identifying number (LCID) so that session partners can designate which set to use if more than one is provided and so that nonloadable sets can be distinguished from loadable character sets. The nonloadable character set values are interpreted as follows:

- X'40' and X'0840' are a space.
- X'08nn' with nn = X'00' through X'3F' and X'FF' are control codes.
- X'41' through X'FE' and X'08nn', with nn = X'41' through X'FE', are language-dependent graphics.

---

## 4.2.2 Loadable Character Sets

When a device allows a user to define and store a character set, the sets are called *loadable character sets*. The loadable character set values are interpreted as follows:

- X'00' through X'3F' and X'FF' are control codes.
- X'40' is a nonloadable space.
- X'41' through X'FE' are loadable graphics.

There is a maximum of 190 loadable graphics. Data stream values X'41' through X'FE' are used to reference these loaded graphics.

---

## 4.3 Orders

Orders are included in the inbound and outbound data streams to provide additional control function. Unlike commands, a number of orders can occur within a chain. Orders have unique 1-byte codes and can occur in any position in the data stream. Many orders permit additional information to follow the order code, for example, buffer addresses or attributes. These longer sequences, called *control sequences*, must be contained within the RU chain. If a control sequence attempts to span a chain, it is ended on the chain boundary.

In the inbound data stream, only the following orders are permitted:

- Start Field (SF)
- Start Field Extended (SFE)
- Set Buffer Address (SBA)
- Set Attribute (SA)

- Graphic Escape (GE).

All orders can be included in the write data stream, either alone or intermixed with display or print data. Orders can be either buffer control or format control orders.

Buffer control orders are executed by the display as they are received in the write data stream. They are not stored in the character buffer. They do the following:

1. Position, define, and format data being written into the character buffer
2. Erase selected unprotected data in the character buffer
3. Reposition the cursor.

All order codes have an EBCDIC value in the range of hexadecimal 00 (X'00') through hexadecimal 3F (X'3F'). Order codes with values in this range but not defined in this chapter are rejected.

Table 4-1 lists the order codes.

Table 4-1. Order Codes		
Order	EBCDIC	ASCII
Start Field (SF)	X'1D'	X'1D'
Start Field Extended (SFE)	X'29'	Note
Set Buffer Address (SBA)	X'11'	X'11'
Set Attribute (SA)	X'28'	Note
Modify Field (MF)	X'2C'	Note
Insert Cursor	X'13'	X'13'
Program Tab (PT)	X'05'	X'09'
Repeat to Address (RA)	X'3C'	X'14'
Erase Unprotected to Address (EUA)	X'12'	X'12'
Graphic Escape (GE)	X'08'	Note

**Note:** The use of these orders requires using the full 8 bits in the associated parameter bytes; therefore, these orders are not supported in ASCII.

Subtopics:

- 4.3.1 Start Field (SF)
- 4.3.2 Start Field Extended (SFE)
- 4.3.3 Set Buffer Address (SBA)
- 4.3.4 Set Attribute (SA)
- 4.3.5 Modify Field (MF)

- 4.3.6 Insert Cursor (IC)
- 4.3.7 Program Tab (PT)
- 4.3.8 Repeat to Address (RA)
- 4.3.9 Erase Unprotected to Address (EUA)
- 4.3.10 Graphic Escape (GE)
- 4.3.11 Format Control Orders

### 4.3.1 Start Field (SF)

The SF order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

The SF order indicates the start of a field. In the read data stream, the display automatically inserts the SF order immediately before each field attribute when it responds to a Read Buffer command, enabling the application program to identify the field attributes.

In a write data stream, this order identifies to the display that the next byte is a field attribute. (The field attribute is described under "Attributes" in topic 4.4.) The display then stores the field attribute at the current buffer address and increments the buffer address by one.

The byte following the SF order in the write data stream is always treated as a field attribute.

If the display receives an SF order, it sets the associated extended field attribute to its default value.

### 4.3.2 Start Field Extended (SFE)

The SFE order is also used to indicate the start of a field. However, the SFE control sequence contains information on the field's properties that are described in the extended field attribute. The SFE order has the following format:

X'29'	Number of Attribute Type-Value Pairs	Attribute Type	Attribute Value
-------	---	-------------------	--------------------

Unlike the SF order, the field attribute does not immediately follow the order. The field attribute is included as a type-value pair in the following form:

Type	Value
X'C0'	Field Attribute

Other field properties, specifically those contained in the extended field attribute, can be included in the SFE control sequence. They are also included as type-value pairs. The number of type-value pairs is indicated in the byte following the SFE order. Valid type-value pairs for describing extended field attributes are shown under "Attribute Values and Selection Rules" in topic 4.4.6.

Each attribute type has a specified default value. When the SFE order is used, all unspecified attribute types are assigned their default value. For example, the following SFE control sequence produces a field with red characters that have normal highlighting and the default character set:

SFE	2	3270	Value	Color	Red
-----	---	------	-------	-------	-----

If SFE is sent with no type-value pairs (zero value for number of pairs), defaults are set. Attribute types and values that are unknown or cannot be maintained and returned inbound by an implementation are rejected.

All attribute types and values are checked for validity. If the same attribute type-value pair appears more than once, the last specification for a repeated attribute type takes effect.

### 4.3.3 Set Buffer Address (SBA)

The SBA order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

The SBA order specifies a new character buffer address from which operations are to start or continue. The buffer address is relative to the origin of the buffer of the partition to which the order and data are directed. The first byte of the control sequence is the SBA order; the last 2 bytes contain the 2-character buffer address:

X'11'	Buffer Address (2 bytes)
-------	--------------------------

In a write data stream, SBA orders can be used to write data into various areas of the buffer. An SBA order can also precede another order in the data stream to specify the following:

- The starting address for a PT, RA, or EUA order
- The address at which a field attribute is to be stored by an SFE or MF order
- The address at which the cursor is to be repositioned by an IC order.

If the SBA order specifies an invalid address (out of range), the write operation is terminated immediately (with a negative response).

In a read data stream sent in response to a Read Modified or Read/Modified All command, SBA orders replace field attributes. (SBA orders are not used in response to a Read Buffer command.)

When the display receives a Read Modified or Read Modified All command, it searches the buffer for field attributes with the MDT bit on. When it finds these field attributes, the display inserts into the data stream an SBA order followed by the buffer address of the field attribute +1.

In the Create Partition structured field, the partition can be defined to operate with either 16-bit addressing or 12- or 14-bit addressing. When 16-bit address mode is specified for a partition, outbound buffer addresses are interpreted as 16-bit binary (all 8 bits of each byte are used for the address), and inbound addresses are generated as 16-bit binary. If the addresses are not in the proper form, the structured field is rejected.

When 12- or 14-bit address mode is specified for the partition (implicit partition 0 is always set to 12/14), bits 0 and 1 of the first address byte following an SBA are flag bits and have the following significance:

#### **Setting Meaning**

B'00' 14-bit binary address follows  
B'01' 12-bit coded address follows  
B'10' Reserved  
B'11' 12-bit coded address follows.

When the flag bits are 00, the next 14 bits (the remainder of this byte and all 8 bits of the next byte) contain a buffer address in binary form. No address translation is necessary.

If the flag bits are 01 or 11, the next 14 bits are interpreted as a 2-character address (6 bits in each byte). The 6 low-order bits of each byte are joined to provide a 12-bit address. The address specifies the buffer position, not the line and column position on the display surface. For example, on a 480-character display, the buffer addresses are from 0 to 479. To specify a 12-bit buffer address of 160 (binary 000010100000), bits 2-7 of the first byte are set to 000010. Bits 2-7 of the second byte are set to 100000.

The flag bit setting of 10 is reserved. Receipt of a 12/14-bit buffer address beginning with the flag bits 10 causes the data stream to be rejected.

If 12/14-bit addressing mode is specified in Create Partition, the format of buffer addresses transmitted inbound depends on the size of the partition being read. If the partition is greater than 4096 characters, then all addresses from that partition are in 14-bit form. If the partition is less than 4096 characters, then all addresses are generated in 12-bit form.

When 16-bit addressing is specified, the partition always generates 16-bit addresses in the inbound data stream.

---

### **4.3.4 Set Attribute (SA)**

The SA order is used to specify a character's attribute type and its value so that subsequently interpreted characters in the data stream apply the character properties defined by the type-value pair. The format of the SA control sequence is as follows:

X'28'	Attribute Type	Attribute Value
-------	----------------	-----------------

An SA order alters the set of character attribute type-value pairs to be applied to all subsequent characters until one of the following occurs:

- A new SA order changes it.
- Another write type command is sent.
- The Clear key is pressed.
- Power at the display is switched off.

These four actions all return the established set of character attribute type-value pairs to their default value.

The attribute type X'00' is always supported by the SA order. All other attribute types are determined by the function sets being supported (see Table 4-6 in topic 4.4.5). All type-value pairs are defined under "Attribute Values and Selection Rules" in topic 4.4.6 .

An EW or EWA command resets the specified portion of the character buffer to nulls, including any field attributes, and resets any extended field attributes and character attributes associated with the nulled characters to their default values. Thus, unless a field attribute overrides the default, any subsequently interpreted characters are displayed using these defaults.

The set of type-value pairs applied during character processing is a composite, by attribute type, of the last value specified in previously encountered SA orders. Table 4-2 provides an example of attribute type-value pairs.

Command or Order	Current Character Attribute Characteristics		
	Highlight	Color	Character Set
Erase/Write Command	Default	Default	Default
Set Attribute <color><red>	Default	Red	Default
Set Attribute <char.set><X'F3'>	Default	Red	X'F3'
Set Attribute <color><blue>	Default	Blue	X'F3'
Set Attribute <highlight><blink>	Blink	Blue	X'F3'
.	.	.	.
.	.	.	.
.	.	.	.

Set Attribute <X'00'><X'00'>	Default	Defau	tDefault
---------------------------------	---------	-------	----------

Inbound data streams are generated with the assumption that the character attributes of the current set are all set to their default values. SA orders are generated as required to indicate changes. SA orders are generated only when a character attribute value changes and only for those character attributes that change and are specified in the inbound reply mode. Attribute types and values that are unknown or that cannot be maintained and returned inbound by an implementation are rejected.

### 4.3.5 Modify Field (MF)

The MF order begins a sequence that updates field and extended field attributes at the current buffer address. After the attributes have been updated, the current buffer address is incremented by one.

The MF control sequence has the following format:

X'2C'	Number of Attribute Type/Value Pairs	Attribute Type	Attribute Value
-------	---	-------------------	--------------------

Attribute types not specified remain unchanged. For example, suppose that a field is defined at buffer location xx with the following properties:

- Protected
- Alphanumeric
- Normal intensity
- Nonselectable
- Nonloadable character set
- Red
- No highlight
- No validation.

Then, interpretation of the following order sequence:

SBAxx	MF	2	Color	Blue	Highlight	Blink
-------	----	---	-------	------	-----------	-------

leaves the field with the following properties:

- Protected
- Alphanumeric
- Normal intensity
- Nonselectable
- Nonloadable character set
- Blue
- Blink
- No validation.

If the current buffer address does not contain a field attribute, the MF order is rejected.

If no attributes are specified (the number of type-value pairs field is 0) then the MF order determines if there is a field attribute at the current buffer address. If so, the current buffer address is incremented by one, and no change is made to the field's properties.

If the same attribute type appears more than once, all attribute types and values are checked for validity, but only the last value specified for a repeated attribute type takes effect.

Invalid attribute types are rejected. Attribute values that are unknown or cannot be maintained and returned inbound by an implementation must be rejected.

---

### 4.3.6 Insert Cursor (IC)

The IC order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

The IC order repositions the cursor to the location specified by the current buffer address. Execution of this order does not change the current buffer address. For example, if IC is issued when the current buffer address is 160 and the cursor is at 80, the cursor is removed from 80 and inserted at 160. The current buffer address at the end of this operation remains 160.

If the current buffer address places the cursor outside the window, the IC order causes automatic scrolling. The position of the cursor after automatic scrolling completes is described in "Scrolling" of Chapter 2, "Partitions."

---

### 4.3.7 Program Tab (PT)

The PT order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

The PT order advances the current buffer address to the address of the first character position of the next unprotected field. If PT is issued when the current buffer address is the location of a field attribute of an unprotected field, the buffer advances to the next location of that field (one location). In addition, if PT does not immediately follow a command, order, or order sequence (such as after the WCC, IC, and RA respectively), nulls are inserted in the buffer from the current buffer address to the end of the field, regardless of the value of bit 2 (protected/unprotected) of the field attribute for the field. When PT immediately follows a command, order, or order sequence, the buffer is not modified.

The PT order resets the character attribute to its default value for each character set to nulls.

The display stops its search for an unprotected field at the last location in the character buffer. If a field attribute for an unprotected field is not found, the buffer address is set to 0. (If the display finds a field attribute for an unprotected field in the last buffer location, the buffer address is also set to 0.)

To continue the search for an unprotected field, a second PT order must be issued immediately following the first one; in reply, the display begins its search at buffer location 0. If, as a result of a PT order, the display is still inserting nulls in each character location when it terminates at the last buffer location, a new PT order continues to insert nulls from buffer address 0 to the end of the current field.

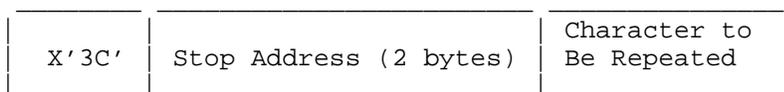
### 4.3.8 Repeat to Address (RA)

The RA order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

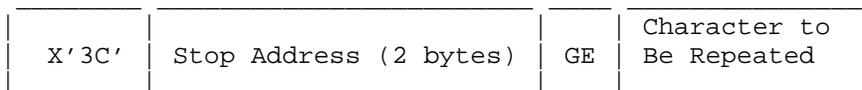
The RA order stores a specified character in all character buffer locations, starting at the current buffer address and ending at (but not including) the specified stop address.

The buffer address is relative to the origin of the buffer of the partition to which the orders and data are directed and only the character buffer locations of that partition are filled with nulls.

The stop address is identified by the 2 bytes immediately following the RA order. The character to be repeated follows the stop address. For data streams that support GE sequences or 2-byte coded character sets, the Character-to-Be-Repeated field can be two bytes long. The format of the RA order is as follows:



OR



The 12/14-bit and 16-bit addressing considerations discussed under "Set Buffer Address (SBA)" in topic 4.3.3 apply to the RA order. Also, if an invalid stop address is specified (for example, greater than 1919 for a 1920-character screen), the write operation is terminated at this point without storing the character. The character to be repeated may be any 8-bit EBCDIC code or 2-byte GE sequence. If an EBCDIC code, including control codes, has a valid graphic representation in the device, then that graphic is displayed. Otherwise, the RA order is rejected (sense code X'1003' or X'1001').

Attribute values defined by a previous SA order are applied to each repeated character.

When the stop address is lower than the current buffer address, RA wraps from the last buffer location to the first. When the stop address equals the current address, the specified character is stored in all buffer locations.

The current buffer address after successful completion of RA is the stop address, that is, one greater than the last buffer location stored into by RA.

Field attributes and their corresponding extended field attributes are overwritten by the RA order, if encountered.

---

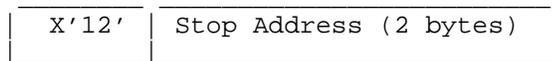
### 4.3.9 Erase Unprotected to Address (EUA)

The EUA order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

The EUA order stores nulls in all unprotected character locations, starting at the current buffer address and ending at, but not including, the specified stop address.

The buffer address is relative to the origin of the buffer of the partition to which the orders and data are directed and only the character buffer locations of that partition are filled with nulls.

The stop address is identified by the two bytes immediately following the EUA order. The format of the EUA order is as follows:



The 12/14-bit and 16-bit addressing considerations discussed under ["Set Buffer Address \(SBA\)"](#) in topic 4.3.3 apply to the EUA order.

If an invalid address is specified, the write operation is terminated at this point and no erasure (storing of nulls) or change of current buffer address occurs (a negative response of X'1005' is returned).

When the stop address is lower than the current buffer address, EUA wraps from the last buffer location to the first. When the stop address equals the current buffer address, all unprotected character locations in the buffer are erased. The current buffer address after successful execution of EUA is the stop address.

Field attributes and extended field attributes are not affected by EUA. Character attributes for every character changed to nulls are reset to their defaults.

---

### 4.3.10 Graphic Escape (GE)

The GE order is used to introduce a graphic character from an alternate character set. Support of graphic escape is optional; when not supported, it is rejected. When supported, its 2-byte format is as follows:

X'08'	Character Code
-------	----------------

The use of GE to select characters from an alternate character set follows these rules:

- The GE+character sequence always takes precedence over any other character set specified in the extended field attribute or character attribute and applies to both inbound and outbound data streams, independent of the inbound reply mode.
- GE+code points of X'00' through X'3F' and X'FF' are undefined and are rejected (sense code X'1003').
- GE+code points of X'40' through X'FE' are valid graphic character codes.
- If there is no alternate character set, the X'40' to X'FE' code points are rejected (sense code X'0863').

---

### 4.3.11 Format Control Orders

The following special control codes are classified as format control orders. These format control orders are stored in the character buffer and displayed as shown in Table 4-3.

Abbrev	Order	EBCDIC	ASCII	Displayed as
NUL	Null	X'00'	X'00'	A blank, suppressed on Read Modified
SUB	Substitu	eX'3F'	-----	A solid circle
DUP	Duplicat	X'1C'	X'1C'	An overscore asterisk
FM	Field Mark	X'1E'	X'1E'	An overscore semicolon
FF	Form Feed	X'0C'	X'0C'	A blank
CR	Carriage Return	X'0D'	X'0D'	A blank
NL	New Line	X'15'	X'0A'	A blank
EM	End of	X'19'	X'19'	A blank

	Medium			
EO	Eight Ones	X'FF'	-----	A blank

NUL is read back as a null (X'00') on a Read Buffer operation, but not read back on Read Modified operations.

NL, EM, FF, and CR are printer control codes with no display function. However, the code must be supported to the extent of being accepted and, on reading back, must appear as NL, EM, FF, and CR respectively. All are displayed as a space.

FM and DUP are displayed as above. When read back, they appear as the FM and DUP codes.

FM and DUP can be entered from the keyboard. They are stored in the display buffer as controls; the current character set selection has no effect on them. They are transmitted to the application program as control codes.

The SUB local function, Error Override, entered from the keyboard, is required only as a part of Field Validation.

---

## 4.4 Attributes

Attributes are sent in the 3270 data stream to define a field, the properties of a field, or a character or characters in a field.

The following three attribute categories are used with the 3270 data stream:

- Field attribute
- Extended field attribute
- Character attribute.

The field attribute describes the field properties and occupies the first character position of each field in the character buffer and on the screen. It is displayed as a blank.

The extended field attribute, which provides additional field properties beyond that provided by the field attribute, defines field characteristics such as color, extended highlighting, the character set to be used, field outlining and what field validation to perform.

The character attribute is associated with individual characters. Each character in the buffer (except for field attributes) has a character attribute that defines a characteristic such as the color, extended highlighting, or the character set of the character. The extended field attribute is subordinate to the character attribute; that is, if a character attribute is present and defines a color, extended highlighting, or character set value other than the default, it overrides one or more of the attribute types identified in the extended field attribute.

Neither extended field attributes nor character attributes occupy a position in the character buffer, and they are not displayed or printed. Only the attribute properties kept in the extended field attribute and character attribute are sent in the data stream. (The control sequences used to send this information are defined under "Orders" in topic 4.3.) No matter which control sequence is used, the attribute's properties are sent as a type and value pair as follows:

Attribute Type	Attribute Value
----------------	-----------------

For example, the data stream contains an order containing an attribute type of *color* and an attribute value of *green*, or a type of *extended highlighting* and a value of *blink*. This pairing of types and values is discussed later in this chapter.

Subtopics:

- 4.4.1 Field Attributes
- 4.4.2 Extended Field Attributes
- 4.4.3 Character Attributes
- 4.4.4 Conflict Resolution Between Attributes
- 4.4.5 Attribute Types and Selection Rules
- 4.4.6 Attribute Values and Selection Rules
- 4.4.7 Mandatory Fill
- 4.4.8 Mandatory Entry
- 4.4.9 Trigger

## 4.4.1 Field Attributes

A field attribute, in addition to defining the start of a field, defines the characteristics of all character locations in the field. These characteristics are shown in Table 4-4 for the 3270 field attributes.

Fields can be protected from modification by the operator or unprotected (available for the operator to modify or enter data). The unprotected definition classifies a field as an input field.

Alphanumeric fields can contain alphabetic, numeric, and symbol characters, plus space and null. Numeric fields are limited to numeric characters, the minus and decimal sign characters, and the duplicate (DUP) control. Numeric fields have special meaning for protected fields, Data Entry keyboards, and the Numeric Lock special feature.

Tab stops are at the first character position following the field attributes of unprotected fields:

Bits 0-1	2	3	4, 5	6	7
Graphic Converter	U/P	A/N	D/SPD	Reserved	MDT

Table 4-4. Bit Definitions for 3270 Field Attributes

Bit	Description
0, 1	The function of bits 0 and 1 is to make the field attribute an EBCDIC/ASCII translatable graphic character. Bits 0 and 1 are set in accordance with <u>Figure C-1 in topic C.0</u> .
2	0 Field is unprotected 1 Field is protected
3	0 Alphanumeric 1 Numeric (causes an automatic upshift of data entry keyboard)  <b>Note:</b> Bits 2 and 3 equal to B'11' cause an automatic skip of a protected field.
4,5	00 Display/not selector pen detectable 01 Display/selector pen detectable 10 Intensified display/selector pen detectable 11 Nondisplay, nondetectable (not printable)
6	Reserved. Must always be 0.
7	MDT identifies modified fields during Read Modified command operations.  0 Field has not been modified. 1 Field has been modified by the operator. MDT can also be set by a program in the data stream.

The field attribute also indicates whether the field can be detected with a selector pen and whether it is to be displayed at a brighter level (intensified) than normal. The Intensify function is the only highlighting function in the 3270 field attribute. The remaining highlighting functions are in the extended field attribute and character attribute.

Each field attribute occupies a location in the buffer but cannot be displayed. During a display, its character location appears as a space.

Field attributes are treated as characters that are protected from operator access; that is, they cannot be replaced by alphanumeric characters entered from the keyboard or modified by use of the selector pen. They can be altered, however, by the following actions:

- When the operator changes an unprotected field, which includes pressing the Erase EOF (EOF) key, the display sets the MDT (bit 7) of the field attribute to 1.
- When the operator uses the Clear key, field attributes and all other characters in a formatted buffer are set to nulls (X'00').
- When the application program writes data, field attributes are not protected from being overwritten by alphanumeric data in the data stream.
- When the Erase Input key is pressed, it sets the MDT bit off in all unprotected fields.

- When the EOF key is pressed, it sets the MDT bit on for the field in which the EOF key is used.

In an SNA/EBCDIC environment, the display ignores bits 0 and 1 in the field attributes received from the application program. Bits 0 and 1 are set in accordance with Figure C-1 in topic C.0, in attributes. This is for compatibility with previously used application programs.

---

## 4.4.2 Extended Field Attributes

Extended field attributes specify the following field properties:

- 3270 field attributes (see "Field Attributes" in topic 4.4.1)
- Field validation
- Extended highlighting
- Foreground color
- Background color
- Character set
- Field outlining
- Transparency.

The field validation functions are mandatory fill, mandatory entry, and trigger. Mandatory fill requires the terminal operator to enter a character for each character position in the field if the content of the field is modified in any way. Mandatory entry requires the operator to enter data in the field. Trigger causes the data entered by the operator to be sent to the application program as soon as the operator fills each such field.

The extended highlighting functions that can be specified in the extended field attribute are blink, underscore, and reverse video.

Foreground color allows for selection of the symbol color in a character cell when it is normally highlighted in an alphanumeric position.

Background color allows for selection of the color of the area surrounding a symbol in a character cell when it is normally highlighted in an alphanumeric partition. Refer to "Query Reply (Alphanumeric Partitions)" in topic 6.9 for more information.

Character set allows users to select the character sets to be used to display data when the user does not wish to use the default nonloadable character set provided with the device.

Field outlining allows the user to emphasize a field by putting a line on the left, right, top, and bottom of a field.

Transparency allows the user to determine whether or not the presentation space behind a character can be viewed. It can have its own value or the value of the extended field attribute, depending upon how it is used.

The attribute's color, highlighting, and character set do not apply to the field attribute position, whereas background transparency does apply.

---

### 4.4.3 Character Attributes

Character attributes specify the following character properties:

- Extended highlighting
- Extended color (referred to as color throughout this book)
- Character set.

The highlighting, color, and character set functions are identical to the functions described under "Extended Field Attributes" in topic 4.4.2 , except that they apply to a specific character rather than to the entire field.

If the character attribute contains a default for highlighting, color, or character set, the attribute value for that attribute type is obtained from the extended field attribute if it exists. The rules for when a character defaults to the field's properties are discussed in greater detail under "Conflict Resolution Between Attributes".

---

### 4.4.4 Conflict Resolution Between Attributes

Extended field attributes can be used to define the color, highlighting, field validation, and character set of the field. At times, however, there may be a conflict in the properties of the field as specified in the extended field attributes, if it exists, and the properties of individual characters in the field as specified in the character attributes. This conflict is resolved as follows:

- If there are no field attributes in the character buffer, that is, the buffer is unformatted, then the character attributes are always used in displaying or printing each character. If a character attribute specifies *default* for a particular property, the device default for that property is used in displaying or printing that character.
- If there are field attributes in the character buffer and if a character attribute specifies *default* for any character property (color, highlighting, or character set), the character is displayed using the value of that property established for the field in the extended field attribute. Otherwise, the character attribute overrides the field attribute.

Character attributes are associated with a character and not with the character's position in the buffer. Thus, whenever a character is overwritten by a new character (or cleared or erased), the old character attribute is overwritten by the character attribute of the new character. If a character is moved, for example, because of an insert or delete operation, the attribute moves with the character.

However, when nulls are converted to spaces, the associated character attribute for each resulting space should be set to:

1. The default character attribute if Entry Assist is in effect.
2. The character attributes of the nulls being converted if Null/Space Processing is in effect. This takes precedence over rule 1.

Table 4-5 summarizes the rules for resolving attribute conflicts.

Table 4-5. Rules for Resolving Attribute Conflicts		
If the screen is	And the character attribute is set to(1)	Then
Formatted	Default	The extended field attribute is used to display/print that character.
Formatted	Nondefault	The character attribute is used to display/print that character.
Unformatt	dDefault or nondefault	The character attribute is used to display/print that character.

(1) Each attribute type (color, highlighting, character set) is treated uniquely.

#### 4.4.5 Attribute Types and Selection Rules

Extended field and character attributes are sent in the data stream as an attribute type-value pair following the SA, SFE, or MF order sequence.

Attribute types are coded according to the following rules:

X'00' - X'7F' Attribute value is a 1-byte binary number; it does not have bit significance.

X'C0' - X'FF' Attribute types have values that are bit-encoded (for example, Field Validation).

Table 4-6 lists the attribute types defined in this chapter. Attribute types not shown are reserved and are rejected (sense code X'1003').

Table 4-6. Attribute Types			
Type Code	Attribute Type	May be used to define:	
		Extended Field Attribute	Character Attribute
X'00'	All character attributes	NA	Yes
X'C0'	3270 Field attribute	Yes	NA
X'C1'	Field validation	Yes	NA
X'C2'	Field outlining	Yes	NA

X'41'	Extended highlighting	Yes	Yes
X'42'	Foreground color	Yes	Yes
X'43'	Character set	Yes	Yes
X'45'	Background color	Yes	Yes
X'46'	Transparency	Yes	Yes
Other	Reserved	NA	NA
<b>Note:</b> NA = Not Applicable			

---

## 4.4.6 Attribute Values and Selection Rules

The value bytes for all defined attribute types are shown in the following pages. Attribute values not shown are rejected. The attribute value of X'00' is mandatory for all supported attribute types. Attribute values that are unknown or cannot be maintained and returned inbound by an implementation also must be rejected.

Subtopics:

- [4.4.6.1 All Character Attributes](#)
- [4.4.6.2 3270 Field Attribute](#)
- [4.4.6.3 Extended Highlighting](#)
- [4.4.6.4 Color](#)
- [4.4.6.5 Character Set](#)
- [4.4.6.6 Field Outlining](#)
- [4.4.6.7 Transparency](#)
- [4.4.6.8 Field Validation](#)

---

### 4.4.6.1 All Character Attributes

All character attributes reset all character attribute types that are specifiable in the SA order to their default value. Attribute types affected are color, highlighting, and character set. The only valid value setting is X'00'; all others are reserved:

Type	Value
X'00'	Reset All Attribute Types (X'00')

The attribute type X'00' can appear only in the SA order.

---

### 4.4.6.2 3270 Field Attribute

The 3270 field attribute identifies the value byte as a field attribute:

Type	Value
X'C0'	Field Attribute

The definition of each bit position of the field attribute value byte is shown in Table 4-4 in topic 4.4.1.

If the 3270 field attribute specifies non-display, it overrides any extended field or character attribute in respect to displaying or printing that field. However, the attribute values specified in the extended field attribute or the character attribute are transmitted inbound.

If the 3270 field attribute specifies protected and the extended field attribute specifies mandatory fill, mandatory fill is ignored.

### 4.4.6.3 Extended Highlighting

Extended highlighting attributes identify the highlighting property of a field or character. The field can have only one highlighting property specified by the extended field attribute (such as blink or reverse video but not both). A highlighting property specified by the extended field attribute does not affect the intensify property specified by the field attribute.

When a character is assigned a highlighting property using the character attribute, the character's property overrides (for that character) the property defined by the extended field attribute:

Type	Value
X'41'	Highlighting

The following table lists valid settings for the highlighting value byte. All other settings are reserved.

#### Setting Highlighting Property

X'00' Default  
 X'F0' Normal (as determined by the 3270 field attribute)  
 X'F1' Blink  
 X'F2' Reverse video  
 X'F4' Underscore.

### 4.4.6.4 Color

Color attributes identify the color properties of a field or character. The field may have only one color property.

When a character is assigned a color property using the character attribute, the character's property overrides (for that character) the property defined by the extended field attribute:

Type	Value
X'42'	Foreground color
X'45'	Background color

The following are valid settings for the color value byte:

**Setting Color Property**

X'00' Default color (defined by Query Reply [Color])

X'F7' Neutral

All others As indicated by Query Reply (Color).

The relationship of a particular color to a data stream value is indicated by the Query Reply (Color) structured field. A device not capable of displaying a particular color may accept the data stream value for the color and display the device default color indicated in the Query Reply (Color) structured field.

Table 4-7 lists the colors and their architected color identifications.

Color	I.D.	Defined
Neutral	X'F0'	(black on displays, white on printers)
Blue	X'F1'	
Red	X'F2'	
Pink	X'F3'	
Green	X'F4'	
Turquoise	X'F5'	
Yellow	X'F6'	
Neutral	X'F7'	(white on displays, black on printers)
Black	X'F8'	
Deep Blue	X'F9'	
Orange	X'FA'	
Purple	X'FB'	
Pale Green	X'FC'	
Pale Turquoise	X'FD'	
Grey	X'FE'	
White	X'FF'	

The 'color' listed above as Neutral with a color identification of X'F7' is defined as White for a display and Black for a printer.

The settings X'00' and X'F7' have unique data stream meanings. The X'00' value selects the device default color indicated in the Query Reply (Color) structured field. The X'F7' value indicates that the color is defined by a triple-plane character set. If a single-plane or nonloadable character set is referenced, the color defaults to the single color specified for the X'F7' value by Query Reply (Color). With triple-plane character sets, the color of each character is composed by combining the red, green, and blue planes defined for that character in the field. Where the planes overlap, the color that is displayed or printed is an additive color, that is, it comprises all colors in the overlap. For example, for a display, if a blue dot and a green dot overlap, the result is a turquoise dot if the color attribute value is X'F7'. If the color attribute value is other than X'F7', the character is a single color.

If a device receives and accepts the color attribute type (X'42' and X'45'), only the colors specified in Query Reply (Color) are used in the presentation of data.

For additional information, see "Query Reply (Color)" in topic 6.13.

---

#### 4.4.6.5 Character Set

Character set attributes identify the character set to be used by the device to display the field or character. The field can have only one character set LCID assigned, but when a character (or characters) is assigned an LCID, the character's ID overrides (for that character) the field's LCID:

Type	Value
X'43'	Character Set

Following are valid settings for the character set value byte:

Setting	Character Set
X'00'	Default character set
X'40' to X'EF'	Local ID for loadable character sets
X'F0' to X'F7'	Local ID for nonloadable character sets
X'F8' to X'FE'	Local ID for 2-byte coded character sets (see <u>Chapter 12, "Double-Byte Coded Character Set (DBCS)-Asia"</u> )
All others	Reserved.

The contents of the nonloadable character sets X'F0' through X'FE' cannot be copied or read from the device. X'FF' signifies a free character set and may appear as an LCID for any character set, but it is invalid if sent as a value byte (sense code X'1003').

For more information on the LCID, see "Load Programmed Symbols (Load PS)" in topic 5.13.

---

#### 4.4.6.6 Field Outlining

Field outlining is the displaying and printing function of the field frame by the combination of horizontal and vertical lines to improve the readability of the screen and print-out. Field outlining is used to highlight data in tabular form, for example, to show the location of input fields:

Type	Value
X'C2'	Field outlining

Field outlining is specified for fields only and surrounds the field by any combination of overline, underline, vertical line on the left edge of the field, and vertical line on the right edge. The vertical line on the left edge of the field is drawn in the field attribute position. The vertical line on the right edge of the field is drawn in the next field attribute position. The height of both vertical lines is from the overline to the underline. The overline is drawn between the previous row and the current row. The underline is drawn between the current row and the next row. The length of both horizontal lines is from the left vertical line to the right vertical line. Four lines make a complete and solid rectangle when all of the four lines are specified to the field.

When the right vertical line is specified for a field and the left line is specified for the next field, these two vertical lines must be overlaid. When the underline is specified for a field and the overline is specified for a field on the next row, these two horizontal lines must be overlaid.

Field outlining is always displayed with normal intensity, non-reversed, non-blinking, default fixed color.

The following are 16 kinds of field outlining available by using combinations of the 4 lines. The different combinations are described below:

- B'00000000' No outlining lines
- B'00000001' Underline only
- B'00000010' Right vertical line only
- B'00000100' Overline only
- B'00001000' Left vertical line only
- B'00000011' Underline and right vertical line
- B'00000101' Underline and overline
- B'00001001' Underline and left vertical line
- B'00000110' Right vertical line and overline
- B'00001010' Right and left vertical lines
- B'00001100' Overline and left vertical line
- B'00000111' Rectangle minus left vertical line
- B'00001011' Rectangle minus overline
- B'00001101' Rectangle minus right vertical line
- B'00001110' Rectangle minus underline
- B'00001111' Rectangle.

---

#### 4.4.6.7 Transparency

Transparency attributes can designate that the space surrounding a character be transparent to the screen operator, or it can designate that the space be opaque:

Type	Value
------	-------

X'46'	Transparency
-------	--------------

These are the possible settings that the transparency attribute can have:

**Setting Meaning**

- X'00' Default
- X'F0' Background is transparent (OR)
- X'F1' Background is transparent (XOR)
- X'FF' Background is opaque (non-transparent).

#### 4.4.6.8 Field Validation

Field validation attributes define the validation properties of the field. The field can have more than one property, and all are optional:

Type	Value
X'C1'	Field Validation

Valid bit settings for the field validation value byte are as follows:

Bit	Setting	Meaning
0-4	B'0000'	Reserved, must be zero
5	B'1'	Mandatory fill
6	B'1'	Mandatory entry
7	B'1'	Trigger.

The field validation functions of mandatory fill, mandatory entry, and trigger are controlled by the extended field attribute at the device.

The SFE and MF orders are the data stream orders used to transmit the types of field validation desired. The attribute type is X'C1', and the type-value pairs that can be specified are described under "Attribute Values and Selection Rules" in topic 4.4.6.

If no type-value pairs are transmitted, no field validation occurs. A field can be given more than one field validation property.

#### 4.4.7 Mandatory Fill

Mandatory fill specifies that, if any data is entered into a field, then the field must be completely filled. The operator, for example, fills a field by doing the following:

- Entering data in every character position of the field
- Not entering data in every position but entering a DUP character or an error character in the last position filled.

An unprotected fill field is *primed* when the operator positions the cursor in the field and presses any of the following keys:

- Any key for entering data
- The Delete key
- The Erase EOF key.

A mandatory-fill field is *unprimed* by any of the following:

- A write operation to that partition
- An operator erase-input keystroke
- Successful fill validation for that field.

Fill validation occurs when the operator tries to move the cursor out of a primed mandatory-fill field. Similarly, when the operator tries to transmit data, if the cursor is in a primed mandatory-fill field, then fill validation occurs.

Before a field is primed or after it is unprimed, the operator can move the cursor through the mandatory-fill field without causing fill validation.

Fill validation checks the field for a DUP character or for null characters. If the field contains a DUP character, fill validation is successful and the keystroke is processed. If the field contains any null characters the following occurs:

- The mandatory-fill input-inhibit condition is activated.
- The alphanumeric cursor is not moved out of the field.

If the field contains no null characters, the keystroke that caused fill validation is processed normally.

The Reset key must be used to remove the input-inhibit condition. After pressing Reset, the operator can proceed to fill the field (with data, blanks, and the SUB and DUP characters).

**Notes:**

1. With multiple partitions, the Jump Partition key is not affected by the validation of a mandatory-fill field; that is, while the screen cursor is positioned within a mandatory-fill field, the operator can jump to another partition. The screen cursor moves, but the current cursor position of the partition does not change. Consequently, the Jump Partition key does not cause fill validation to occur.
2. Similarly, the Jump Partition key does not affect validation. If the operator causes the MDT to be set to 1 and then jumps to another partition and back again, fill validation applies only at field exit time.
3. If a field has an extended field attribute with mandatory fill and trigger set on, fill validation occurs before the trigger action.
4. If the cursor is positioned in a primed mandatory-fill field, any

enter action, for example pressing a PF key, causes fill validation to occur.

5. If mandatory fill is specified for a protected field, it has no effect on the field.
6. If the host sends to the display the MDT bit in a mandatory-fill field, the field is transmitted inbound on a Read Modified operation. Thus, if the operator does not type any data into such a field, this field, which may contain some nulls (that is, an incomplete fill field), is transmitted inbound.
7. Similarly, if a mandatory-fill field is selectable, operator selection causes the MDT bit to be set on. Again, if the operator does not type any data into such a field, the incomplete fill field is transmitted inbound on a Read Modified operation.

---

## 4.4.8 Mandatory Entry

Mandatory entry attributes specify that the terminal operator cannot transmit character data without first making an entry in the field. The operator can transmit character data using:

- The Enter key
- A PF key
- A selector-pen detect on a field with a designator of &
- A cursor select on a field with a designator of &
- A magnetic slot reader.

When the operator tries to transmit data, the extended field attributes and field attributes are examined to determine if any field has both the mandatory-entry and unprotected attribute properties. If such a field exists for which the MDT bit is 0, no data is transmitted and an input-inhibit condition is raised.

The cursor is moved to the first field in the presentation space that has an MDT bit of 0 and the unprotected and mandatory-entry properties. The operator must use the Reset key to remove the input-inhibit condition. The operator can then proceed to enter data (at least one character, blank, SUB, or DUP). The operator must redo the enter action for the data that was inhibited.

---

## 4.4.9 Trigger

Trigger attributes specify that the host is to be notified (with an AID) once the terminal operator has finished entering character data in the field. The trigger property can be specified for any field. Trigger fields allow field-by-field validation and editing to be performed by the host. As the operator completes each such field, the field is transmitted to the host for validation. The host can then give an affirmative or negative reply. The details of this process are described below.

A trigger field is *primed* when the operator positions the cursor in the field and presses a data entry key, the Delete key, or the Erase EOF key.

A trigger field is *unprimed* by a write operation to that partition, by an operator erase-input keystroke, or by the trigger action for that field.

Trigger action occurs when the operator tries to move the cursor out of a primed trigger field. Before a field is primed, or after it is unprimed, the operator can move the cursor through the trigger field without causing a trigger action. Trigger action causes AID X'7F' to be sent to the application program together with the cursor address, field address, and field data.

The trigger action is caused by any keystroke (for example, TAB, UP, DOWN, or data entry) that would move the cursor out of the trigger field. This keystroke determines the next position of the cursor. However, the cursor remains in the trigger field while the host is validating the field. If an affirmative reply is received, then the cursor is moved to the next position unless it is overridden by an IC order contained in the reply.

While the display is awaiting a reply from the host, the operator can continue to type data. All keystrokes are accepted and queued (except as listed below) until a reply is received. The cursor is not moved, however, and keystrokes are not displayed until an affirmative reply is received from the host.

The attention and system request keystrokes are not queued but are acted on immediately. In addition, system request purges the queue.

The reply from the host can be affirmative or negative. A null RU chain with CD or EB, or a write operation with keyboard restore to the partition containing the trigger field, is an affirmative reply. This reply causes resumption of keystroke processing from the keystroke queue.

The queued keystrokes are processed after the operation is completed; that is, after the orders and data have been processed. If the write operation is caused by an Outbound 3270DS structured field in a WSF command, the queued keystrokes are processed after the Outbound 3270DS structured field is processed and before any subsequent structured fields in WSF are processed.

A null RU without CD or EB is not a reply or a read operation to any partition. A Read Partition structured field is rejected because the device is in retry state.

A read command is not a reply, but it is not rejected if one of the following occurs:

- The RM command transmits the trigger field.
- The RMA command transmits all fields that have the MDT bit set on.

Normally, the MDT bit will be set on (as a result of data entry). If the field is selectable, however, then deselection resets the MDT bit. Consequently, it is possible that the MDT bit is 0 when a trigger field is transmitted inbound.

The keystroke queue is unaffected by the operation and keystroke queuing continues. This permits the host application to obtain data, other than the trigger field itself, that may be needed for validation of the field.

Any other operation is a negative reply. The negative reply causes the keystroke queue to be purged and an input-inhibit condition to be raised. (Typically, the host might perform a write, without keyboard restore, to display an error message, highlight the field or character in error, and reposition the cursor.)

Details of the possible replies are given below. The terms and concepts used here are explained in "Read Operations from Partitions" in topic 3.7.

In summary, an affirmative reply is as follows:

- A WSF command containing an Outbound 3270DS structured field, addressed to the inbound partition (INPID). The structured field specifies a write command and contains a WCC that specifies keyboard restore.
- A W command with a WCC that specifies keyboard restore, if INPID is 0.
- A null RU chain with CD or EB.

In summary, a negative reply is as follows:

- A WSF command containing any of the following structured fields:
  - Outbound 3270DS, addressed to INPID, that specifies a write partition command and contains a WCC that does not specify keyboard restore.
  - Outbound 3270DS addressed to INPID that specifies an EW or EWA command.
  - Outbound 3270DS addressed to a partition where the PID is not INPID.
  - A control operation, namely, those operations requested by the Reset Partition, Load Programmed Symbols, Set Inbound Reply Mode, Create Partition, Destroy Partition, Activate Partition, and Set Window Origin structured fields.
- A W command with a WCC that does not specify keyboard restore, or an EAU command, if INPID is 0.
- A W, EW, EWA, or EAU command, if INPID is nonzero.
- A WSF command with no data included in the structured field.

A read operation does not constitute a reply, and the following are not replies:

- A Read Modified, Read Modified All, or Read Buffer command.
- A Read Partition structured field contained in a WSF command.

Where a WSF command contains several structured fields, the first

structured field is interpreted as the reply; a subsequent structured field is not regarded as a reply.

If the reply from the host is a write operation that includes an IC order, then the cursor is positioned at the address specified in the data. If the reply is negative, the cursor is moved from its position within the trigger field to where the host has placed it, and the invalid field input-inhibit condition is raised.

**Note:** These replies apply only if the command is processed successfully; that is, if there are no data stream errors.

Subtopics:

- 4.4.9.1 Programming Notes

---

## 4.4.9.1 Programming Notes

When programming, you should be aware of the following facts about trigger fields:

- To avoid operator confusion, the application program should not move the cursor out of the trigger field on a negative reply.
- If the reply is affirmative, the cursor is moved from its position within the trigger field to where the application program has placed it, and the keystroke queue is processed. Any data entry keystrokes are placed at the host-specified position.
- The ability to move the cursor allows logical cursor movement to be programmed; that is, the position of the cursor can be made dependent on the data content of a previous field. If such a function is implemented, you should ensure that the operator understands the protocol for that application.
- Trigger action occurs if scrolling action moves the cursor out of a primed trigger field.
- If the operator performs an enter action when an unmodified mandatory-entry field exits, the mandatory-entry input-inhibit condition is activated, and the cursor moves to the mandatory-entry field. Thus, if this occurs while the cursor is positioned in a trigger field, the cursor movement causes a trigger action.

An affirmative reply from the host causes the queued keystrokes to be processed. In this case, however, they are generally ignored because of the input-inhibit condition.

In this situation, moreover, and only in this situation, if the reply from the host was a write operation containing an IC order, the mandatory-entry field cursor position overrides the host-specified cursor position. In other words, after the reply from the host, the cursor is positioned in the mandatory-entry field and the mandatory-entry input-inhibit indicator is displayed.

- A change of the active partition does not cause trigger action.
- If the keystroke queue becomes full before a write is received from the host, an input-inhibit condition is raised.
- The MDT bit is not changed by the trigger action. The host must decide what to do when a field check fails and then issue the appropriate orders in the data stream.
- Normally, the MDT bit is set on as a result of data entry. If the field is selectable, however, deselection resets the bit. Consequently, it is possible that the MDT bit is 0 when a trigger field is sent inbound.
- With multiple partitions, the Jump Partition key is unaffected by the validation of a trigger field; that is, while the screen cursor is positioned within a trigger field, the operator can jump to another partition. The screen cursor moves, but the current cursor position of the partition does not change. Consequently, the Jump Partition key does not cause a trigger action. Similarly, validation is not affected by the Jump Partition key.
- If a field has both the mandatory fill and the trigger attributes, fill validation occurs before the trigger action.
- If the cursor is in a trigger field when the operator performs an enter action, this does not cause a trigger action unless an empty mandatory-entry field exists.

---

## 4.5 Processing of Character Attributes

When an extended field attribute is used to describe properties of a field and that field also has characters described by character attributes, the character attribute overrides the extended field attribute for that character. The term *character attribute*, as used in the processing discussion below, applies to the effective character attribute determined from the interaction of the extended field attribute and the character attribute.

For all cases except where a triple-plane character set is referenced and the color value byte is X'F7' (neutral), the processing is as follows:

1. If the character attribute contains an ID for a *single-plane* character, the character code point and the character set value byte are used to obtain the character definition.
2. If the character attribute contains an ID for a *triple-plane* character set, the character code point and the character set value are used to obtain *three* character definitions, which are then combined to produce a single definition.
3. If the character attribute has a highlight value of *reverse video*, then the character definition obtained from step 1 or 2 is complemented; that is, all 0 bits are changed to 1's, and all 1 bits are changed to 0's (assuming the device supports reverse video).

highlighting).

4. If the character attribute has a highlight value of *underscore*, then an underscore bar is effectively combined into the character definition obtained from step 1 or 2.

Where a triple-plane character set is referenced and the color value byte is X'F7', the processing is as follows:

1. The character code point and the character set value byte are used to obtain three character definitions, one from each of the red, green, and blue planes.
2. If the character attribute has a highlight value of *reverse video*, then each of the three character definitions is complemented (assuming that the device supports *reverse video*).
3. If the character attribute has a highlight value of *underscore*, then an underscore bar is effectively combined into each of the three character definitions obtained from step 1.
4. Each of the three character definitions is displayed in the color associated with its plane.

Where there is an overlap in character definitions, that is, if a dot is defined on more than one plane, then the color displayed for that dot is an additive color, as shown in the following table (except as defined in note 2):

Overlap in	Displays/Prints as
Red and Blue	Pink
Blue and Green	Turquoise
Green and Red	Yellow
Green, Red, and Blue	Neutral

This is illustrated in the following example:

@..BB...	.....	.....	...BB...
@..BB...	.....	.....	...BB...
@..BB...	.....	.....	...BB...
@..BB...	.....	GGGGGGGG	GGGTTGGG
@..BB...	.....	GGGGGGGG	GGGTTGGG
@..BB...	.RRRRRR.	GGGGGGGG	GYYNNYYG
@..BB...	.RRRRRR.	GGGGGGGG	GYYNNYYG
@..BB...	.RRRRRR.	.....	.RRPPRR.
@..BB...	.RRRRRR.	.....	.RRPPRR.
@..BB...	.....	.....	...BB...
Blue	Red	Green	Result

Where: G = Green, R = Red, B = Blue, T = Turquoise, P = Pink,  
 N = Neutral, Y = Yellow, and . = none.

**Notes:**

1. Where identical definitions occur in all three planes, the operation of a triple-plane character set has the same result as if a single-plane character set had been referenced.
2. On devices that do not support a specified color, including colors that result from color-plane overlap, that color is displayed or printed in the device default color. (See "Query Reply (Color)" in topic 6.13 for more information.)

## 4.6 Defaults for Attributes

Extended field attributes always assume the default condition when coded X'00'. Character attributes assume defaults somewhat differently. If the character to which the character attribute applies is in a field (see the "Formatted Screen" column in Table 4-8) the X'00' code specifies that the properties associated with the field are to be used.

Table 4-8 shows the attribute defaults.

Table 4-8. Attribute Default Conditions			
Attribute Type	Default Condition		
	For a Field	For a Character	
		Formatted Screen	Unformatted Screen
Field attribute	Unprotected, A/N, attribute display/nondetect, no intensify, MDT off	Not applicable	Not applicable
Color (foreground and background)	As specified in Query Reply (Color)	Default to field color	As specified in Query Reply (Color)
Extended highlight	As specified in Query Reply (Highlight)	Default to field highlight	As specified in Query Reply (Highlight)
Character set	Default to character set X'00'	Default to field character set	Default to character set X'00'
Field validation	None	Not applicable	Not applicable
Field	No outlining line	Not applicable	Not applicable

outlining			
Transparen	yAs specified in Query Reply (Field outlining)	Default to field background transparency	As specified in Query Reply (Background Transparency)

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 5.0 Chapter 5. Outbound/Inbound and Outbound Structured Fields

Subtopics:

- 5.1 Introduction
  - 5.2 Structured Field Grouping
  - 5.3 Structured Field Self-Defining Parameters
  - 5.4 Outbound Structured Fields
  - 5.5 Activate Partition
  - 5.6 Begin/End of File
  - 5.7 Create Partition
  - 5.8 Destroy Partition
  - 5.9 Erase/Reset
  - 5.10 Load Color Table
  - 5.11 Load Format Storage
  - 5.12 Load Line Type
  - 5.13 Load Programmed Symbols (Load PS)
  - 5.14 Modify Partition
  - 5.15 Outbound Text Header
  - 5.16 Outbound 3270DS
  - 5.17 Present Absolute Format
  - 5.18 Present Relative Format
  - 5.19 Read Partition
  - 5.20 Request Recovery Data
  - 5.21 Reset Partition
  - 5.22 Restart
  - 5.23 SCS Data
  - 5.24 Select Color Table
  - 5.25 Select Format Group
  - 5.26 Set Checkpoint Interval
  - 5.27 Set MSR Control
  - 5.28 Set Partition Characteristics
  - 5.29 Set Printer Characteristics
  - 5.30 Set Reply Mode
  - 5.31 Set Window Origin
  - 5.32 Type 1 Text Outbound
  - 5.33 Outbound/Inbound Structured Fields
  - 5.34 Data Chain
  - 5.35 Destination/Origin
  - 5.36 Object Control
  - 5.37 Object Data
  - 5.38 Object Picture
  - 5.39 OEM Data
  - 5.40 Save/Restore Format
  - 5.41 Select Intelligent Printer Data Stream (IPDS) Mode
-

## 5.1 Introduction

This chapter describes the outbound/inbound and outbound structured fields that can be included in the 3270 data stream. To provide additional controls and transmit various data types other than characters, it is necessary to use structured fields. The structured field syntax permits variable length data and controls to be encoded in such a way as to facilitate processing a sequence of fields into component fields without having to examine every byte.

Variable-length structured fields are achieved by providing a length as the first parameter of the structured field, as follows:

```
|<----structured field 1----->|<-----structured field 2---->|
```



```
|<-----length 1----->|<-----length 2----->|
```

In the 3270 data stream, outbound structured fields are introduced with the WSF command. This command does not contain explicit control information as do other 3270 commands; it simply means all subsequent data is in a structured field format. A WCC does not follow the WSF command as it does other write type commands.

Following the WSF command, all data in the transmission must be in structured field format. Figure 5-1 shows the form of a structured field.

**PICTURE 8**

Figure 5-1. Structured-Field Format

Each structured field contains a 2-byte length field. This field defines the length of the structured field (including the length bytes). Next follows an ID field that defines the function of the structured field, followed by parameters and data in the format defined by the ID code. A length of zero on an outbound or inbound 3270DS indicates one of the following:

- The length of the structure was not determined.
- This is the last or only structured field in the transmission (for SNA, the transmission = chain).
- The length of the structured field should be determined using the end of the transmission (for SNA, transmission = chain).

Except for the use of a Length parameter value of zero, a structured field with a one-byte type parameter will be rejected if the Length field value is less than three. For a structured field with a two-byte type field, the minimum Length field value is four. This means that a structured field requires at least an ID. Thus, sending only the Length field (i.e. 0000), as a zero length structured field is invalid.

The outbound structured fields are described under "Outbound Structured Fields" in topic 5.4. The fields and their ID codes are shown in alphabetical order in the following list. Major operations within the field are listed by their operation code.

ID	Name	ID	Name
0E	Activate Partition	01 nn	Read Partition
0F85	Begin/End of File	nn= 02	Query
0C	Create Partition	nn= 03	Query List
0D	Destroy Partition	nn= 6E	Read Modified All
03	Erase/Reset	nn= F2	Read Buffer
0F05	Load Color Table	nn= F6	Read Modified
0F24	Load Format Storage	1030	Request Recovery Data
0F07	Load Line Type	00	Reset Partition
06	Load Programmed Symbols	1033	Restart
0F0A	Modify Partition	41	SCS Data
0F71	Outbound Text Header	0F04	Select Color Table
40	Outbound 3270DS	4A	Select Format Group
4B	Present Absolute Format	1032	Set Checkpoint Interval
4C	Present Relative Format	0F01	Set MSR Control
0F08	Set Partition Characteristics	0F84	Set Printer Characteristics
09	Set Reply Mode	0B	Set Window Origin
0FC1	Type 1 Text Outbound		

The outbound/inbound structured fields are described under "Outbound/Inbound Structured Fields" in topic 5.33. The fields and their ID codes are shown in alphabetical order in the following list:

ID	Name
0F21	Data Chain
0F02	Destination/Origin
0F11	Object Control
0F0F	Object Data
0F10	Object Picture
0F1F	OEM Data
1034	Save/Restore Format
0F83	Select IPDS Mode.

---

## 5.2 Structured Field Grouping

Structured fields cannot span transmission. An error occurs if a transmission ends before a structured field's length count is satisfied or if an entire control sequence does not appear in the same structured field. However, you can group related structured fields into a single logical entity. When structured fields are grouped as described in this section, the non-spanning rules apply on a group basis rather than on a structured field basis.

Grouping applies only to certain structured fields. Those structured fields that can be grouped must have a Group parameter in bits 0 and 1 of the first flag byte. The definition of the Group parameter is as follows:

B'00' Continue  
B'01' End  
B'10' Begin  
B'11' Only.

In the first structured field of a group, the Group parameter is set to Begin. The group parameter is set to Continue in the second through the next-to-last structured fields of the group. The group parameter is set to End in the last structured field of the group. When the structured field is not a member of a group of related structured fields, the group parameter is set to Only; the non-spanning rules apply in this case.

In addition to the explicit termination of a structured field group by a group parameter set to End, the following terminates (without error) all groups in process to or from the device:

- An EW/EWA command with WCC = Reset
- Bind (SNA only)
- Power on reset
- Clear Local Function (for example, the Clear key)
- Copy command (BSC printer only).

Subtopics:

- 5.2.1 Spanning
- 5.2.2 Data Chaining (Non-SNA)

---

## 5.2.1 Spanning

Spanning is an application of the structured field grouping functions. The spanning application is the more general use of structured field grouping. Spanning allows the grouping of multiple structured fields of the same type into a single logical entity. The most significant use of spanning is in situations where the data to be sent exceeds the single structured field length limit (32B) and sending multiple independent structured fields is not acceptable.

The data to be sent can be divided into multiple structured fields without regard to control sequence boundaries; that is, control sequences may span structured fields and transmissions within the group.

When more than one transmission is used to send the group of structured fields utilizing spanning, the following apply:

- The length count of the last structured field in a transmission must be satisfied at the end of the transmission. That is, a structured field cannot span transmissions (except if the Data Chaining structured field is used in conjunction with spanning, limited to the non-SNA environment).
- In SNA on outbound (to the device) more than one transmission can be sent without sending a CD until the last transmission of the group.

The last structured field of a transmission can use a length value of

zero. The non-spanning rules always apply at the end of the last structured field of the group.

There can be more than one structured field group in process at the same time to or from a device providing either of the following is met:

- The structured field groups have different structured field types.
- If two or more of the structured field groups have the same structured field types, these groups must be associated with different partitions.

Similarly, individual (non-grouped) structured fields can be interspersed with group members, to or from a single device, provided either of the following is met:

- The individual structured fields are of a different type than any of the structured fields being grouped.
- Any individual structured field that has the same type as one or more of the structured fields being grouped, must be associated with a different partition than that of the group or groups.

On outbound (to the device) transmissions, the following causes termination of a group in process and rejection of the transmission.

- Receiving a structured field with the Group parameter = Continue or End when the group is not in existence.
- Receiving a structured field with the Group parameter = Begin for a partition that already has a group, with the same structured type, in process.
- Receiving an individual (non-grouped) structured field for a partition which has a group, with the same structured field type, in process.
- Receiving a transmission starting with any 3270 command except WSF or EW/EWA (with WCC = Reset).
- A structured field length count is not satisfied at the end of a transmission.

On inbound (from the device) transmissions, the following are invalid:

- Sending a structured field with the Group parameter = Continue or End when the group is not in existence; that is, group was not started by a previous structured field with Group parameter = Begin.
- Sending a structured field with the Group parameter = Begin from a partition which already has a group, with the same structured field type, in process.
- Sending an individual (non-grouped) structured field from a partition that has a group, of the same type structured fields, in process.

- Sending a transmission where the length count of a structured field is not satisfied at the end of the transmission.
- When one or more structured field groups are in process, sending a transmission that does not start with AID X'88'.

---

## 5.2.2 Data Chaining (Non-SNA)

Data chaining is a unique application of structured field grouping; its use is limited to the Data Chain structured field.

The SNA protocols provide a chaining function that allows a long message to be divided up into small transmissions to match the capability of a device. The division can be done without regard to control sequence or structured field boundaries. Non-SNA protocols do not provide this type of function; however, a chaining-like function is provided at the data stream level by use of the structured field grouping function in the Data Chain structured field.

The data chaining function allows the grouping of structured fields of different types, including structured fields that do not support the Group parameter. The data chaining is provided by the Data Chain structured field, which uses the Group parameter to provide the chaining control. The data to be sent (which must be in structured field form) can be divided without regard to structured field or control sequence boundaries into a number of transmissions of a size to accommodate the device capability.

A WSF and a Data Chain structured field are added at the start of each transmission. The length value (X'06') of the Data Chain structured field covers just the Data Chain structured field; it does not include the rest of the data in the transmission.

The Data Chain structured field in the first transmission has the Group parameter = Begin. In the next through the next-to-last transmission, the Group parameter is set to = Continue. In the last transmission, the Group parameter = End. In the first transmission, a structured field must start immediately following the Data Chain structured field. However, in subsequent transmissions, this is not required. The data following can be a continuation of a structured field started in a previous transmission. That is, the length count of a structured field can span transmissions. If the length count of a structured field is not satisfied at the end of a transmission (excluding the last transmission of a data chain), the remainder of the data to satisfy the count starts immediately *after* the Data Chain structured field in the next transmission. The length count of a structured field could in some cases carry across a number of transmissions.

In addition to independent structured fields, the data being chained can contain one or more groups of structured fields using the spanning function.

On outbound (to the device) transmissions, any of the following causes termination of the data chain and rejection of the transmission:

- Receiving a Data Chain structured field with Group parameter =

Continue or End without having previously started a data chain with a Data Chain structured field with the Group parameter = Begin

- Receiving a Data Chain structured field with the Group parameter = Begin while a data chain already exists
- Receiving a transmission that starts with any 3270 command except a Copy (BSC only), a WSF (followed by a valid Data Chain structured field) or an EW/EWA (with the WCC = Reset).

On inbound (from the device) transmissions, the following are invalid:

- Sending a Data Chain structured field with the Group parameter = Continue or End without having first begun the data chain with a Data Chain structured field with the Group parameter = Begin
- Sending a Data Chain structured field with the Group parameter = Begin while a data chain already exists.

The INCTRL parameter in the Data Chain structured field allows the host application to enable/disable the use of inbound data chaining. The default (for instance, Power-On Reset (POR)) is inbound data chaining disabled. This prevents a host application receiving a structured field it does not recognize. Once enabled, inbound data chaining is allowed until any one of the following occurs:

- Receiving an outbound Data Chain structured field with INCTRL = Inbound Data Chaining Disabled
- Receiving an EW/EWA command with WCC = Reset
- POR
- Clearing a local function (for instance, pressing the Clear key).

An outbound Data Chain structured field with INCTRL = No Change causes no change in the inbound data chaining state. If inbound data chaining is already enabled, an outbound Data Chain structured field with INCTRL = Enable leaves inbound data chaining enabled. If the inbound data chaining is disabled, an outbound Data Chain with INCTRL = Disabled leaves inbound data chaining disabled.

---

## 5.3 Structured Field Self-Defining Parameters

Self-defining parameters are extensions to structured fields that require optional parameters. This parameter format allows for 255 such optional parameters, each containing up to 127 bytes.

Subtopics:

- [5.3.1 Format of Self-Defining Parameters](#)
- [5.3.2 Additional Content Description](#)

---

### 5.3.1 Format of Self-Defining Parameters

All self-defining parameters have the following format:

Bytes	Content Description
0	Parameter length
1	Parameter identifier value
2-n	Parameters

**Note:** All reserved fields must be zeros; in the tables shown in this chapter, the word *Reserved* represents this value.

---

### 5.3.2 Additional Content Description

**Parameter length** is a 1-byte binary value in the range 2 to 127. Values less than 2 are invalid; values greater than 127 are reserved (the high-order bit of the length byte must be zero). The length value includes 1 byte for Parameter Length and 1 byte for the Parameter Identifier Value. There is no default value.

**Parameter identifier value** is a 1-byte binary value that characterizes the information content of this self-defining parameter. It can have any value listed in the outlined value list. There is no default value.

**Parameters** are variable length fields that contain data needed to perform the function specified by the Parameter Identifier Value. This field can be from zero to 125 bytes long.

---

## 5.4 Outbound Structured Fields

The outbound structured fields (application to the device) are defined in the following pages. Structured field type codes not shown are rejected. Unless specifically stated to the contrary, any bits or fields classified as reserved are checked for zero value. Nonzero values are rejected (see Appendix A, "SNA Sense Codes").

---

## 5.5 Activate Partition

The Activate Partition structured field activates a partition.

Subtopics:

- 5.5.1 Function
- 5.5.2 Format

---

## 5.5.1 Function

The Activate Partition structured field functions as follows:

1. If the named partition does not exist or has a window of zero extent, the structured field is rejected. (See Table A-1 in topic A.0.)
2. The named partition is made active.
3. The screen cursor is displayed at the current cursor position of the named partition.
4. Any saved input-inhibit conditions for the partition are raised, and the appropriate indicators are displayed.
5. Input-inhibit conditions are handled as follows:
  - a. System-lock and PWAIT conditions are saved for the active partition. Their indicators are removed.
  - b. All other input-inhibit indicators remain displayed.

---

## 5.5.2 Format

The following table shows the format of the Activate Partition structured field:

Bytes	Content	Content Description
0-1	<b>L</b>	<b>X'0004'</b> Length of this structured field
2	<b>SFID</b>	<b>X'0E'</b> Identifies this structured field as Activate Partition
3	<b>PID</b>	Partition identifier (X'00' through X'7E')

---

## 5.6 Begin/End of File

This section describes the Begin/End of File outbound structured field.

Subtopics:

- 5.6.1 Function
- 5.6.2 Format

---

### 5.6.1 Function

The Begin/End of File structured field indicates to the device that a new file is beginning or that the current file is completed.

---

## 5.6.2 Format

The following table shows the format of the Begin/End of File structured field:

Byte	Bit	Content	Content Description
0 - 1		<b>L</b>	<b>X'0007'</b> Length of this structured field
2 - 3		<b>SFID</b>	<b>X'0F85'</b> Identifies this structured field as Begin/End of File
4		<b>PID</b>	Partition identifier
5		<b>FLAG 1</b>	
	0 - 1	SFID	<b>B'00'</b> Reserved <b>B'01'</b> End of File is being sent <b>B'10'</b> Begin of File is being sent <b>B'11'</b> Reserved
	2 - 7	SFID	These bits are reserved
6		<b>FLAG 2</b>	Reserved

---

## 5.7 Create Partition

This section describes the Create Partition outbound structured field.

Subtopics:

- [5.7.1 Function](#)
- [5.7.2 Format](#)
- [5.7.3 Additional Content Description](#)
- [5.7.4 Define Presentation Space Types Self-Defining Parameter](#)

---

### 5.7.1 Function

The Create Partition structured field creates a new partition.

If the display is in implicit partition state, the Create Partition destroys implicit partition 0 and creates a partition identifier equal to

PID. The display state changes from implicit partition state to explicit partitioned state and the created partition is made active.

If the display is in explicit partitioned state, the Create Partition does the following:

- It destroys the partition with identifier equal to PID, if it exists, and replaces it.
- If a partition does not exist with an identifier equal to PID, it creates a new partition with an identifier equal to PID. The newly created partition is made active if it is the only partition with a window of nonzero extent.

The conditions shown below occur whether the display is in the implicit or explicit partitioned state. Parameter references are to the Create Partition Structured Field Base description which follows the list below:

- The viewport of the new partition is assigned to origin (RV, CV) relative to the usable area, width WV and height HV. If any of the following conditions exist, the structured field is rejected:
  - $WV=0$  and modify partition is not supported.
  - $HV=0$  and modify partition is not supported.
  - $RV+HV >$  useable area height if the Unit of Measure (UOM)=X'0' or X'1' or  $RV+(HV+PH) >$  useable area height (points) if UOM=X'2'
  - $CV+WV >$  useable area width if the UOM=X'0' or X'1' or  $CV+(WV \times PW) >$  useable area width (points) if UOM=X'2'.
  - The viewport overlaps another viewport.
  - The viewport cannot be contained within the usable area.
- The current buffer address for the new partition is set to zero.
- The current cursor position for the new partition is set to the window origin.
- The nonloadable character set is established from the list returned in the character sets Query Reply according to the value of BASE.

Whenever the partition is active, keyboard operation (including the association of each key with its I/O code) is established by the specified Base Character Set.

- The window origin offset is set to RW, CW. If the window is not wholly contained within the presentation space, an error occurs.

---

## 5.7.2 Format

The following table shows the format of the Create Partition structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'0C'</b> Identifies this structured field as Create Partition
3		<b>PID</b>	The identifier for the new partition X'00' through X'7E'
4	0-3	<b>UOM</b>	The unit of measure:  <b>B'0000'</b> Character cells <b>B'0001'</b> Addressable points (H thru PH) <b>B'0010'</b> Addressable points (RV & CV) <b>Other</b> Reserved
	4-7	<b>A-MODE</b>	Address mode:  <b>B'0000'</b> 12/14-bit binary <b>B'0001'</b> 16-bit binary <b>B'0010'</b> Text <b>Other</b> Reserved
5		<b>FLAGS</b>	
	0	<b>RES</b>	Reserved
	1	<b>PROT</b>	<b>B'0'</b> Unprotected partition <b>B'1'</b> Protected partition
	2	<b>COPY</b>	Type of host-initiated local copy:  <b>B'0'</b> Viewport <b>B'1'</b> Presentation space
	3-4	<b>RES</b>	Reserved
	5-7	<b>BASE</b>	Base character set index
6-7		<b>H</b>	The height of the presentation space
8-9		<b>W</b>	The width of the presentation space
10-11		<b>RV</b>	The y, or row, origin of the viewport relative to the top edge of the usable area
12-13		<b>CV</b>	The x, or column, origin of the viewport

			relative to the left side of the usable area
14-15		<b>HV</b>	The height of the viewport
16-17		<b>WV</b>	The width of the viewport
18-19		<b>RW</b>	The y, or row, origin of the window relative to the top edge of the presentation space
20-21		<b>CW</b>	The x, or column, origin of the window relative to the left edge of the presentation space
22-23		<b>RS</b>	The number of units to be scrolled in a vertical multiple scroll
24-25		<b>RES</b>	Reserved
26-27		<b>PW</b>	The number of points in the horizontal direction in a character cell in this presentation space
28-29		<b>PH</b>	The number of points in the vertical direction in a character cell in this presentation space

Working backwards from byte 29, the previously defined parameters can be progressively omitted by specifying the appropriate length of the structured field.

The actual size and position of the partition viewport on the screen is computed using the parameters PW and PH or their defaults.

The following table shows the defaults for omitted parameters:

<b>Parameter</b>	<b>Default Value</b>
PID	0
UOM	B'0000'
A-MODE	B'0000'
Flags	X'00'
H	Default screen height specified in Bind
W	Default screen width specified in Bind
RV	X'0000' (Top screen row)
CV	X'0000' (Left-most screen column)
HV	Smaller of H and height of usable area
WV	Smaller of W and width of usable area
RW	X'0000' (Top presentation space row)
CW	X'0000' (Left-most presentation space column)
RS	X'0001' if H>HV; X'0000' if H=HV
CS	X'0001' if W>WV; X'0000' if W=WV

PW	AW (Usable area cell width)
PH	AH (Usable area cell height)

---

### 5.7.3 Additional Content Description

The content of the Create Partition structured field is further described as follows:

**UOM** Units of Measure. Bits 0-3 (byte 4) refer to the units of measure to be used in subsequent parameters of this structured field, which define the geometrical characteristics of the partition.

**For this**

**UOM**

**value: Parameters are interpreted as follows:**

B'0000' In the row, column coordinate system

B'0001' In the all points addressable system

B'0010' In the row, column coordinate system except for fields RV and CV, which are interpreted in the all points addressable system.

UOM does not affect the type of data or the coordinate system that must be used when writing to the partition.

The ability to support the various UOM options is indicated by the device in the Usable Area and Alphanumeric Partitions Query Reply structured fields as follows:

- If the Cell parameter of the Usable Area Query Reply is B'0', only UOM = B'0000' is supported.
- If the Cell parameter is B'1' and the APA parameter of the Alphanumeric Partition's Query Reply is B'0', UOM = B'0000' or B'0010' is supported.
- If the Cell parameter is B'1' and the APA parameter is B'1', UOM = B'0000', B'0001', or B'0010' is supported.

**A-MODE** If A-MODE is B'0010', a default text environment is established (Adjust off, no Tabs, text width = W, and left margin = 1).

**RS** If RS is X'FFFF', vertical scrolling is device dependent. The device determines the amount of data movement for explicit requests from the operator and for automatic scrolling.

**FLAGS** Flags may be specified as follows:

**PROT** If the PROT flag is specified as B'1', the newly created partition is marked as protected.

**COPY** The COPY flag determines whether the viewport or presentation space are locally copied. If the copy flag is set to B'0', the contents of the character buffers and attribute buffers (if any) that represent the viewport of the active partition are utilized. If the COPY flag is B'1', the contents of the character buffers and attribute buffers (if any) that represent the presentation space of the active partition are utilized. See Chapter 8, "Printer Operations" for details of local copy operation.

**BASE** The BASE parameter is used to select the appropriate base character set for a partition by indexing down the list of base character set descriptors in the order returned in the Character Sets Query Reply.

**Notes:**

1. If any of the errors identified above occur during the processing, the structured field is rejected. (See Table A-1 in topic A.0.)
2. Any explicitly created partitions are destroyed by BIND, UNBIND, ACTLU, DACTLU, ACTPU and DACTPU.
3. If at the end of a transmission, the display is in send or contention state and all partitions have zero extent windows, the structured field is rejected.

---

## 5.7.4 Define Presentation Space Types Self-Defining Parameter

The Define Presentation Space Types self-defining parameter defines the presentation spaces for this partition.

The presentation space types are always ordered Alphanumerics, Image, and Graphics with Alphanumerics as the topmost layer.

**Notes:**

1. If an Image-only presentation space is created, the UOM must be addressable points.
2. If this self-defining parameter is omitted, an Alphanumeric presentation space will be created (the default is X'00').

The following table shows the format of the Define Presentation Space Types self-defining parameter:

---

Byte	Content	Content Description
0	<b>L</b>	Length
1	<b>SDPID</b>	<b>X'05'</b> Identifies this self-defining parameter as Define Presentation Space Types
2	<b>FLAGS</b>	Reserved
3-n	<b>PTYPES</b>	PS types defined  <b>X'00'</b> Alphanumerics <b>X'01'</b> Reserved <b>X'02'</b> Image <b>X'03'</b> Graphics

---

## 5.8 Destroy Partition

This section describes the Destroy Partition outbound structured field.

Subtopics:

- [5.8.1 Function](#)
- [5.8.2 Format](#)

---

### 5.8.1 Function

The Destroy Partition structured field destroys a named partition. If the named partition is the only existing partition, the Destroy Partition structured field creates default implicit partition zero. *Destroy* means the act of doing the following:

1. Removing the PID from further accessibility
2. Prohibiting any further display from the character buffer that had been assigned to the PID until that buffer space is erased or overwritten
3. Clearing or overwriting the displayed viewport associated with the PID.

If the identified partition does not exist, Destroy Partition has no effect.

If the named partition is the only existing partition, Destroy Partition does the following:

1. Destroys the named partition
2. Resets the display to implicit partition state and creates the default implicit partition zero

3. Resets any input-inhibit conditions
4. Resets INOP to Read Modified and INPID to zero.

If the named partition is not the only existing partition, Destroy Partition does the following:

1. Activates the next partition with a window of nonzero extent if the Destroy Partition is active. If all other partitions have zero extent windows, Destroy Partition activates the next partition, that is:
  - a. The cursor is displayed at the current cursor position of that partition if it has a nonzero extent window.  
  
If it has a zero extent window, the cursor is displayed in the right hand end of the operator information area.
  - b. Any saved PWAIT or SYSTEM LOCK conditions for that partition are raised, and the appropriate indicators are displayed.
  - c. If the specified PID = INPID, then INPID is reset to zero, and INOP is reset to Read Modified.
2. Destroys the named partition.

**Note:** If at the end of a transmission, the SLU is in send or contention state and all partitions have zero extent windows, an error situation exists.

## 5.8.2 Format

The following table shows the format of the Destroy Partition structured field:

Byte	Content	Content Description
0-1	L	X'0004' Length of this structured field
2	SFID	X'0D' Identifies this structured field as Destroy Partition
3	PID	The partition identifier X'00' through X'7E'

## 5.9 Erase/Reset

This section describes the Erase/Reset outbound structured field.

Subtopics:

- [5.9.1 Function](#)
- [5.9.2 Format](#)

## 5.9.1 Function

The Erase/Reset structured field resets the device to implicit partition state destroying all existing (implicit or explicit) partitions. The function creates an implicit partition zero with default partition characteristics and of default size  
 if IPZ = B'0' or of alternate size if IPZ = B'1'.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

## 5.9.2 Format

The following table shows the format of the Erase/Reset structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0004' Length of this structured field
2		SFID	X'03' Identifies this structured field as Erase/Reset
3		FLAGS	
	0	IPZ	Implicit partition size:  B'0' Default size B'1' Alternate size
	1-7	RES	Reserved

## 5.10 Load Color Table

This section describes the Load Color Table outbound structured field.

Subtopics:

- [5.10.1 Function](#)
- [5.10.2 Format](#)

### 5.10.1 Function

The Load Color Table structured field defines a loadable color table.

Several loadable color tables can be defined within one Load Color Table structured field. If the same entry is referred to more than once, the last definition is the one that is used. Also, the entire table can be reinitialized with a copy of the non-loadable color table.

---

### 5.10.2 Format

The following table shows the format of the Load Color Table structured field:

Byte	Bit	Content	Content Description
0 - 1		<b>L</b>	Length of this structured field
2 - 3		<b>SFID</b>	<b>X'0F05'</b> Identifies this structured field as Load Color Table
4 - N		<b>COMMAND</b>	Command

**Note:** For information regarding the format and operation of the Command parameter, refer to related graphics documentation.

---

## 5.11 Load Format Storage

This section describes the Load Format Storage outbound structured field.

Subtopics:

- [5.11.1 Function](#)
- [5.11.2 Format](#)
- [5.11.3 Additional Content Description](#)

---

### 5.11.1 Function

The Load Format Storage structured field provides the means of distributing 3270 data stream display formats to the Format Storage Auxiliary Device and requesting reports of Format Storage Auxiliary Device storage utilization. The formats are stored in the control unit and are accessed by the Select Format Group, Present Absolute Format, and Present Relative Format structured fields.

---

### 5.11.2 Format

The following table shows the format of the Load Format Storage structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F24'</b> Identifies this structured field as Load Format Storage
4		<b>FLAGS</b>	
	0-1	RES	Reserved, must be set to B'11'.
	2	FSEL	Format Selection:  <b>B'0'</b> Format is not user selectable. <b>B'1'</b> Format is user selectable.
	3	SS	Screen Size for user selected formats:  <b>B'0'</b> Default screen size. <b>B'1'</b> Alternate Screen Size.
	4-7	RES	Reserved, must be set to zeros.
5		<b>FLAGS</b>	Reserved, must be set to zeros.
6		<b>OPRND</b>	Operand:  <b>X'01'</b> Add <b>X'02'</b> Delete Format <b>X'03'</b> Delete Group <b>X'04'</b> Reset All <b>X'05'</b> Request Summary Status <b>X'06'</b> Request Group Status
7-14		<b>LCLNME</b>	Local name for user selectable formats
15-30		<b>FGNME</b>	Format Group name
31-46		<b>FNME</b>	Format name
47-n		<b>FMTDATA</b>	The format to be stored

With the exception of the FMTDATA field, all the parameters shown must be present. If the OPRND parameter = ADD, then the FMTDATA must also be present. If the OPRND parameter is any other valid value, the FMTDATA field is not present.

The names provided by the LCLNME, FGNME, and FNME parameters are in EBCDIC.

### 5.11.3 Additional Content Description

The content of the Load Format Storage structured field is further described as follows:

- L** These bytes specify the length of any structured field. The length must be explicitly specified and therefore the bytes are not allowed to be sent with a value of zero.
- FSEL** This flag indicates whether or not the user can select the format in addition to host application selection. Host selection is accomplished by the Select Format Group, Present Absolute Format, or Present Relative Format structured fields. An FSEL value of B'1' indicates that both the user and host application selection of the format is allowed.
- SS** This flag indicates whether the format applies to the default or to the alternate screen size. The screen sizes referred to are those associated with the device returning the Format Storage Auxiliary Device Query Reply. This flag applies only when the format is selected by the user.
- OPRND** Indicates the operation the structured field performs. The operands are:
- ADD** The structured field adds a format to format storage if a format of the same name does not already exist in format storage. If a format of the same name already exists, the contents of that format are replaced by the contents of the FMTDATA field.
- If the group directory does not exist, it is created.
- If a local name for user selected formats is specified, it is catalogued in the local names directory.
- Available status is returned if the Add operation is successful. Exception status is returned if there is insufficient storage to add the format or if group name or format name is omitted.
- DELETE FORMAT**  
This deletes the format named in the FNME field.
- If a local name exists for the deleted format, it is removed from the local names directory.
- Available status is returned if the Delete operation is successful or if the specified format is not found. Exception status is returned if the group name or format name is omitted.
- DELETE FORMAT GROUP**  
This deletes the format group named in the FGNME field. The associated group directory is deleted and affected local names are removed from the local names directory.

If no group name is specified, exception status is returned. If the specified group name is not found, available status is returned.

#### **RESET ALL**

This deletes all formats in all format groups and clears the local names directory. Available status is then returned.

#### **REQUEST SUMMARY STATUS**

Indicates a request for the device to report the following status by the Format Status self-defining parameter of the Exception/Status structured field:

- Number of groups assigned
- Total number of formats loaded
- Total number of local formats used
- The amount of format storage spaces available.

#### **REQUEST GROUP STATUS**

Requests the device to report the number of formats currently loaded in the group specified by the FGNME parameter. The device reports the requested information by the Group Status self-defining parameter of the Exception/Status structured field.

If the request is made for a non-existent group name, then the number of formats is set to zero.

#### **LCLNME**

Provides a local name for the format that the user can use for selection. LCLNME (and the later FGNME and FNME) are in the form of alphanumeric characters. For example, the name 'BA-----' would appear as X'C2C1-----'. All the bytes must be in the range of X'40' through X'FE'. The LCLNME parameter is applicable only when the FSEL flag indicates user selection of formats is allowed (FSEL = B'1'). If the FSEL flag indicates user selection of formats is not allowed (FSEL = B'0'), LCLNME must be set to zeros.

If a local name for user names is specified, it is catalogued in the local names directory.

#### **FGNME**

Provides the Format Group name. When OPRND = ADD or DELETE, it indicates the Format Group to which the format is to be added or deleted. When OPRND = Delete Group, it indicates the Format Group to be deleted. This parameter is ignored if OPRND = Reset All.

If the group name is omitted, exception status is returned.

#### **FNME**

Provides the name of the format. When OPRND = ADD, it provides the name assigned to the format being loaded. When OPRND = Delete Format, it indicates the name of the format to be deleted. This parameter is ignored if OPRND = Delete Group or Reset.

If the format name is omitted for Add or Delete, exception status is returned.

**FMTDATA** This is the actual format being added to format storage. It is present only when the OPRND parameter = ADD. It is an error condition if present when OPRND is other than ADD.

The data comprising the format is exactly the same as would be sent to a 3270 data stream display if the format were to be displayed immediately rather than stored for future use. A Load Format Storage structured field can contain only one format.

---

## 5.12 Load Line Type

This section describes the Load Line Type outbound structured field.

Subtopics:

- [5.12.1 Function](#)
- [5.12.2 Format](#)

---

### 5.12.1 Function

The Load Line Type structured field specifies the controlling environment defined line types.

The controlling environment defined line types are identified by code points greater than X'40' in the Set Line Type, and Push and Set Line Type orders. The maximum number of different controlling environment defined line types is 190. The controlling environment defined line types are specified in self-defining parameters, which specify the format of the line type. Several line type definitions can be provided within a single structured field.

---

### 5.12.2 Format

The following table shows the format of the Load Line Type structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F07'</b> Identifies this structured field as Load Line Type
4-n		<b>COMMAND</b>	Command

**Note:** For the definition of the format and operation of the Command parameter, refer to the appropriate graphics or image product publications.

---

## 5.13 Load Programmed Symbols (Load PS)

This section describes the Load Programmed Symbols outbound structured field.

Subtopics:

- [5.13.1 Function](#)
- [5.13.2 Format](#)
- [5.13.3 Additional Content Description](#)
- [5.13.4 Symbol Envelope Table Self-Defining Parameter](#)
- [5.13.5 Terminator Self-Defining Parameter](#)
- [5.13.6 Compression Function](#)
- [5.13.7 Character Cell Division](#)
- [5.13.8 Compression Process](#)
- [5.13.9 Comparison Rules and Header Bits](#)
- [5.13.10 Creating the Compressed Bit String](#)
- [5.13.11 Terminator Bits](#)
- [5.13.12 Examples of the Compression Algorithm in Use](#)

---

### 5.13.1 Function

The Load Programmed Symbols structured field loads symbol and character definition information into the device.

This structured field causes characters to be loaded into contiguously addressable slots in program storage. The storage area in the device is called Read/Write storage (RWS) and, when supported, is identified in the device's Character Set Query Reply. This storage is made up of 191 contiguously addressable slots, associated with positions (X'40' to X'FE') in the data stream. Remember that the slot associated with X'40' cannot be loaded and contains a blank.

If this structured field does not contain data, the operation executes anyway. It executes as the parameters specify and no data is loaded. The characteristics of a character set can be changed without altering the contents of the character set. For example, a character set previously defined as requiring skip suppression can have this parameter changed to require no skip suppression.

---

### 5.13.2 Format

The following table shows the format of the Load Programmed Symbols structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'06'</b> Identifies this structured field as Load Programmed Symbols (Load PS)

3		FLAGS	
	0	Basic/Ex	Basic or extended form:  <b>B'0'</b> Basic form: only bytes 0 through 6 present <b>B'1'</b> Extended form: length determined by byte 7
	1	CLEAR	Clear loadable character set:  <b>B'0'</b> Do not clear loadable character set. <b>B'1'</b> Clear all character slots not loaded.
	2	SKIP	Skip suppress (Note 1):  <b>B'0'</b> Suppression not required <b>B'1'</b> Suppression required
	3-7	TYPE	Data format type:  <b>B'00000'</b> Reserved  <b>B'00001' (Type 1)</b> 18-byte form; the first 2 bytes contain a 16-bit vertical slice, the following 16 bytes contain 8-bit horizontal slices. For a 9 x 12 character matrix the last 4 bytes contain binary zero.  <b>B'00010' (Type 2)</b> Type 1 compressed  <b>B'00011' (Type 3)</b> Row loading (from top to bottom)  <b>B'00100' (Type 4)</b> Type 3 compressed  <b>B'00101' (Type 5)</b> Column loading (from left to right)  <b>B'00110' (Type 6)</b> Type 5 compressed  <b>B'01000' (Type 8)</b> Vector  <b>Others</b> Reserved

4		<b>LCID</b>	Local character set ID: Use X'40' to X'EF'. Use X'FF' to indicate that RWS associated with this LCID is free (not assigned). Other values are reserved.
5		<b>CHAR</b>	Beginning code point X'41' through X'FE'
6		<b>RWS</b>	Loadable Character Set RWS Number

Bytes 7 to 12 are parameters for the extended form of Load PS.

Byte	Bit	Content	Content Description
7		<b>P Length</b>	Length of parameters for extended form, including the length parameter. The parameters defined below can be progressively included by specifying the appropriate length. Omitted parameters are assumed to have the defined default indicated by the zero value for that parameter.
8		<b>FLAGS</b>	
	0	APA	All points addressable (APA):  <b>B'0'</b> APA <b>B'1'</b> Not APA
	1	CB	LCID compare bit:  <b>B'0'</b> LCID compare <b>B'1'</b> No LCID compare
	2	OB	Operator selectable by PS key:  <b>B'0'</b> Operator selectable <b>B'1'</b> Not operator selectable
	3	MULTID	Multiple LCID:  <b>B'0'</b> Release any other symbol set with the same LCID. <b>B'1'</b> Release any other symbol set with the same LCID unless it was loaded with MULTID=B'1'.
	4	USET	Use Symbol Envelope Table.  <b>B'0'</b> Do not use S.E.T. information. <b>B'1'</b> Use S.E.T. information.
	5-6	RES	Reserved

	7	SDPP	Self-Defining Parameters Present:
			<b>B'0'</b> No self-defining parameters present <b>B'1'</b> Self-defining parameters follow
9		LW	Number of X-units in character cell (width of character matrixes)
10		LH	Number of Y-units in character cell (depth of character matrixes)
11		SUBSN	Subsection ID:
			<b>X'00'</b> One-byte codes <b>X'41'- X'FE'</b> Subsection ID for 2-byte coded data <b>Other</b> Reserved
12		COLOR	Color planes:
			<b>X'00'</b> All planes <b>X'01'</b> Blue plane <b>X'02'</b> Red plane <b>X'04'</b> Green plane <b>X'nn'</b> Color attribute value
13		ST. SUBS	Starting Subsection Identifier:
			<b>X'00' (default)</b> Starting Subsection defined by QUERY REPLY--CHARACTER SETS. <b>X'41'- X'FE'</b> Starting Subsection Identifier for 2-byte coded character set specified by LCID in this Load PS. <b>Others</b> Reserved
14		ECHAR	Ending code point
15		NW	Number of width pairs
16		NH	Number of height pairs
17		RES	Reserved - must be zero

**Notes:**

1. Values for parameters other than those shown above are rejected; invalid parameter values are also rejected. (See Table A-1 in topic A.0.)
2. The lost space resulting from setting of the skip suppress to B'1' is made up following the last display/print line; that is, the last display/print line moves up from the bottom of the physical presentation space. For a printer, setting of skip suppress to B'1' must not result in loss of forms sync following a form feed (FF) or forms overflow.

The PS data follows in the format described in bits 3-7 of byte 3.

Byte	Bit	Content	Content Description
M to N		DATA	Character matrices

---

### 5.13.3 Additional Content Description

The content of the Load Programmed Symbols structured field is further described as follows:

**BASIC/EXT** The extended-form bit in byte 3 indicates whether the structured field is of the extended form. The extended form contains additional information associated with copy operations, character matrix size, and color. If the device does not support the extended form and this bit is set to 1, the data stream is rejected with a sense code of X'1005' or X'1001'.

**SKIP** If skip suppression is on, any row in the usable area containing characters from this character set has the skip suppressed that would normally follow that row. The characters on the following row are then vertically adjacent to characters on the current row. If the skip suppression flag is not set, no skip suppression is required for characters from this character set

**CLEAR** If the Clear flag is set on, all slots in the specified RWS are cleared. The character definitions in the data portion of the structured field, interpreted according to the data format type (byte 3), are then loaded into contiguously addressable slots in the PS RWS, starting at the position defined by Char.

If the Clear flag is set but no character matrix data is included in the structured field, the addressed character set is cleared. Subsequent data stream references to this character set produce blank characters. For a dynamically allocated symbol store, this means that the storage is freed. For a preallocated symbol store, this means that every symbol slot is set to binary zero.

**TYPE**

If the data format type specifies *compressed* data (type 2, 4, or 6), the data is first decompressed. (See "Compression Function" in topic 5.13.6.)

For data of type 1 or type 2 (decompressed), each set of 18 contiguous bytes defines a character in a 9-by-16 character cell.

For data of type 3 (decompressed), each set of LW contiguous bits defines one row of a character matrix. The number of LH contiguous rows defines a character as a LW-by-LH matrix of dots, where LW is the width of the character matrix and LH is the height of the character matrix. If LW and LH are not supplied in the extended form of the Load PS structured field, or if they are supplied and set to zero, the values of LW and LH are determined by the device. Each character definition is extended with bits that can be ignored to a multiple of 8 bits. (In other words, the next definition commences on a byte boundary.)

For data of type 5 (decompressed), each set of LH contiguous bits defines one column of a character cell. The number of LW contiguous columns defines a character as an LW-by-LH matrix of dots. The bits are extended in the same fashion described for type 3 data.

Loading of character definitions continues until either of the following is true:

- The data is exhausted. In this case the last complete cell definition in the data is loaded. If there are any excess bits, they are ignored and -RSP (with a sense code of X'1001' or X'1005') is returned.
- The slot corresponding to X'FE' is loaded. Excess data is ignored and -RSP (with a sense code of X'1001' or X'1005') is returned.

**LCID**

The LCID specified in this structured field is released from any previously associated PS RWS number if the MULTID is equal to B'0' in this structured field, or if the MULTID is equal to B'1' in this structured field and was equal to B'0' in the structured field that previously loaded that LCID. There can be several PS sets, all with the same LCID co-residing in different slot sizes or with different types. Each can be loaded with MULTID equal to B'0'. The PS set RWS number released is assigned an LCID = X'FF'; this is not done until the LCID specified in this structured field is assigned to the associated RWS.

The PS RWS number specified in this structured field is released from any previously assigned LCID. Any subsequent reference to this released LCID causes an error condition. The LCID and PS RWS numbers specified in this structured field are associated for any subsequent data stream processing.

An LCID of X'FF' indicates that this PS RWS is free (not assigned). The operator cannot select a PS RWS with an LCID = X'FF'.

On a Load PS, when an error condition is detected before the contents of the associated RWS are altered, an error sense code is sent, but the LCID is not changed. Where the error condition occurs during the update of the RWS, the error condition is sent, and the LCID is updated to equal the one specified in Load PS. This applies to a sequence of Load PSs sent after a WSF. Also, any sequence of Load PSs following the failing Load PS is not executed.

**RWS** Byte 6 indicates the physical RWS to be loaded. There is a fixed relationship between the physical RWS number and the attribute selection keys defined for PS. The value in CHAR indicates the first slot to be loaded and must be in the range X'41' through X'FE'.  
If CHAR is outside this range, the data stream is rejected (invalid parameter, sense code X'1005' or X'1001').

**CHAR** The content of this byte specifies the code point at which loading of symbol definitions or symbol envelope table values begins. The code point is incremented by one for each pair of a symbol definition and a symbol envelope table parameter (if both are provided in this structured field). Otherwise the code point is incremented by one for each symbol definition or symbol envelope table parameter (if only one category of information is provided in this structured field). Code points whose symbol definitions or symbol envelope table values are not updated by this command are unaltered (subject to the Clear parameter).

Any particular Load PS Set command can have more or less symbol envelope table values than symbol definitions. Only values or definitions explicitly provided cause any update to the symbol envelope table values or to the symbol definitions. In other words, symbol definitions without corresponding symbol envelope table values update only the symbol definition and leave any existing symbol envelope table values unaltered and vice versa. On invocation, any code point that does not have symbol envelope table values or a symbol definition is treated as a space or as having symbol envelope table values of all zeros.

The above rule permits the symbol envelope table values and the symbol definitions to be loaded independently (before or after).

**APA** When set to B'1' (not APA), this bit implies that fewer than all points can be printed to allow a performance gain for specific devices. For example, 3287 not APA attempts to print all characters in one head sweep across the print line.

**CB** When this bit is set to B'0' (Compare), the LCID of this character set is compared with character set LCIDs in the printer to determine whether there is a match. If the LCIDs match and the CB bits are both zero, the copy operation is performed using the corresponding LCID in the printer. If not, characters from the nonloadable character set of the printer are used.

When set to 1, the LCID of this character set is not compared with LCIDs of character sets in the printer. The copy operation is performed using the base character set of the printer.

**OB**

When set to B'1', it signifies that this character set is intended for output only. Thus, the PS key normally associated with the RWS containing this character set cannot be enabled by the terminal operator while the key is connected to this LCID. Selection of individual character sets can, therefore, be disabled, even though the Set Reply Mode structured field allows character set selection. When the Set Reply Mode structured field disables character set selection, selection is disabled for all character sets independent of the setting of the OB bit for each character set.

The SDPP indicator specifies whether or not there are self-defining parameters provided prior to the symbol definitions in this structured field.

**LW/LH**

If LW and LH for the character matrix dimensions are specified and are nonzero, then the character matrixes defined by LW and LH are loaded into the specified PS RWS. (LW must be not more than the character slot width of the referenced character set. LH must be not more than the character slot height of the referenced character set.) They are loaded so that the first bit aligns with the upper left-hand corner of the character slot.

If either LW or LH exceeds the size of the character slot for the referenced character set, this structured field is rejected.

If LW and LH are not specified or are set to zeros, the device assumes that the size of the character matrices transmitted in the Load PS structured field is the same as the device default character slot size.

If data of types 1 or 2 has been specified, then only the values LW = 9 and LH = 16 are valid. Any other values are rejected.

**COLOR**

A character set with triple-plane capability has three color planes into which characters can be loaded. For any code point (X'41' to X'FE') within the character set, each plane can be loaded independently. That is, a different character definition can be loaded into each of the primary planes in the character set for that code point.

For a triple-plane character set, if the Color field is B'001', B'010', or B'100', the character set data is loaded only into the specified character slots in that plane. Other color values are reserved and rejected.

For a triple-plane character set, if the Color field bits are omitted or if the Color field is B'000', then the character set data is loaded into the specified character slots in all three planes.

Additionally, a symbol can be constructed using the color attribute values supported by the device as reported in the Color Query Reply. Since a device must support a power-of-two number of colors, the color attribute values for the complete

character slot resolves into color constituent parts (known as the primaries) for that code point. If some further load for that code point uses a different color attribute value then that too resolves into color constituent parts. This can destroy a previous loading of that color constituent part. When that symbol is invoked, the color constituent parts are ORed for each pel of the character slot (to reconstitute the color attribute value of that pel).

For a single-plane character set, if the Color field is B'001', B'010' or B'000', the data stream is rejected.

**ECHAR** This is the EBCDIC code point of the last symbol definition or Symbol Envelope Table entry contained in this Load PS structured field. If there are fewer symbol definitions or Symbol Envelope Table entries provided than "ECHAR - CHAR + 1," the symbol definitions and the Symbol Envelope Table entries of the unspecified code points are unaffected. If there are more symbol definitions or Symbol Envelope Table entries provided than "ECHAR - CHAR + 1," the structured field is rejected and the condition of the store remains as it was prior to this Load PS structured field. The condition, however, is subject to the Clear parameter and with the symbol definitions or Symbol Envelope Table entries in the range CHAR to ECHAR (inclusive) updated. This parameter is relevant only to symbol sets specifying that the ending code point must be included in any Load PS structured field addressed to them.

**NW** This is the number of pairs of width indentation for this symbol set. Besides setting this number for the whole symbol set, this parameter also prescribes the format of the Symbol Envelope Table self-defining parameter.

**NH** This is the number of pairs of height indentation for this symbol set. Besides setting this number for the whole symbol set, this parameter also prescribes the format of the Symbol Envelope Table self-defining parameter.

---

### 5.13.4 Symbol Envelope Table Self-Defining Parameter

The Symbol Envelope Table self-defining parameter provides the information pertinent to a single symbol. Therefore, there can be as many of these self-defining parameters as necessary. The first of these is the information for the symbol identified by the Char parameter. Every symbol of the set has the same numbers of indentation pairs. If this command creates a Symbol Envelope Table for this symbol store, then any symbols not assigned explicit values assume all zero values. Subsequent Load PS commands only alter the pair values (subject to the Clear parameter) for code points for which values are explicitly provided.

The following table shows the format of the Symbol Envelope Table self-defining parameter:

Byte	Content	Content Description
0	L	Length of parameter
1	SDPID	X'01' Identifies this self-defining parameter as Symbol Envelope Table

2 to 1 + 2NW	<b>WPAIRS</b>	Pairs of width indentation values.
2+2NW to 2NW +2NH+1	<b>HPAIRS</b>	Pairs of height indentation values.

The content of the Symbol Envelope Table self-defining parameter is further described as follows:

**WPAIRS** A list of pairs of 1-byte indentation values. Each pair is an indentation value from the left and right edges of the definition cell respectively and is expressed in the same units as the definition cell. There can be as many pairs of values as necessary to correctly envelop the symbol. If there is only one pair, then the values are assumed for the entire cell. Otherwise, the first pair of values applies at the bottom of the cell, and the last pair of values applies at the top of the cell. The sum of the values of any pair must not exceed the width of the definition cell.

**HPAIRS** A list of pairs of 1-byte indentation values. Each pair is an indentation value from the bottom and top edges of the definition cell respectively and is expressed in the same units as the definition cell. There can be as many pairs of values as necessary to correctly envelop the symbol. If there is only one pair, its values are assumed for the entire cell. Otherwise, the first pair of values applies at the left of the cell, and the last pair of values applies at the right of the cell. The sum of the values of any pair must not exceed the height of the definition cell.

---

### 5.13.5 Terminator Self-Defining Parameter

The Terminator self-defining parameter indicates the cessation of self-defining parameters. Any data following in this structured field is interpreted as symbol definitions.

The following table shows the format of the Terminator self-defining parameter:

Byte	Content	Content Description
X to Y	<b>SDPID</b>	<b>X'02FF'</b> Terminates self-defining parameters

---

### 5.13.6 Compression Function

Symbol definition bit strings can be transmitted by the Load PS structured field function in uncompressed or compressed form. When the Type = 2, 4, or 6, the compression function proceeds in four parts:

- The cell is divided into appropriate pieces called *slices* and the slices are further subdivided into 4-bit *elements*.
- The elements are compared to determine pattern matching.
- The elements are grouped into *bunches*.
- The element comparisons are encoded as *compressed character matrixes*.

An uncompressed symbol definition requires either 18 bytes of data (display) or 10 bytes of data (printer) to be transmitted. *Compression*, as described here, is a method for reducing the number of bytes transmitted.

An uncompressed symbol definition is created by dividing the character cell within which a symbol is formed into bytes (slices), as shown under "Character Cell Division" in topic 5.13.7. The symbol is defined by encoding the bits (dots) in each byte (slice) as B'1' if the dot is to be *on* or B'0' if the dot is to be *off*. The dot pattern representing the symbol is thus formed.

Byte (slice) 1 is understood to represent the leftmost upper vertical 8 dots in the display matrix or the leftmost vertical 8 dots in the printer matrix. The string of 144 bits (display) or 80 bits (printer) thus encoded represents the uncompressed symbol definition. A comparison process, comparing elements (four bits) in the uncompressed bit string with reference elements selected from the same bit string, is used to compress the data.

## 5.13.7 Character Cell Division

The character cell for a display or printer character position is divided into slices as shown in Figure 5-2 in topic 5.13.8 and in Figure 5-3 in topic 5.13.8. A slice corresponds to a byte, and the bits to the pels of the character cell.

In the case of the packed vertical slice, type 6, the slices correspond to columns of bits from the character matrix. Thus, the slice length is the same as the character matrix height (LH), while the number of slices corresponds to the matrix width (LW). For example, the leftmost column of the character matrix image corresponds to slice 1.

The packed horizontal slice, type 4, is similar though rotated 90 degrees. The slices correspond to the rows of the character matrix and the slice length is the same as the matrix width (LW), while the number of slices is the width height (LH). The topmost row becomes the first slice, as shown in Figure 5-2.

Once the character cell has been sliced in an appropriate manner, the slices can be thought of as forming a data string, beginning with slice 1, the 0 bit in each slice at the left, as follows:

Slice 1	Slice 2	0 1 2 3 4 5 6 7	....	Slice 10 or 18
---------	---------	-----------------	------	----------------

Each group of four bits is termed an *element*. Each slice is extended as necessary with zero bits so that the slice length is a multiple of four bits. The bit string forming the symbol definition is compressed by comparing each element to the corresponding element in another slice (or 0), and encoding the successful comparisons or matches. The compressed bit string is generated according to the matches and mismatches that occur in the comparison process.

### 5.13.8 Compression Process

In creating a type 2 (display) or type 6 (printer) compressed bit string for an individual symbol, an algorithm based on one of four comparison rules is used. A header (of one to four bits) is used at the start of each symbol definition to signal which of the four comparison rules was used in the compression.

PICTURE 9

Figure 5-2. Vertical and Horizontal Slicing of a Character Cell

The compressed bit strings for all the symbols being defined are concatenated without regard for the byte boundaries, and terminator bits are added to make the total bit string fit into an integral number of bytes. Type 2 or Type 6 data defining a full set of symbols in a Load PS structured field function is as follows:

H	SD	H	SD	H	SD		H	SD	H	SD	T
Symbol 1	Symbol 2	Symbol 3	....	Symbol 189	Symbol 190						

H Header bits  
SD Symbol definition bits  
T Terminator bits.

"Comparison Rules and Header Bits" in topic 5.13.9 describes the comparison rules and header bits, the creation of the compressed bit string, the terminator bits, and provides examples of compressing the symbol definition bit strings for the symbol in Figure 5-3.

PICTURE 10

Figure 5-3. Type 1 Data Format -- An Example Dot Pattern Encoded

### 5.13.9 Comparison Rules and Header Bits

The four comparison rules that follow are used in creating a compressed

symbol definition bit string from a Type 1 (display) or Type 5 (printer) uncompressed symbol definition bit string. Encoding the results of the comparisons is discussed in "Creating the Compressed Bit String" in topic 5.13.10.

- Comparison Rule 1 (Header bit = B'0')

Each element is compared with an element consisting of all 0 bits.

- Comparison Rule 2 (Header bits = B'1')

Each element is compared with the corresponding element in the previous slice; for example, the first element of slice 2 is compared with the first element of slice 1, the second element of slice 2 with the second element of slice 1, the first element of slice 3 with the first element of slice 2, and so on. Since slice 1 has no previous slice, each element of slice 1 is compared with a 0 element.

- Comparison Rule 3 (Header bits = B'110')

Each element is compared with the corresponding element in the next-to-previous slice; for example, the first element of slice 3 is compared with the first element of slice 1, the second element of slice 3 with the second element of slice 1, the first element of slice 4 with the first element of slice 2, and so on. Since slices 1 and 2 have no next-to-previous slice, each element of slices 1 and 2 is compared with a 1 element.

- Comparison Rule 4 (Header bits = B'1110')

The symbol definition consisted of 0 bits only and no comparison is required. It is a blank symbol.

If all of the header bits contain anything other than the bit combinations listed here, the data stream is rejected.

---

## 5.13.10 Creating the Compressed Bit String

The element comparisons are encoded by taking the elements four at a time (two slices -- the slice pair referred to in the following discussions) and comparing them with their corresponding elements in a reference slice pair that is created (following the comparison rules). Because the elements are compared four at a time, it is convenient to regard the 18 slices of the Type 1 data string or the 10 slices of a Type 5 data string symbol definition as being made up of 9 or 5 slice pairs. The elements of the slice pairs are compressed as follows:

1. Compare the first Type 1 or 5 slice pair with the reference slice pair. If the two slice pairs are identical, put a 0 bit in the symbol definition bit string and repeat step 1 for the next slice pair. If the two slice pairs are not identical, put a 1 bit in the symbol definition bit string and continue with step 2.
2. Compare, in turn, each element in the Type 1 or 5 slice pair with the corresponding element in the reference slice pair. For each element

that matches (that is, if the elements being compared are the same), put a 0 bit in the symbol definition bit string. For each element that does not match (that is, if the elements being compared are not the same), put a 1 bit in the symbol definition bit string followed by a copy of the four bits of the non-matching element from the Type 1 or 5 slice pair.

3. Repeat steps 1 and 2 through the ninth slice pair of a Type 1 string or the fifth slice pair of a Type 5 string.

---

### 5.13.11 Terminator Bits

When the bit strings for all symbols have been created and concatenated, the Type 2 or Type 6 data string is completed with 1 bits to make up an integral number of bytes. There must be at least four of these terminator 1 bits, even if they spill over into a further byte. The number of 1 bits required thus ranges from 4 (minimum) to 11 (maximum).

---

### 5.13.12 Examples of the Compression Algorithm in Use

The following three examples show how a Type 1 data string for a particular symbol is compressed into a Type 2 data string. In these examples, the symbol whose Type 1 data string is being compressed is the one shown in [Figure 5-3](#) in [topic 5.13.8](#).

The data string for that symbol presented as nine slice pairs is as follows:

```
Slice pair 1: X'1010'  
Slice pair 2: X'0000'  
Slice pair 3: X'0000'  
Slice pair 4: X'8040'  
Slice pair 5: X'2010'  
Slice pair 6: X'0804'  
Slice pair 7: X'12FF'  
Slice pair 8: X'0000'  
Slice pair 9: X'0000'.
```

For the particular symbol used in these examples, comparison rule 1 yields the shortest bit string; for any other symbol, however, the comparison rule that yields the shortest bit string depends on the symbol's particular dot pattern.

*Using Comparison Rule 1*

**PICTURE 11**

Figure 5-4. Example of Compression Algorithm Using Comparison Rule 1

With comparison rule 1, the header is B'0' and the symbol definition bit

string is created by comparing each Type 1 slice pair with an all-zeros reference slice pair as shown in Figure 5-4. The resulting bit string, including the header, is as follows:

```
0110 0010 1000 1000 1110 0001 0100 0110 0100 1000 1010 1100
0010 1001 0100 1011 1111 1111 00
```

The original Type 1 bit string of 144 bits is compressed into 74 bits.

*Using Comparison Rule 2*

 PICTURE 12

Figure 5-5. Example of Compression Algorithm Using Comparison Rule 2

With comparison rule 2, the header is B'10' and the symbol definition bit string is created by comparing each Type 1 slice pair with a reference slice pair composed of the previous slices, as shown in Figure 5-5. The resulting bit string including the header is as follows:

```
1011 0001 0001 1000 0000 0111 0000 1010 0011 0010 0100 0101
1000 0110 0001 0100 1010 0101 1111 1111 1110 0001 0000 000
```

The original Type 1 bit string of 144 bits is compressed to 95 bits.

*Using Comparison Rule 3*

 PICTURE 13

Figure 5-6. Example of Compression Algorithm Using Comparison Rule 3

With comparison rule 3, the header is B'110' and the symbol definition bit string is created by comparing each Type 1 slice pair with a reference slice pair composed of the next-to-previous slices, as shown in Figure 5-6. The resulting bit string, including the header, is as follows:

```
1101 1000 1010 0010 1100 0001 0000 0011 1000 0101 0001 1001
0010 0010 1100 0011 0001 0000 1010 0101 0010 1111 1111 1110
1000 0100 0010 0000
```

The original Type 1 bit string of 144 bits has been compressed to 112 bits.

---

## 5.14 Modify Partition

This section describes the Modify Partition outbound structured field.

Subtopics:

- 5.14.1 Function
- 5.14.2 Format

---

## 5.14.1 Function

The Modify Partition structured field varies the partition parameters such as protection, viewport position, or window origin.

The structured field is checked for validity. If any of the following conditions exists, the structured field is rejected:

- The named partition does not exist.
- A reserved bit in a flag byte is not zero.
- The viewport overlaps another viewport.
- The viewport cannot be contained within the usable area.
- The window is not completely contained in the presentation space.
- Bytes 8-11 are not X'00000000'.

Working backward from byte 31 through byte 6, parameters can be omitted progressively by specifying the appropriate length. If a parameter is omitted in this way, its value is not altered. The specified values replace those currently in effect.

If the viewport size is changed, the window size is changed accordingly. If as a result, the cursor is outside the window, the window is moved to contain it. Vertically, the window is moved by the minimum number of rows. Horizontally, the window moves so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor is at one of the two center positions.)

If the window origin changes and the cursor is outside the new window, the cursor moves by the minimum number of rows or columns so that it is within the new window.

If the size of the window is varied to zero and this partition is Active, the next partition (if any) with a nonzero window is activated. If every partition has a zero extent window, this partition remains active but the operator cannot type data into it, and the screen cursor is placed at the right-hand end of the indicator row.

If, at the end of a transmission, the display is in send or contention state and the active partition has a zero extent window, an error situation exists.

If this is the only partition with a window of nonzero extent, then it is Activated.

Some reserved fields in Modify Partition correspond to parameters in

Create Partition. These partition parameters (for example, A-MODE) cannot be modified.

## 5.14.2 Format

The following table shows the format of the Modify Partition structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F0A'</b> Identifies this structured field as Modify Partition
4		<b>RES</b>	Reserved
5		<b>PID</b>	Partition identifier
6		<b>RES</b>	Reserved
7		<b>FLAGS</b>	
	0	RES	Reserved
	1	PROT	<b>B'0'</b> Unprotected partition <b>B'1'</b> Protected partition
	2-7	RES	Reserved
8-9		<b>RES</b>	Reserved
10-11		<b>RES</b>	Reserved
12-13		<b>RV</b>	Row origin of viewport
14-15		<b>CV</b>	Column origin of viewport
16-17		<b>HV</b>	Height of viewport
18-19		<b>WV</b>	Width of viewport
20-21		<b>RW</b>	Row offset of window origin
22-23		<b>CW</b>	Column offset of window origin
24-25		<b>RS</b>	Vertical scroll units
26-27		<b>RES</b>	Reserved
28-29		<b>PW</b>	Number of horizontal points in a character cell
30-31		<b>PH</b>	Number of vertical points in a character cell

Each parameter is described under "Create Partition" in topic 5.7.

---

## 5.15 Outbound Text Header

This section describes the Outbound Text Header structured field.

Subtopics:

- [5.15.1 Function](#)
- [5.15.2 Format](#)
- [5.15.3 Additional Content Description](#)

---

### 5.15.1 Function

The Outbound Text Header structured field establishes initial conditions for a text partition.

The structured field is checked for validity. If any of the following conditions occurs, the structured field is rejected:

- The partition A-MODE is not X'2'.
- The named partition does not exist.
- The OP-Type does not specify Write or EW.
- Reserved flag bits are not B'0'.
- LHDR < 2.
- The HDR contains any graphics or unsupported controls.
- The HDR contains inconsistent formatting controls (for example, LM > width).
- A parameter error occurs in any of the HDR controls.
- CC > W-1.

---

### 5.15.2 Format

The following table shows the format of the Outbound Text Header structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F71'</b> Identifies this structured field as Outbound Text Header
4		<b>PID</b>	Partition identifier
5		<b>OP-TYPE</b>	Operation type:
			<b>X'F1'</b> Write
			<b>X'F5'</b> EW

6		<b>WCC</b>	
	0-4	RES	Reserved
	5	SFID	<b>B'1'</b> Sound alarm
	6	SFID	<b>B'1'</b> Keyboard restore
	7	SFID	<b>B'1'</b> Reset MDT
7		<b>RES</b>	Reserved
8		<b>RES</b>	Reserved
9		<b>LVL</b>	Cursor level
10-11		<b>CRO</b>	Cursor row offset
12-13		<b>CC</b>	Cursor column offset
14-15		<b>LHDR</b>	Header length includes itself (minimum value is 2)
16-n		<b>HDR</b>	Initial format controls

### 5.15.3 Additional Content Description

The content of the Outbound Text Header structured field is further described as follows:

**OP-TYPE** If EW is specified, the following occurs:

- The current buffer address is reset to zero, so subsequent text data is loaded at the start of the buffer.
- The text buffer is cleared.
- The MDT is reset.
- The window origin (row,column) is reset to (1,1).
- If Write is specified, the current buffer address is unchanged, so subsequent text data is appended to existing data.

**WCC** If the WCC specifies reset MDT, the MDT is reset.

**LVL** If LVL is X'FF', the LVL value is ignored. Otherwise, LVL

identifies the element at which the cursor is to be placed (the first element in any position is at level 1).

If LVL is X'00', level 1 is assumed.

If there is no element at the specified level, the highest level element at that (row, column) position is assumed.

**CRO** If CRO is X'FFFF', the cursor row is not changed. Otherwise, the cursor is placed in the specified row (CRO+1), of the presentation space.

**CC** If CC is X'FFFF', the cursor column is not changed. Otherwise, the cursor is placed in the specified column (CC+1), of the presentation space.

**HDR** If the HDR data is present, this is processed; otherwise, the existing environment is unchanged.

The HDR data is not stored in the text buffer and is not transmitted inbound.

A subsequent Type 1 Text structured field can contain an insert cursor control. This then overrides the cursor position defined by Outbound Text Header.

Formatting of the text buffer is started at the end of the next Outbound Text structured field or at the end of the RU chain, whichever occurs first. When formatting is complete, the cursor position is calculated and the WCC Keyboard Restore and Alarm bits take effect.

If the cursor is outside the window, the window moves to contain it. Vertically, the window moves by the minimum number of rows. Horizontally, the window is moved so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor is at one of the two center positions.)

---

## 5.16 Outbound 3270DS

This section describes the Outbound 3270DS structured field.

Subtopics:

- 5.16.1 Function
- 5.16.2 Format
- 5.16.3 Additional Content Description

---

### 5.16.1 Function

The Outbound 3270DS structured field directs 3270 data stream orders and

data to a named partition.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

### 5.16.2 Format

The following table shows the format of the Outbound 3270DS structured field: rules=horiz split=yes scale='.9'. rules=none split=yes scale='.9' arrange='1 2 3 / 4 5 6' expand mindepth=p5.

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'40' Identifies this structured field as Outbound 3270DS
3	PID	Partition identifier (X'00' through X'7E')
4	CMND	Partition command:
		X'F1' Write - The associated data is written to the buffers of the specified partition.
		X'F5' EW - The viewport and buffers of the specified partition are cleared, and the data is written to the buffers as in a Write.
		X'7E' EWA - Same action as EW.
		X'6F' Erase All Unprotected - All the unprotected fields in the named partition are set to nulls.
		X'F7' BSC Copy
5	WCC CCC	3270 WCC byte for X'F1', X'F5', and X'7E' CCC for BSC copy (X'F7')
6-7	DATA FROM ADDRESS	3270 Orders and data for X'F1', X'F5', and X'7E' BSC Copy (X'F7')
8-n	DATA	3270 Orders and data for X'F1', X'F5', and X'7E'

**Note:** EW and EWA do not change the screen size. Use the Erase/Reset structured field to do this.

### 5.16.3 Additional Content Description

The content of the Outbound 3270DS structured field is further described as follows:

**PID** Specifies to which partition the data in this structured field

is to be directed. The specified partition must be defined or a data stream error occurs. (See Table A-1 in topic A.0.) On a device that does not support multiple partitions, partition 0 must always be specified.

**CMND** Contains control information for the partition. When CMND = X'6F', no WCC appears in the structured field. CMND values not shown are reserved and are rejected (sense code X'1003' or X'1001').

If the CMND byte specifies BSC Copy, the copy is valid only if the following are true:

- The *from* device and the *to* device are in implicit partition state.
- The communications must be remote BSC.
- The Outbound 3270DS structured field carrying the BSC Copy command must be the last structured field in the transmission.

An error sense code is returned if these conditions are not met. See Table A-1 in topic A.0 for other error conditions and the sense codes that are returned.

**WCC** Contains the write control character as defined for the 3270 data stream in "Write Commands." When CMND = X'F1', bits 0 to 3 of the WCC byte are ignored for displays, and bits 0 and 1 are ignored for printers. If PID is not equal to 0, only the indicated partition is reset.

If no WCC byte is defined following an EW or EWA command, then no erasing or resetting occurs. The structured field is treated as a negative reply to a trigger, but it does not acknowledge any outstanding read or entry.

---

## 5.17 Present Absolute Format

This section describes the Present Absolute Format outbound structured field.

Subtopics:

- 5.17.1 Function
- 5.17.2 Format
- 5.17.3 Additional Content Description
- 5.17.4 Data Stream

---

### 5.17.1 Function

The Present Absolute Format structured field invokes a *named format* and

presents it to a device.

To present a format that does not require relocation, the Present Absolute Format structured field can be used. This structured field invokes the named format and presents it to a display or printer. The content of the Format Presentation Command (FPC) subfield specifies the type of command (Write, EW, or EWA) used to present the format. The WCC used in the presentation is specified in the WCC subfield. The orders and data used to generate the format are retrieved from storage in the control unit. The retrieval is based on the Format Name specified in the Name subfield as qualified by the Format Group Name that is currently selected for the specified partition. If a Present Format request is made before selecting a format group for the partition, a Format Group Not Selected error condition is reported, and the Present Format request is rejected.

When the outboard node is unable to retrieve the specified format, the Format Not Found situation is handled as an error condition.

---

## 5.17.2 Format

The following table shows the format of the Present Absolute Format structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'4B'</b> Identifies this structured field as Present Absolute Format
3	<b>PID</b>	Partition identifier (X'00' - X'7E')
4	<b>FPC</b>	Format Presentation Command:  <b>X'F1'</b> The data associated with the named format is written to the buffer of the specified partition (Write). <b>X'F5'</b> The viewport and buffer of the specified partition are cleared, and the data associated with the named format is written to the buffer of the specified partition (EW). <b>X'7E'</b> Same action as an EW.
5	<b>WCC</b>	3270 WCC byte
6-n	<b>NAME</b>	Name of the format being invoked

### Notes:

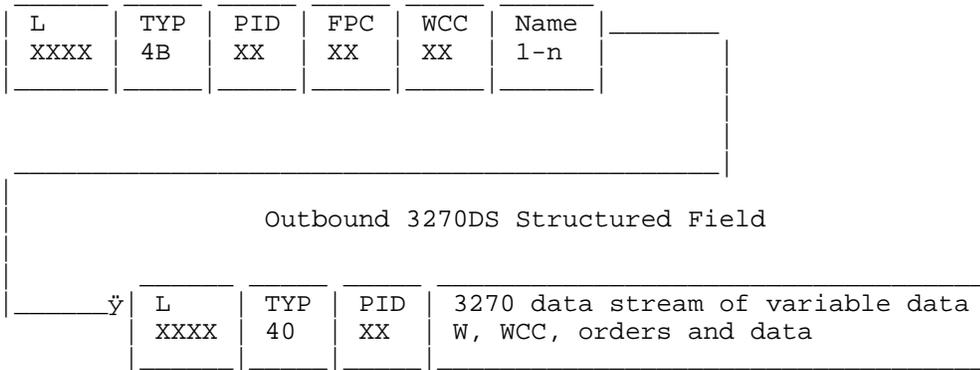
1. If PID is a text partition, an error occurs.
  2. EW and EWA do not change the screen size. Use the Erase/Reset structured field to change the screen size.
-

### 5.17.3 Additional Content Description

Values of the Format Presentation command that are not shown are reserved. If transmitted, they are rejected. WCC bits 0,2 and 3 are ignored if transmitted to a display. If bit 1 of the WCC = 1, a reset is performed. If PID is not equal to zero, only the indicated partition is reset.

To invoke a format that does not require relocation, to merge it with variable data transmitted in the data stream, and to present it to a device, a sequence of Present Absolute Format structured field and Outbound 3270DS structured fields can be used. The variable data is transmitted in the Outbound 3270DS structured field, as follows:

Present Absolute Format Structured Field



---

### 5.17.4 Data Stream

The relationship between the Format Presentation Command and the 3270DS Outbound structured field command is as follows:

- The first operation performed is that specified by the Present Absolute Format structured field. The command is taken from the Format Presentation Command subfield, the WCC is taken from the WCC subfield, and the orders and data used are retrieved from storage in the control unit based on the format name specified in the Name subfield.
- The second operation performed is that specified by the Outbound 3270DS structured field. The command is taken from byte 4, the WCC is taken from byte 5, and the orders and data from bytes 6 through *n* of the structured field.

*Example 1:* To erase the screen, present a format, and to merge variable data, an EW command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

*Example 2:* To present a format without erasing the screen and to merge variable data, a Write command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

---

## 5.18 Present Relative Format

This section describes the Present Relative Format outbound structured field.

Subtopics:

- [5.18.1 Function](#)
- [5.18.2 Format](#)
- [5.18.3 Additional Content Description](#)

---

### 5.18.1 Function

The Present Relative Format structured field invokes a *named format* and increments the buffer addresses contained in the format data stream by the 16-bit binary offset value, specified in the Format Offset Value subfield, before processing the format data for presentation to a device.

To present a format that requires relocation, the Present Relative Format structured field must be used. This structured field functions like the Present Absolute Format structured field, except the buffer addresses contained in the named format are incremented by the 16-bit binary value, before processing the format data stream for presentation to a device. If a Present Format request is made before selecting a format group for the partition, a Format Group Not Selected error occurs, and the Present Format request is rejected.

When the outboard node is unable to retrieve the specified format, the Format Not Found condition is handled as an error condition.

---

### 5.18.2 Format

The following table shows the format of the Present Relative Format structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'4C'</b> Identifies this structured field as Present Relative Format
3	<b>PID</b>	Partition identifier (X'00'-X'7E')
4-5	<b>FOV</b>	Format Offset Value: 16-bit binary offset value that is added to the buffer addresses contained in the named format prior to the data stream processing.
6	<b>FPC</b>	Format Presentation Command:

		<b>X'F1'</b> Write - The data associated with the named format is written to the buffer of the specified partition. <b>X'F5'</b> EW - The viewport and buffer of the specified partition are cleared, and the data associated with the named format is written to the buffer of the specified partition. <b>X'7E'</b> Same action as an EW.
7	<b>WCC</b>	3270 WCC Byte
8-n	<b>NAME</b>	Name of the format being invoked.

**Note:** If PID is a text partition, an error occurs.

### 5.18.3 Additional Content Description

The content of the Present Relative Format structured field is further described as follows:

**FPC** Values of the Format Presentation Command not shown are reserved. If transmitted, they are rejected.

**WCC** WCC bits 0, 2 and 3 are ignored if transmitted to a display. If WCC bit 1 = 1, a reset is performed. If PID is not equal to zero, only the indicated partition is reset.

To merge variable data with a relocated format, a sequence of Present Relative Format and Outbound 3270DS structured fields must be used, as follows:

Present Relative Format Structured Field

L	TYP	PID	FOV	FPC	WCC	Name
XXXX	4C	XX	XXXX	XX	XX	1-n

Outbound 3270DS Structured Field

ÿ	L	TYP	PID	3270 data stream of variable data
	XXXX	40	XX	W, WCC, orders and data

The structured fields are processed in the sequence in which they are transmitted in the data stream. Since structured fields can be used in various combinations, each structured field is processed as a separate entity. Using sequences of structured fields in the data stream allows multiple formats to be invoked and merged with multiple streams of variable data within one WSF transmission. The Format Offset Value has no effect on the Outbound 3270DS structured field.

## 5.19 Read Partition

This section describes the Read Partition outbound structured field.

Subtopics:

- [5.19.1 Function](#)
- [5.19.2 Format](#)
- [5.19.3 Additional Content Description](#)

---

### 5.19.1 Function

The Read Partition structured field reads data from a specific partition or it queries the device.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

The Read Partition structured field provides a means for the host to explicitly read a partition. In addition, the host can use the Read Partition structured field to request a 3270 data stream device or workstation to report functions supported by the device or workstation. This information is returned to the host in the form of Query Reply structured fields.

The reply to Read Partition (Query) is sent as a sequence of Query Reply structured fields, as follows:

X'88'	Query Reply	Query Reply
-------	-------------	-------------

---

### 5.19.2 Format

The following table shows the format of the Read Partition structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'01'</b> Identifies this structured field as Read Partition
3		<b>PID</b>	Partition identifier:
			<b>X'00'</b> - <b>X'7E'</b> (read operations), or <b>X'FF'</b> (query operations)

4		<b>TYPE</b>	The type of operation to be performed:
			<b>X'02'</b> Query
			<b>X'03'</b> Query List
			<b>X'6E'</b> Read Modified All (RMA)
			<b>X'F2'</b> Read Buffer (RB)
			<b>X'F6'</b> Read Modified (RM)
5	0-1	<b>REQTYP</b>	Request Type - present only for Type = X'03' (QUERY LIST)
			<b>B'00'</b> QCODE List
			<b>B'01'</b> Equivalent + QCODE List
			<b>B'10'</b> All
			<b>B'11'</b> Reserved
	2-7	<b>SFID</b>	Reserved
6-n		<b>QCODE</b>	Query Reply Codes for Type X'03'

### 5.19.3 Additional Content Description

The content of the Read Partition structured field is further described as follows:

**TYPE** Can contain one of the following values:

X'02' The structured field ends after byte 4.

X'03' Byte 5 is a flag byte. Bytes 6- n contain the QCODEs of the Query Replies being requested.

**Note:** Read type codes not shown here are reserved for future extension, and are rejected.

**REQTYP** Present only if TYPE = X'03'.

B'00' QCODE List indicates the only Query Replies being requested are those specified in bytes 6-n. If the value is B'00' but no list is present (count field is valid), a Null Query Reply is returned.

B'01' (Equivalent + QCODE List) indicates all the Query Replies that would be sent in response to a Query are sent in addition to those QCODES (if any) that are specified in the QCODE list (bytes 6-n). No duplicate Query Replies are sent. If the QCODE list requests a Query Reply that would be sent anyway, because of the B'01' flag, the Query Reply is sent only once.

B'10' All indicates that all the Query Replies that are supported are sent. If a QCODE list is present (bytes 6-n), the All flag overrides the list; that is, the list is ignored.

**QCODES** Contains the Query Reply codes requested when the TYPE value is X'03'.

The same QCODE can appear more than once in the list (bytes 6-n). However, only one Query Reply is returned for a particular QCODE value regardless of how many times it appears in the list.

All QCODE values in the list are valid. Those QCODES not supported are ignored. However, if none of the QCODES in the list are supported, a null Query Reply is returned.

If any of the following conditions exist, Read Partition is rejected (see Table A-1 in topic A.0):

- The device is in retry state (sense code X'0871').
- The terminal is not in receive (RCV) or contention (CONT) state.
- Read Partition is not the last structured field in the RU chain.
- The RU chain does not specify CD.
- | ◦ The operation type is Query or Query List and the PID is not X'FF'.
- The operation type is Read Modified, Read Modified All, or Read Buffer, and PID does not exist.
- The operation type is Read Modified All or Read Buffer, and PID specifies a text partition.

Otherwise, the following steps are performed:

1. The enter-inhibit condition is raised.
2. If the PID specified in Read Partition is not X'FF', INPID is set to the specified PID.
- | 3. INOP is set to the type of operation specified in Read Partition Type

Field.

4. For Read Modified (type code X'F6'), Read Modified All (type code X'6E'), or Read Buffer (type code X'F2') the data is transmitted inbound in the format specified by the operation and the inbound reply mode with an AID of X'61'.
5. For a Query (type code X'02'), or Query List (type code X'03') operation, a set of Query Replies is transmitted inbound. These replies describe the features on that device.
6. The display is placed in RCV state.
7. The device is placed in retry read state.

---

## 5.20 Request Recovery Data

This section describes the Request Recovery Data outbound structured field.

Subtopics:

- [5.20.1 Function](#)
- [5.20.2 Format](#)

---

### 5.20.1 Function

The Request Recovery Data structured field requests recovery data for Print Job Restart when sent from PLU to SLU.

---

### 5.20.2 Format

The following table shows the format of the Request Recovery Data structured field:

Byte	Content	Content Description
0 - 1	L	X'0005' Length of this structured field
2 - 3	SFID	X'1030' Identifies this structured field as Request Recovery Data
4	SFID	Reserved

This structured field must flow to enable the SLU to send the recovery data to the PLU.

---

## 5.21 Reset Partition

This section describes the Reset Partition outbound structured field.

Subtopics:

- [5.21.1 Function](#)
- [5.21.2 Format](#)

---

### 5.21.1 Function

The Reset Partition structured field resets definable partition characteristics for the specified partition (PID) to their default values:

Partition Characteristic	Default
Inbound Reply mode	Field
Format Group Selection	No format group selected
Set Partition Characteristics	None defined.

Reset partition has no effect if the partition A-MODE is text.

---

### 5.21.2 Format

The following table shows the format of the Reset Partition structured field:

Byte	Content	Content Description
0-1	L	X'0004' Length of this structured field
2	SFID	X'00' Identifies this structured field as Reset Partition
3	PID	Partition identifier (X'00' - X'7E')

---

## 5.22 Restart

This section describes the Restart outbound structured field.

Subtopics:

- [5.22.1 Function](#)
- [5.22.2 Format](#)

---

### 5.22.1 Function

The Restart structured field indicates that the restart is in progress and that a certain number of pages and lines should be bypassed before printing starts again using the data that follows. This structured field is sent from the PLU to the SLU.

## 5.22.2 Format

The following table shows the format of the Restart structured field:

Byte	Content	Content Description
0 - 1	<b>L</b>	Length of this structured field
2 - 3	<b>SFID</b>	<b>X'1033'</b> Identifies this structured field as Restart
4	<b>RES</b>	Reserved
5 - 6	<b>START PAGE</b>	Number of pages to skip on restart
7 - 8	<b>START LINE</b>	Number of lines to skip on page for restart
9 - N	<b>SCS DATA</b>	SCS data (noncompressed and noncompact) to set up for restart

The SNA Character String (SCS) data field must include the required Set Horizontal Format (SHF) and/or Set Vertical Format (SVF) and other SCS data for restart. The first byte of Function Management (FM) data resumes at the checkpoint spot or at the start of the SCB string or structured field containing the checkpoint spot in the following RU chain. There are three other data fields:

**START PAGE** Start Page is the indication sent by the PLU to the SLU of the number of pages that are to be bypassed prior to printing during a restart operation.

**START LINE** Start Line is the indication sent by the PLU to the SLU of the number of lines on the starting page that are to be bypassed prior to printing during a restart operation.

**SCS DATA** Data is sent to re-establish various parameters to the state at the time of the checkpoint (for example, the SVF and SHF codes indicated by the vertical and horizontal offsets in the Recovery Data structured field). Any SCS codes (control or graphic) can be included. Counts are reset to those at the time of the checkpoint after processing the SCS codes within the structured field.

## 5.23 SCS Data

This section describes the SCS Data outbound structured field.

Subtopics:

- [5.23.1 Function](#)
- [5.23.2 Format](#)

---

### 5.23.1 Function

The SCS Data structured field intermixes SCS data with other structured fields. The SCS data portion of the SCS Data structured field is treated in the same way as SCS data not delimited by an SCS Data structured field.

---

### 5.23.2 Format

The following table shows the format of the SCS Data structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'41'</b> Identifies this structured field as SCS Data
3	<b>PID</b>	Partition identifier (X'00' - X'7E')
4-n	<b>DATA</b>	SCS data

---

## 5.24 Select Color Table

This section describes the Select Color Table outbound structured field.

Subtopics:

- [5.24.1 Function](#)
- [5.24.2 Format](#)

---

### 5.24.1 Function

The Select Color Table structured field selects either the nonloadable or the loadable color table.

---

### 5.24.2 Format

The following table shows the format of the Select Color Table structured

field:

Byte	Content	Content Description
0 - 1	<b>L</b>	<b>X'0006'</b> Length of this structured field
2 - 3	<b>SFID</b>	<b>X'0F04'</b> Identifies this structured field as Select Color Table
4 - 5	<b>COMMAND</b>	Command

**Note:** For information regarding the format and operation of the Command parameter, refer to related graphics documentation.

---

## 5.25 Select Format Group

This section describes the Select Format outbound structured field.

Subtopics:

- [5.25.1 Function](#)
- [5.25.2 Format](#)

---

### 5.25.1 Function

The Select Format Group structured field selects the format group that is to be the source of formats in subsequent operations.

The Select Format Group structured field defines, for the specified partition, a single format group that is to be the current source of formats for subsequent Present Absolute and Present Relative Format requests for that partition. The last group selected remains current until it is reset or replaced by a subsequent Select Format Group for that partition. Subsequent Present Absolute and Present Relative Format requests specifying a particular partition are qualified by the current format group for that partition.

If a Present Absolute or Present Relative Format request is made before a format group is selected, it is rejected.

**Note:** If PID is a text partition the structured field is rejected.

---

### 5.25.2 Format

The following table shows the format of the Select Format structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'4A'</b> Identifies this structured field as Select Format Group
3	<b>PID</b>	Partition identifier (X'00' - X'7E')
4-n	<b>GRP</b>	Format group

---

## 5.26 Set Checkpoint Interval

This section describes the Set Checkpoint Interval outbound structured field.

Subtopics:

- [5.26.1 Function](#)
- [5.26.2 Format](#)

---

### 5.26.1 Function

The Set Checkpoint Interval structured field passes from the host to the terminal the number of pages that are to be in the interval between checkpoints. It contains the recovery data needed by the host to recover from the error.

This structured field resets all previous checkpointed information. The checkpoint counters begin with the first FM data byte following the structured field.

---

### 5.26.2 Format

The following table shows the format of the Set Checkpoint Interval structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0007'</b> Length of this structured field
2-3	<b>SFID</b>	<b>X'1032'</b> Identifies this structured field as Request Recovery Data
4	<b>SFID</b>	Reserved
5-6	<b>INTERVAL</b>	Checkpoint interval -- Specifies the number of pages in the interval between terminal checkpoints. This number enables checkpointing of jobs that use the short forms without taking an excessive number of checkpoints. A zero value indicates that

## 5.27 Set MSR Control

This section describes the Set MSR Control outbound structured field.

Subtopics:

- [5.27.1 Function](#)
- [5.27.2 Format](#)
- [5.27.3 Additional Content Description](#)

### 5.27.1 Function

The Set MSR Control structured field allows the application program associated with the host to set the device input states and their associated indicators. In addition, this structured field allows the application program to set user indicators.

### 5.27.2 Format

The following table shows the format of the Set MSR Control structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	<b>X'000A'</b> Length of this structured field
2-3		<b>SFID</b>	<b>X'0F01'</b> Identifies this structured field as Set MSR Control
4		<b>PID</b>	Partition identifier
5		<b>TYPE</b>	MSR type = X'01' All other values reserved
6		<b>STATE MASK</b>	Bit i=B'1' selects states to be modified by corresponding bit in byte 7:
	0	SFID	<b>USER</b> User mode
	1	SFID	<b>LOCK</b> Locked /Unlocked
	2	SFID	<b>AUTO</b> Auto Enter
	3	SFID	<b>AI1S</b> Audible Ind 1 Suppress
	4	SFID	<b>AI2S</b> Audible Ind 2 Suppress

	5-7	SFID	RES Reserved
7	0-7	STATE VALUE	Bit i=B'0' sets state off Bit i=B'1' sets state on
8		IND MASK	Bit j=B'1' selects corresponding indicator
9		IND VALUE	Bit j=B'0' sets indicator off Bit j=B'1' sets indicator on

### 5.27.3 Additional Content Description

This structured field is sent by the application to control the following device states and indicators.

#### Mask Byte 6 (bits 0-4)

Identifies the state of the device. When these bits are set to B'1' the corresponding bit in byte 7 determines whether the state is on or off.

**USER** This state of the device is set by the application, and the meaning is determined by the application.

**LOCK** When this state is set, the device is disabled for input.

#### **AUTO ENTER**

When this state is set, an input operation on the device will cause an Enter operation from the device, regardless of the type of card read. When this state is reset, an input operation on the device will not cause an Enter operation from the device unless a secure card is read.

**AI1S** When the AUDIBLE IND 1 SUPPRESS state is set, the internally generated audible indication 1 is suppressed.

**AI2S** When the AUDIBLE IND 2 SUPPRESS state is set, the internally generated audible indication 2 is suppressed.

#### **STATE VALUE**

Determines whether the state is on or off. If the corresponding bit in byte 6 is set to B'1', apply the bit in byte 7.

#### **IND MASK and IND VALUE**

These indicators are set by the application program to convey information to the operator of the device. For the Set MSR Control structured field Type X'01', magnetic reader indicators 0-4 are defined. These indicators are reset by the next read to type the correct partition. The device states set by the Set MSR Control structured field are device related, while the

magnetic reader indicators are partition related. Therefore, the magnetic reader indicators are saved on a partition basis and only affect the device when the partition is active.

---

## 5.28 Set Partition Characteristics

This section describes the Set Partition Characteristics outbound structured field.

Subtopics:

- [5.28.1 Function](#)
- [5.28.2 Format](#)
- [5.28.3 Operation](#)
- [5.28.4 Viewport Outline Self-Defining Parameter](#)
- [5.28.5 Enable User Call-up Self-Defining Parameter](#)
- [5.28.6 Select Base Character Set Self-Defining Parameter](#)

---

### 5.28.1 Function

The Set Partition Characteristics structured field provides a base for optional parameters which further define the partition characteristics beyond those given in the Create Partition structured field.

---

### 5.28.2 Format

The following table shows the format of the Set Partition Characteristics structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structure
2-3		<b>SFID</b>	<b>X'0F08'</b> Identifies this structured field as Set Partition Characteristics
4		<b>PID</b>	Partition identifier (X'00' - X'7E' or X'FF')
5-6		<b>FLAGS</b>	Reserved

---

### 5.28.3 Operation

The Set Partition Characteristics structured field is used to change operational characteristics at a logical terminal. The characteristics which are subject to change must have default values or definitions at logical terminal initialization (power on) or reset time. These definitions are how the characteristics would behave in a 3270 subsystem that did not allow for changing of the characteristics and are referred to as *3270 Default Function definitions*.

When the Set Partition Characteristics structured field is allowed to change the operational characteristics, the changes can be directed to apply to all partitions or specific partitions. When the changes are to apply to all partitions, they are referred to as *Logical Terminal Characteristics*. When the changes are to apply to a specific partition, they are referred to as *Partition Characteristics*.

An inheritance structure exists between the 3270 defaults, the logical terminal, and the partitions. This inheritance is for the partition to inherit from the 3270 default definition. Thus, without any Set Partition Characteristics structured field, all partitions would behave as a logical terminal which did not support the structured field. When a Set Partition Characteristics structured field is sent with a PID of X'FF', the accompanying SDPs set the logical terminal characteristics. Now it is these settings instead of the 3270 default functions which are inherited by present or future partitions created in the logical terminal. These functional definitions for a given self-defining parameter apply until the logical terminal characteristic is altered or the logical terminal is reset.

Finally, when the Set Partition Characteristics SDPs are directed to the specific partition, the characteristics are set for that partition only. Partition Characteristics take precedence over the Logical Terminal Characteristics and the 3270 Default Function Definition for a given self-defining parameter. These functional definitions for a given self-defining parameter apply until the partition characteristic is altered or the partition is reset.

This structured field can be sent any number of times to change the characteristics of the addressed partition (Partition ID of X'00' - X'7E') or the logical terminal (Partition ID of X'FF').

A partition characteristic established by this structured field is retained until the partition no longer exists (see reset information following) or until the characteristic is changed by another occurrence of this structured field having a self-defining parameter which alters it.

If no self-defining parameters are included in this structured field, then the Partition Characteristics structured field has no effect.

An error situation exists when any of the following arises:

- The structured field is not supported.
- The named partition does not exist.
- The self-defining parameter is not supported.
- The Set Partition Characteristics structured field is addressed to the logical terminal (PID = X'FF'), but the Partition Characteristics Query Reply does not indicate logical terminal support of an included self-defining parameter.
- The Set Partition Characteristics structured field is addressed to a partition (PID = X'00' - X'7E'), but the Partition Characteristics Query Reply does not indicate partition support of an included self-defining parameter.

The Set Partition Characteristics structured field can include one or more self-defining parameters. The self-defining parameters provide a functional definition of their respective operations. At the logical terminal level, the self-defining parameters can specify that:

- The 3270 Default Function Definition is to be inherited.
- The functional definition is the same as the 3270 Default Function Definition but is not inherited.
- That one of the alternative functional definitions is to apply.

For a specific partition, the self-defining parameters can specify that:

- The Logical Terminal Characteristics functional definition is to be inherited.
- The functional definition is the same as the Logical Terminal Characteristics but is not inherited.
- One of the alternative functional definitions is to apply.

The self-defining parameters can be sent as a 2-byte structure, length byte and self-defining parameter identifier only, or as a structure containing more than two bytes when the parameter values are included. A self-defining parameter more than two bytes long explicitly provides a functional definition at the addressed level (logical terminal or partition) even though this functional definition can be synonymous with a functional definition at a lower level.

When a self-defining parameter is more than two bytes long, but less than its maximum potential length, the omitted parameters assume defined values to complete the functional definition at the addressed level. When a 2-byte self-defining parameter is sent (length byte and identifier byte only), the functional definition at the addressed level (logical terminal or partition) is set to inherit.

Parameters set by a self-defining parameter addressed to the logical terminal are reset to inherit by the following:

- A Set Partition Characteristics structured field addressed to the logical terminal and including a self-defining parameter that includes only the length byte and the identifier byte.
- An Erase/Reset structured field, a WCC reset bit on with Erase/Write or Erase/Write Alternate command, Power off, Bind, Clear local function, and System Request local function. Note that destroying the last partition does not reset the parameters.

Parameters set by a self-defining parameter addressed to an individual partition are reset to inherit when:

- A Set Partition Characteristics structured field is addressed to that partition and includes a self-defining parameter that includes only

the length byte and the identifier byte.

- An individual partition is reset by Partition Reset, Erase/Reset, Create Partition for a partition with the same PID, Destroy Partition, WCC reset bit on with Erase/Write or Erase/Write Alternate in command form or in an Outbound 3270DS structured field.

---

## 5.28.4 Viewport Outline Self-Defining Parameter

The Viewport Outline self-defining parameter specifies the maximum thickness of the distinguishing outline of a viewport. The outline of the viewport begins at the outermost edge of the outermost row or column of the viewport. This outline consumes up to Byte 2 pel positions from the outermost edge of the viewport. The outline thickness is specified by the outline thickness value.

The following table shows the format of the Viewport Outline self-defining parameter:

Byte	Content	Content Description
0	L	X'06' Length of this structure
1	SDPID	X'01' Identifies this self-defining parameter as Viewport Outline Parameters
2	OT	Top edge outline thickness
3	OB	Bottom edge outline thickness
4	OL	Left edge outline thickness
5	OR	Right edge outline thickness

The values specified in bytes 2 through 5 of the Viewport Outline self-defining parameters range in value from zero to the value specified in the Viewport Outline structured field of the Partitions Characteristics Query Reply.

An error exists when any of the following arises:

- The self-defining parameter specifies an incorrect length.
- A self-defining value is outside the range specified as supported by the Query Reply.

---

## 5.28.5 Enable User Call-up Self-Defining Parameter

This Set Partition Characteristics self-defining parameter is used to provide for the selection of the operational mode of User Selected Formats.

The following table shows the format of the Enable User Call-up

self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'03'</b> Length of structure
1		<b>SDPID</b>	<b>X'04'</b> Identifies this self-defining parameter as Local User Call-up
2		<b>FLAGS</b>	5
	0	<b>ENABLE</b>	User Selectable Formats
			<b>B'0'</b> Disable User Selectable Formats <b>B'1'</b> Enable User Selectable Formats
	1	<b>SEARCH</b>	Perform name search
			<b>B'0'</b> Perform name search globally <b>B'1'</b> Perform name search qualified by the currently selected group name.
	2-7	<b>RES</b>	Must be set to zeros

The content of the Enable User Call-up self-defining parameter is further described as follows:

**ENABLE** User selected formats are enabled if the SDP is received with this bit set to B'1'. If the bit is set to B'0', they are disabled.

**SEARCH** If this bit is set to B'0', the local name search is performed for all stored formats. If this bit is set to B'1', the local name search is performed only for formats assigned to the currently selected group.

Subtopics:

- 5.28.5.1 Operation

---

### 5.28.5.1 Operation

These parameters remain set until one of the following conditions occurs:

- A Set Partition Characteristics structured field is received that resets the parameter.
- An Erase/Write or Erase/Write Alternate command is received with WCC = Reset.

- An SNA bind is received.
- An Erase/Reset structured field is received.

**Note:** These parameters are not reset by the Clear key, SYS REQ key, TEST key, or by turning on the power to the device.

## 5.28.6 Select Base Character Set Self-Defining Parameter

This Set Partition Characteristics self-defining parameter is used to select the Base Character Set that is to be used at the device.

The following table shows the format of the Select Base Character Set self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'04'</b> Length of structure
1		<b>SDPID</b>	<b>X'05'</b> Identifies this self-defining parameter as Select Base Character Set
2		<b>FLAGS1</b>	
	0-3	<b>BASE</b>	Base Character Set Index
	4-7		Reserved - must be set to zeros
3		<b>FLAGS2</b>	Reserved - must be set to zeros

The content of the Select Base Character Set self-defining parameter is further described as follows:

**BASE** The Base Character Set is established by indexing the list of Base Character Set descriptors in the order returned in the Character Sets Query Reply.

## 5.29 Set Printer Characteristics

This section describes the Set Printer Characteristics outbound structured field.

Subtopics:

- [5.29.1 Function](#)
- [5.29.2 Format](#)
- [5.29.3 Early Print Complete Self-Defining Parameter](#)

---

## 5.29.1 Function

The Set Printer Characteristics (SPC) structured field allows the application to control the setting and resetting of certain functions and modes within a control unit terminal (CUT) printer. See "Early Print Complete Self-Defining Parameter" in topic 5.29.3 .

The SPC structured field is valid only in non-SNA environments.

---

## 5.29.2 Format

The following table shows the format of the Set Printer Characteristics structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F84'</b> Identifies this structured field as Set Printer Characteristics
4		<b>FLAGS</b>	
	0	RSTALL	Reset All Characteristics
			<b>B'0'</b> Off (No reset all) <b>B'1'</b> On (Reset all)
	1-7	Reserved	Reserved
5		<b>FLAGS</b>	Reserved

The base can be sent without any self-defining parameters appended when a reset of all previously set functions is desired without setting any new ones (RSTALL=ON). If the base is sent without any self-defining parameters appended with No Reset All indicated (RSTALL = OFF), it is treated as a No-op by the printer and a normal response is returned.

Within a WSF transmission, the SPC must be sent as the first structured field in the transmission. However, if the SPC structured field is used with the Data Chain structured field, it must be sent immediately following a Data Chain (Begin or Only) structured field. If it is used after a Data Chain (Begin) structured field, and a recoverable error occurs before the end-of-chain, the unit of recovery is the chain. An example of recovery would be to retry the chain from the beginning.

The Early Print Complete (EPC) self-defining parameter is used to set an early print complete mode of operation on or off. This parameter provides for enabling/disabling of the printer operator control. (EPC mode can be on or off depending on the setting of the operator control.) The default is printer operator control enabled.

If the RSTALL flag in the base part of the structured field equals Reset, the default is set. However, when the Early Print Complete self-defining parameter is present, the SREPC flag overrides the RSTALL flag.

For example, if RSTALL = Reset and SREPC = EPC on (operator control disabled), the EPC mode is on and operator control is disabled.

### 5.29.3 Early Print Complete Self-Defining Parameter

The Early Print Complete (EPC) self-defining parameter can be appended to the base portion of this structured field. EPC allows the loading of the printer buffer and the printing operation to overlap, thereby improving throughput to high-speed printers. EPC is supported only in non-SNA environments. See "Early Print Complete (EPC) Operations" in topic 8.9.

The following table shows the format of the Early Print Complete self-defining parameter:

Byte	Bit	Content	Content Description
0		L	'03' Length of parameter
1		SDPID	X'01' Identifies this self-defining parameter as Early Print Complete
2		SREPC	Set/Reset Early Print Complete
	0-1	SFID	B'00' Enable operator control B'01' EPC Off - disable operator control B'10' EPC On - disable operator control B'11' Reserved
	2-7	Reserved	Reserved

Prior to use of the SPC structured field, an application should determine the characteristics of the printer by issuing a Read Partition Query List structured field. The Query List structured field must either contain a "Q" code of X'A9' or indicate ALL.

When EPC is on, the printer returns a positive completion status for the print information received after it is verified but before it is printed. If an error occurs while this information is printing, the error is reported and associated with print information received later. Because the error is not properly synchronized with print information to which it pertains, error handling can be affected.

If an SPC structured field is detected out of sequence, it is rejected with an Op Check sense code sent to the host.

All set printer characteristics are reset to default by any of the following:

- Receipt of an EW or EWA command with WCC = Reset
- Power-on reset (POR)
- RSTALL=X'1B'.

If a system performs its error handling by reprinting the job, EPC should be turned on at the beginning of the job and turned off with the last piece of print information for the job. This requires the code that manages EPC to recognize job boundaries. In the SNA environment, LU1 can be used to overlap the loading of a printer's buffer with the print operation if the serial load and print required by LU3 results in performance problems.

---

## 5.30 Set Reply Mode

This section describes the Set Reply Mode outbound structured field.

Subtopics:

- [5.30.1 Function](#)
- [5.30.2 Format](#)
- [5.30.3 Additional Content Description](#)

---

### 5.30.1 Function

The Set Reply Mode structured field defines the format of any subsequent inbound data streams from an alphanumeric partition by defining the values required by the host application.

The Reply Modes Query Reply is used to inform the host of the attribute orders that the device supports in both the inbound and the outbound direction and the modes that can be set by the Set Reply Modes structured field.

The Set Reply Mode structured field is sent outbound to devices that support at least the Extended Field Mode of operation. The Extended Field Mode indicates support of the Start Field Extended (SFE) and the Modify Field (MF) orders. See ["Start Field Extended \(SFE\)"](#) in topic 4.3.2 and ["Modify Field \(MF\)"](#) in topic 4.3.5. Character Mode can be selected if the device has indicated its support in the Reply Modes Query Reply. See ["Set Attribute \(SA\)"](#) in topic 4.3.4.

Subsequent inbound data streams from the named partition sent inbound in response to a read command are in the format defined by the Set Reply Mode structured field. This reply mode remains in effect until set by another Set Reply Mode or until reset by a reset action.

The inbound data stream from a Text partition is not controlled by a Reply Mode. Hence, a Set Reply Mode structured field sent to a text partition is rejected.

---

### 5.30.2 Format

The following table shows the format of the Set Reply Mode structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'09'</b> Identifies this structured field as Set Reply Mode
3	<b>PID</b>	Partition identifier (X'00' through X'7F')
4	<b>MODE</b>	<b>X'00'</b> Field mode <b>X'01'</b> Extended field mode <b>X'02'</b> Character mode <b>Other</b> Reserved
5-n	<b>ATTR LIST (Present only if mode is X'02')</b>	The type codes for the attribute types that: <ul style="list-style-type: none"> <li>◦ Are transmitted using SA orders if mode X'02' is specified.</li> <li>◦ The operator can select from the keyboard. (The operator can select the attribute value to associate with the keyed data.)</li> </ul>

### 5.30.3 Additional Content Description

The values that can appear in each mode are listed below:

**Mode**

**Values**

Field

3270 orders (SBA, SF)  
 Characters  
 Field attributes  
 GE

Extended field

3270 orders (MF, SBA, SF, SFE) Characters  
 Field attributes  
 Extended field attributes  
 GE

Character

3270 orders (MF, SBA, SF, SFE, SA)  
 Characters  
 Field attributes  
 Extended field attributes  
 Character attributes listed in Set Reply Mode structured field  
 GE.

In field mode or extended field mode, attribute selection by the operator is not allowed. Any operator selection allowed by a previous Set Reply Mode structured field is reset to the default value.

In character mode, the listed attribute types can have values selected by the operator. Any operator selection current when Set Reply Mode is

interpreted is not reset. Attribute selection by the operator for attribute types not explicitly listed is not allowed.

Mode values not shown here are reserved and are rejected. Attribute types other than those defined are rejected.

---

## 5.31 Set Window Origin

This section describes the Set Window Origin outbound structured field.

Subtopics:

- [5.31.1 Function](#)
- [5.31.2 Format](#)

---

### 5.31.1 Function

The Set Window Origin structured field changes the position of the window within the presentation space of the specified partition.

If the named partition does not exist, an error situation arises. (See [Table A-1 in topic A.0.](#))

The window of the specified partition is repositioned to the new origin. If the following condition occurs, the window is not moved and a sense code is returned:

$$\begin{aligned} RW + HW &> H \\ CW + WW &> W \end{aligned}$$

where:

HW    Height of window (viewport)  
H     Height of presentation space  
WW    Width of window (viewport)  
W     Width of presentation space.

The current cursor position for the named partition is not changed unless the new window origin results in the current cursor position's being outside the window. If outside the window, the current cursor position is moved by the minimum number of row and columns needed to place it in the nearest row or column of the window.

**Note:** The Modify Partition structured field can also be used to change the window position.

---

### 5.31.2 Format

The following table shows the format of the Set Window Origin structured

field:

Byte	Content	Content Description
0-1	L	X'0008' Length of this structured field
2	SFID	X'0B' Identifies this structured field as Set Window Origin
3	PID	Partition identifier (X'00' through X'7E')
4-5	RW	Row position of the window origin
6-7	CW	Column offset of the window origin

---

## 5.32 Type 1 Text Outbound

This section describes the Type 1 Text Outbound structured field.

Subtopics:

- [5.32.1 Function](#)
- [5.32.2 Format](#)
- [5.32.3 Additional Content Description](#)

---

### 5.32.1 Function

The Type 1 Text Outbound structured field transmits data to a text partition.

The structured field is checked for validity. If any of the following conditions exist, the structured field is rejected:

- The partition A-MODE is not X'2'.
- The named partition does not exist.

---

### 5.32.2 Format

The following table shows the format of the Type 1 Text Outbound structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2-3	SFID	X'0FC1' Identifies this structured field as Type 1 Text Outbound
4	PID	Partition identifier

5-6	<b>RES</b>	Reserved
7-n	<b>DATA</b>	Text data

### 5.32.3 Additional Content Description

The content of the Type 1 Text Outbound structured field is further described as follows:

**DATA** The text data is interpreted and placed in the text buffer at the current buffer address, that is, following any previous data in the buffer.

If any of the following errors occur during this process, the structured field is rejected:

- An unsupported X'2B' control is detected.
- A parameter error occurs in a X'2B' control.
- An unsupported 1-byte control is detected.
- The text data cannot be contained in the text buffer.

During this interpretation, nulls (X'00') are treated as No-op. They are not placed in the text buffer.

As the text data is interpreted, the current buffer address is incremented by one for each byte placed in the buffer. The current buffer address is updated both by host writes, and by operator interaction (that is, it is always the address of the first empty byte in the buffer).

The buffer is formatted to the presentation space.

If DATA contains an insert cursor control, the screen cursor is placed at the presentation position of the element immediately following this insert cursor control. If there are no elements in DATA following the insert cursor, the screen cursor is placed at the presentation position immediately following that of the last element of DATA.

The window and cursor positions are then resolved as follows:

- If the resulting cursor position is inside the window, the window position is unchanged.
- If the cursor position is outside the window, the window moves to contain it. Vertically, the window moves by the minimum number of rows. Horizontally, the window moves so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor is at one of the two center positions.)

The host can use the Modify Partition structured field to set the window

origin.

---

## 5.33 Outbound/Inbound Structured Fields

The structured fields described in the following sections are structured fields that can be transmitted either from the application to the device or from the device to the application. The structured field format is the same for both inbound and outbound structured fields.

---

## 5.34 Data Chain

This section describes the Data Chain outbound/inbound structured field.

Subtopics:

- [5.34.1 Function](#)
- [5.34.2 Format](#)
- [5.34.3 Additional Content Description](#)

---

### 5.34.1 Function

The Data Chain structured field provides a data-chaining function for non-SNA environments. It identifies the individual transmissions as being part of a single message.

This structured field uses the structured field grouping function described in "[Structured Field Grouping](#)" in [topic 5.2](#) to provide a chaining function. Items such as structured fields and control code sequences contained in the message can span transmissions that are related through Data Chain structured fields. This allows a message to be divided into transmissions without any examination or manipulation of the message content. The Data Chain structured field was defined for use in the non-SNA environment. Its use in the SNA environment is unnecessary because of the chaining function provided in SNA.

---

### 5.34.2 Format

The following table shows the format of the Data Chain structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0006' Length of this structured field
2-3		SFID	X'0F21' Identifies this structured field as Data Chain
4	0	RES	Reserved
	1-2	GROUP	B'00' Continue

			B'01' End B'10' Begin B'11' Only
	3-4	INCTRL	Inbound Control
			B'00' No change B'01' Enable Inbound Data Chaining B'10' Disable Inbound Data Chaining B'11' Reserved
	5-7	RES	Reserved
5		<b>FLAGS</b>	Reserved

---

### 5.34.3 Additional Content Description

The content of the Data Chain structured field is further described as follows:

**GROUP** Establishes whether the data following the Data Chain is the beginning, continuation, or the end of a chain of grouped transmissions. The value B'11' (only) indicates the entire message is contained in the transmission that follows the Data Chain structured field.

**INCTRL** This parameter is only set in the outbound Data Chain structured field to control how the inbound data is sent. It is not set in the inbound Data Chain structured field and should be zeros.

An outbound Data Chain structured field with INCTRL = B'01' (Enable Inbound Data Chaining) allows the device to use the Data Chain structured field for inbound transmissions. If inbound data chaining is already enabled, it remains enabled.

An outbound Data Chain structured field with INCTRL = B'10' (Disable Inbound Data Chaining) prevents the device from using the Data Chain structured field for inbound transmissions. If inbound data chaining is already disabled, it remains disabled until subsequently enabled by an outbound Data Chain structured field with INCTRL = B'01'. (Enable Inbound Data Chaining).

An outbound Data Chain structure field with INCTRL = B'00' (no change) has no effect on the enable/disabled state.

---

## 5.35 Destination/Origin

This section describes the Destination/Origin outbound/inbound structured field.

Subtopics:

- [5.35.1 Function](#)
- [5.35.2 Format](#)
- [5.35.3 Additional Content Description](#)

## 5.35.1 Function

The Destination/Origin structured field identifies the destination or origin of structured fields in a single session multi-device (workstation) implementation.

Outbound (from the host) the ID indicates the destination of the structured fields that follow. Inbound (to the host), the ID indicates the origin of the structured fields that follow.

At the beginning of a transmission, the destination or origin is the default (primary display).

Once a Destination/Origin structured field establishes the destination or origin, it applies until either another Destination/Origin structured field establishes a new destination or origin or a new transmission starts.

## 5.35.2 Format

The following table shows the format of the Destination/Origin structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0008' Length of this structured field
2-3		SFID	X'0F02' Identifies this structured field as Destination/Origin
4		FLAGS	
	0-1	INCTRL	Input Control:
			B'00' Enable input B'01' No change B'10' Disable input B'11' Reserved
	2-7	RES	Reserved
5		FLAGS	Reserved
6-7		SFID	Identifies destination or origin of the structured fields which follow in the data stream

---

### 5.35.3 Additional Content Description

The content of the Destination/Origin structured field is further described as follows:

**INCTRL** Applies only on outbound (to the auxiliary device); on inbound the INCTRL flag is ignored.

B'00' Allows the auxiliary device to send data. If the auxiliary device is already enabled, it remains enabled.

B'01' A change does not occur in the enabled/disabled status.

B'10' The auxiliary device is not permitted to send data until subsequently enabled by a Destination/Origin structured field with INCTRL flag = B'00' - If the auxiliary device is already disabled, the INCTRL flag = B'10' causes no change.

If a Destination/Origin structured field is directed to the base display (ID = X'0000'), the INCTRL flag applies on a global basis: all the auxiliary devices supported are enabled, disabled, or unchanged as a group.

**Note:** There is one exception where an auxiliary device can send input without being enabled. An Exception Condition structured field, reporting unavailability of the auxiliary device, can be sent in reply to a Destination/Origin structured field sequence attempting to use the auxiliary device.

**DOID** The valid values for the DOID are:

- X'0000' (permanently assigned to the primary display)
- All ID values returned in the Query Replies for Auxiliary devices.

All other values are invalid and must be rejected.

---

## 5.36 Object Control

This section describes the Object Control outbound/inbound structured field.

Subtopics:

- [5.36.1 Function](#)
  - [5.36.2 Format](#)
  - [5.36.3 Additional Content Description](#)
-

## 5.36.1 Function

The Object Control structured field carries the Object Data between the device and the host. It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the Object Data.

## 5.36.2 Format

The following table shows the format of the Object Control structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F11'</b> Identifies this structured field as Object Control
4		<b>PID</b>	Partition identifier
5		<b>FLAGS</b>	
	0-1	SPANF	Spanning flag (outbound)
			<b>B'00'</b> Not first and not last (middle) <b>B'01'</b> Last but not first <b>B'10'</b> First but not last <b>B'11'</b> First and last (only)
	2-3	MODE	Control interpretation mode (outbound)
			<b>B'00'</b> Immediate mode <b>B'10'</b> Store mode <b>Other</b> Reserved
	4-7	RES	Reserved
6		<b>OBJTYP</b>	Object type
			<b>X'00'</b> Graphics <b>X'01'</b> Image
7-n		<b>DATA</b>	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

---

### 5.36.3 Additional Content Description

The content of the Object Control structured field is further described as follows:

- SPANF** Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.
- MODE** Specifies how the Object Data contained in the DATA parameter (byte 7-n) should be interpreted in the device. This flag is applicable to outbound transmissions only.
- B'00' Immediate mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise they are rejected.
- B'10' Store mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise it is rejected.
- OBJTYP** Contains the object type being carried in this structured field.
- DATA** Contains the self-describing Object Data commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

---

## 5.37 Object Data

This section describes the Object Data outbound/inbound structured field.

Subtopics:

- [5.37.1 Function](#)
- [5.37.2 Format](#)
- [5.37.3 Additional Content Description](#)

---

### 5.37.1 Function

The Object Data structured field carries the object data between the device and the host.

It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object data.

---

## 5.37.2 Format

The following table shows the format of the Object Data structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F0F'</b> Identifies this structured field as Object Data
4		<b>PID</b>	Partition identifier
5		<b>FLAGS</b>	
	0-1	SPANF	Spanning flag (outbound)
			<b>B'00'</b> Not first and not last (middle) <b>B'01'</b> Last but not first <b>B'10'</b> First but not last <b>B'11'</b> First and last (only)
	2-3	MODE	Control interpretation mode (outbound)
			<b>B'00'</b> Immediate mode <b>B'10'</b> Store mode <b>Other</b> Reserved
	4-7	RES	Reserved
6		<b>OBJTYP</b>	Object type
			<b>X'00'</b> Graphics <b>X'01'</b> Image
7-n		<b>DATA</b>	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

## 5.37.3 Additional Content Description

The content of the Object Data structured field is further described as follows:

**SPANF** Permits the segmentation of a structured field into several

related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.

**MODE** Specifies how the object data contained in the DATA parameter (byte 7-n) should be interpreted in the device. This flag is applicable to outbound transmissions only.

B'00' Immediate mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise they are rejected.

B'10' Store mode: For graphics, data units contained within the DATA parameter are stored if the device has this capability; otherwise it is rejected.

**OBJTYP** Contains the object type being carried in this structured field.

**DATA** Contains the self-describing Object Data commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

---

## 5.38 Object Picture

This section describes the Object Picture outbound/inbound structured field.

Subtopics:

- [5.38.1 Function](#)
- [5.38.2 Format](#)
- [5.38.3 Additional Content Description](#)

---

### 5.38.1 Function

The Object Picture structured field carries the object pictures between the device and the host. It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object pictures.

---

### 5.38.2 Format

The following table shows the format of the Object Picture structured field:

Byte	Bit	Content	Content Description
0-1		L	Length of this structured field

2-3		<b>SFID</b>	<b>X'0F10'</b> Identifies this structured field as Object Picture
4		<b>PID</b>	Partition identifier
5		<b>FLAGS</b>	
	0-1	<b>SPANF</b>	Spanning flag (outbound)
			<b>B'00'</b> Not first and not last (middle) <b>B'01'</b> Last but not first <b>B'10'</b> First but not last <b>B'11'</b> First and last (only)
	2-3	<b>MODE</b>	Interpretation mode (outbound)
			<b>B'00'</b> Immediate mode <b>B'10'</b> Store mode <b>B'01'</b> Reserved <b>B'11'</b> Store and draw mode
	4-7	<b>RES</b>	Reserved
6		<b>OBJTYP</b>	Object type
			<b>X'00'</b> Graphics <b>X'01'</b> Image
7-n		<b>DATA</b>	Data appropriate to the object type. For the format and contents of this parameter, refer to the appropriate graphics or image publications.

### 5.38.3 Additional Content Description

The content of the Object Picture structured field is further described as follows:

**SPANF** Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.

**MODE** Specifies how the object pictures contained in the DATA parameter (byte 7-n) should be interpreted in the device. This flag is applicable to outbound transmissions only.

B'00' Immediate mode: For graphics, each chained segment is executed immediately as it is received and not stored.

Non-chained segments and extended drawing routines are stored if the device has this capability; otherwise they are rejected.

B'10' Store Mode: Each object picture contained within the DATA parameter is stored if the device has this capability; otherwise it is rejected. No operation occurs.

B'11' Store and draw mode: For graphics, each segment and extended drawing routine contained within the DATA parameter is stored if the device has this capability; otherwise it is rejected. Additionally, chained segments are executed.

**OBJTYP** Contains the object type being carried in this structured field.

**DATA** Contains the self-describing Object Picture commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

---

## 5.39 OEM Data

This section describes the OEM Data outbound/inbound structured field.

Subtopics:

- [5.39.1 Function](#)
- [5.39.2 Optional Parameters](#)
- [5.39.3 Format](#)
- [5.39.4 Additional Content Description](#)

---

### 5.39.1 Function

The OEM Data structured field carries the non-IBM-defined data (or a value-added version) to or from another equipment manufacturer's (OEM) auxiliary device. An OEM device is defined here as a device that is manufactured outside of IBM. The device can carry either an outside manufacturer logo or an IBM logo.

The data to or from an auxiliary device on a 3270 data stream workstation must be in structured field form. The OEM Data structured field provides a means of carrying the manufacturer's defined data (or a value added version) to or from an OEM auxiliary device.

---

### 5.39.2 Optional Parameters

All parameters through byte 6 must be present. Parameters in bytes 7-n are optional. An OEM Data structured field of length 7 is valid; for example, it could be used to end a group when there is no more data available to send.

### 5.39.3 Format

The following table shows the format of the OEM Data structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F1F'</b> Identifies this structured field as OEM Data
4		<b>RES</b>	Reserved
5	0-1	<b>GROUP</b>	<b>B'00'</b> Continue <b>B'01'</b> End <b>B'10'</b> Begin <b>B'11'</b> Only
	2-7	<b>RES</b>	Reserved for flags
6		<b>FLAGS</b>	Reserved
7-n		<b>DATA</b>	Optional data

### 5.39.4 Additional Content Description

The content of the OEM Data structured field is further described as follows:

**GROUP** Provides a spanning function; that is, it allows data to be divided into two or more OEM Data structured fields without regard to control sequence boundaries.

**DATA** The OEM Auxiliary Device Query Reply returned for a particular OEM auxiliary device indicates whether the data in the OEM Data structured field contains the manufacturer-defined data stream or a registered value-added version of the manufacturer-defined data stream.

## 5.40 Save/Restore Format

This section describes the Save/Restore outbound/inbound structured field.

Subtopics:

- [5.40.1 Function](#)
- [5.40.2 Format](#)
- [5.40.3 Additional Content Description](#)

### 5.40.1 Function

The Save/Restore Format structured field is used for saving and restoring a Format Parameter Control Block (FPCB) either at the device (a printer) or at the host location. The FPCB consists of parameters such as top, bottom, left, and right margins. Values are initially set by default values at the device and later modified by SCS control codes received in the data stream.

The function allows the host application to rapidly switch from one set of formats to another. For example, the host application may create parameter set 1 and operate with it for awhile and then save it for later use. It can then create parameter set 2, operate with it as required and then save set 2. Following this, the host application can restore set 1 and continue the operation.

### 5.40.2 Format

The following table shows the format of the Save/Restore structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'1034'</b> Identifies this structured field as Save/Restore Format
4		<b>FLAGS</b>	
	0	SAVE/RES	<b>B'0'</b> Save the current Format Parameter Control Block (FPCB). <b>B'1'</b> Restore saved FPCB.
	1	SEC/PRI	<b>B'0'</b> FPCB saved at secondary <b>B'1'</b> FPCB saved at primary
	2-7	RES	Reserved
5-n		<b>FPCB</b>	Contents of the FPCB that is to be saved or restored

### 5.40.3 Additional Content Description

The content of the Save/Restore structured field is further described as follows:

- SAVE/RES** The SAVE/RES flag determines whether the FPCB is to be saved or restored. If SAVE/RES is set to B'0', the information is sent to FPCB storage. If SAVE/RES is set to B'1', the information is sent from FPCB storage to the device.
- SEC/PRI** This flag determines the location where the FPCB is to be stored. If the flag is set to B'0', the FPCB is saved in the device and if the flag is set to B'1', the FPCB is sent to the host location for storage.
- FPCB** The Format Parameter Control Block (FPCB) is a device-dependent control block that contains formatting parameters established by power-on defaults or subsequently received SNA Character String controls such as:
- Set Horizontal Format (SHF)
  - Set Vertical Format (SVF)
  - Set Line Density (SLD)
  - Set Print Density (SPD).

The printer has a current FPCB and one FPCB save area. There are no constraints on the number of save areas at the host location.

The FPCB must contain all of the information necessary to save or restore all format parameters which the printer supports. After a save/restore operation of any given FPCB, either at the printer or the host, the format parameters in effect remain exactly the same as those in effect before the save/restore operation. The save/restore operation itself causes no movement of the current position in the presentation space; that is, the presentation space position is not saved or restored.

At power-on time, the printer initializes the save area to the default values of the device.

---

## 5.41 Select Intelligent Printer Data Stream (IPDS) Mode

This section describes the Select IPDS Mode outbound/inbound structured field.

Subtopics:

- [5.41.1 Function](#)
- [5.41.2 Format](#)

---

### 5.41.1 Function

The Select IPDS Mode structured field selects the IPDS mode when IPDS is supported in 3270 non-SNA. Refer to your product publications for information on how the data is processed when IPDS is selected.

---

## 5.41.2 Format

The following table shows the format of the Select IPDS Mode structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0006'</b> Length of this structured field
2-3	<b>SFID</b>	<b>X'0F83'</b> Identifies this structured field as Select IPDS Mode
4-5	<b>FLAGS</b>	Reserved

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 6.0 Chapter 6. Inbound Structured Fields

Subtopics:

- 6.1 Introduction
- 6.2 Exception/Status
- 6.3 Inbound Text Header
- 6.4 Inbound 3270DS
- 6.5 Recovery Data
- 6.6 Type 1 Text Inbound
- 6.7 Query Replies
- 6.8 Query Reply
- 6.9 Query Reply (Alphanumeric Partitions)
- 6.10 Query Reply (Auxiliary Device)
- 6.11 Query Reply (Begin/End of File)
- 6.12 Query Reply (Character Sets)
- 6.13 Query Reply (Color)
- 6.14 Query Reply (Cooperative Processing Requestor)
- 6.15 Query Reply (Data Chaining)
- 6.16 Query Reply (Data Streams)
- 6.17 Query Reply (DBCS-Asia)
- 6.18 Query Reply (Device Characteristics)
- 6.19 Query Reply (Distributed Data Management)
- 6.20 Query Reply (Document Interchange Architecture)
- 6.21 Query Reply (Extended Drawing Routine)
- 6.22 Query Reply (Field Outlining)
- 6.23 Query Reply (Field Validation)
- 6.24 Query Reply (Format Presentation)
- 6.25 Query Reply (Format Storage Auxiliary Device)
- 6.26 Query Reply (Graphic Color)
- 6.27 Query Reply (Graphic Symbol Sets)
- 6.28 Query Reply (Highlighting)
- 6.29 Query Reply (IBM Auxiliary Device)
- 6.30 Query Reply (Image)
- 6.31 Query Reply (Implicit Partition)
- 6.32 Query Reply (IOCA Auxiliary Device)
- 6.33 Query Reply (Line Type)
- 6.34 Query Reply (MSR Control)
- 6.35 Query Reply (Null)
- 6.36 Query Reply (OEM Auxiliary Device)
- 6.37 Query Reply (Paper Feed Techniques)
- 6.38 Query Reply (Partition Characteristics)
- 6.39 Query Reply (Port)
- 6.40 Query Reply (Procedure)
- 6.41 Query Reply (Product Defined Data Stream)
- 6.42 Query Reply (Reply Modes)
- 6.43 Query Reply (RPQ Names)
- 6.44 Query Reply (Save/Restore Format)
- 6.45 Query Reply (Segment)
- 6.46 Query Reply (Settable Printer Characteristics)
- 6.47 Query Reply (Storage Pools)

- 6.48 Query Reply (Summary)
- 6.49 Query Reply (Text Partitions)
- 6.50 Query Reply (Transparency)
- 6.51 Query Reply (Usable Area)
- 6.52 Query Reply (3270 IPDS)

## 6.1 Introduction

This chapter describes the inbound structured fields, including all of the Query Reply structured fields. These inbound structured fields are structured fields that are transmitted only one way, from the device to the host application. Inbound structured fields are preceded by an AID of X'88', as follows:

(AID) X'88'	Structured Field
----------------	------------------

The inbound structured fields are listed below in the order that they are discussed in this chapter:

ID	Name
0F22	Exception/Status
0FB1	Inbound Text Header
80	Inbound 3270DS
1031	Recovery Data
0FC1	Type 1 Text Inbound
81	Query Reply
84	Query Reply (Alphanumeric Partitions)
99	Query Reply (Auxiliary Device)
9F	Query Reply (Begin/End of File)
85	Query Reply (Character Sets)
86	Query Reply (Color)
AB	Query Reply (Cooperative Processing Requestor)
98	Query Reply (Data Chaining)
A2	Query Reply (Data Streams)
91	Query Reply (DBCS-Asia)
A0	Query Reply (Device Characteristics)
95	Query Reply (Distributed Data Management)
97	Query Reply (Document Interchange Architecture)
B5	Query Reply (Extended Drawing Routine)
8C	Query Reply (Field Outlining)
8A	Query Reply (Field Validation)
90	Query Reply (Format Presentation)
94	Query Reply (Format Storage Auxiliary Device)
B4	Query Reply (Graphic Color)
B6	Query Reply (Graphic Symbol Sets)
87	Query Reply (Highlighting)
9E	Query Reply (IBM Auxiliary Device)
82	Query Reply (Image)
A6	Query Reply (Implicit Partition)
AA	Query Reply (IOCA Auxiliary Device)
B2	Query Reply (Line Type)
8B	Query Reply (MSR Control)
FF	Query Reply (Null)
8F	Query Reply (OEM Auxiliary Device)
A7	Query Reply (Paper Feed Techniques)
8E	Query Reply (Partition Characteristics)
B3	Query Reply (Port)
B1	Query Reply (Procedure)

9C Query Reply (Product Defined Data Stream)  
88 Query Reply (Reply Modes)  
A1 Query Reply (RPQ Names)  
92 Query Reply (Save/Restore Format)  
B0 Query Reply (Segment)  
A9 Query Reply (Settable Printer Characteristics)  
96 Query Reply (Storage Pools)  
80 Query Reply (Summary)  
83 Query Reply (Text Partitions)  
A8 Query Reply (Transparency)  
81 Query Reply (Usable Area)  
9A Query Reply (3270 IPDS).

Structured field type codes not listed are rejected. Unless specifically stated to the contrary, any bits or fields classified as reserved must be checked for zero value; nonzero values are rejected.

---

## 6.2 Exception/Status

This section describes the Exception/Status inbound structured field.

Subtopics:

- [6.2.1 Function](#)
- [6.2.2 Format](#)
- [6.2.3 Additional Content Description](#)
- [6.2.4 Auxiliary Device Exception Condition Self-Defining Parameter](#)
- [6.2.5 Auxiliary Device Status Self-Defining Parameter](#)
- [6.2.6 Failing Format Status Self-Defining Parameter](#)
- [6.2.7 Format Status Self-Defining Parameter](#)
- [6.2.8 Group Status Self-Defining Parameter](#)

---

### 6.2.1 Function

The Exception/Status structured field allows the reporting of exception/status information at the application level.

All parameters shown for the base structured field must be present. The structured field is allowed to carry only one exception/status condition. The Exception/Status structured field accepts one of five self-defining parameters:

- Auxiliary Device Exception Condition
- Auxiliary Device Status
- Failing Format Status
- Format Status
- Group Status.

A length value of X'0000' is invalid.

---

### 6.2.2 Format

The following table shows the format of the Exception/Status structured

field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>SFID</b>	<b>X'0F22'</b> Identifies this structured field as Exception/Status
4		<b>PID</b>	Partition identifier
5		<b>FLAGS</b>	
	0	<b>AVAILSTA</b>	Availability status:  <b>B'0'</b> Status not provided <b>B'1'</b> Status provided
	1-7	<b>RES</b>	Reserved
6		<b>FLAGS</b>	Reserved

---

### 6.2.3 Additional Content Description

The content of the Exception/Status structured field is further described as follows:

**PID** Provides the PID of the partition associated with the reported exception condition or status. When used for reporting an exception condition or status for a direct accessed auxiliary device, the Exception/Status structured field must be preceded by a Destination/Origin structured field. Also, when reporting an exception condition or status of a direct accessed auxiliary device, the PID parameter in the Exception/Status structured field has no meaning and should be ignored. The PID should be set to X'FF'.

**AVAILSTAT** Indicates whether the exception condition being reported requires an availability status when the reported exception clears. This is set to B'0' when the exception code definition does not require a subsequent availability status. It is set to B'1' when the definition of the exception code being carried by the Exception/Status structured field requires an availability status when the condition clears.

---

### 6.2.4 Auxiliary Device Exception Condition Self-Defining Parameter

The Auxiliary Device Exception Condition self-defining parameter reports exception conditions associated with a direct accessed auxiliary device. Although the error codes may have the same value as existing SNA defined codes, the meanings are not necessarily the same.

The following table shows the format of the self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'06'</b> Length of self-defining parameter
1		<b>SDPID</b>	<b>X'01'</b> Identifies this parameter as Auxiliary Device Exception Condition
2-3		<b>RESERVED</b>	Reserved
4-5		<b>EXCODE</b>	The specific exception code. It defines the direct accessed auxiliary exception condition.

**EXCODE** The specific, direct access, auxiliary exception code. The following list defines the valid codes for reporting exception conditions on direct access auxiliary devices:

0801

Invalid/unrecognized DOID in the Destination/Origin structured field. AVAILSTAT must be set to B'0'.

0802

DOID valid, but the auxiliary device is not available because of an intervention required condition (for example, out of paper, power off, or processing code not resident). Available status is sent when the condition clears. AVAILSTAT must be set to B'1'.

0803

Retired.

0804

Buffer overrun.

0805

Insufficient storage. The loading of the formats could not be completed because storage was exhausted.

0806

The format or group name was not specified in the Load Format Storage structured field.

0807

Data error.

084B

Temporary insufficient resource. The application does not have a buffer available or is busy. The device chooses whether to set send status when the condition clears and set AVAILSTAT accordingly.

084C

The auxiliary device data in the transmission exceeds the limit specified in the LIMOUT parameter of the Query Reply for the auxiliary device. AVAILSTAT must be set to B'0'.

---

## 6.2.5 Auxiliary Device Status Self-Defining Parameter

The Auxiliary Device Status self-defining parameter reports a change of status. The sending of certain exception codes that indicate unavailability requires that an available status be sent when the device becomes available.

The following table shows the format of the Auxiliary Device Status self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'04'</b> Length of self-defining parameter
1		<b>SDPID</b>	<b>X'02'</b> Identifies this parameter as Auxiliary Device Status
2-3		<b>STATCODE</b>	Specific Status Code

### STATCODE

Defines the specific status code. The following list defines the valid status codes for auxiliary devices:

0000 Acknowledged. The formats were successfully loaded, and no exception occurred.

0001 Auxiliary device available.

---

## 6.2.6 Failing Format Status Self-Defining Parameter

The Failing Format Status self-defining parameter must be sent whenever the sending of multiple formats in a transmission is supported and a format error is detected.

The following table shows the format of the Failing Format Status self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'22'</b> Length of this self-defining parameter
1		<b>SDPID</b>	<b>X'03'</b> Identifies this parameter as Failing Format
2-17		<b>FGNME</b>	Group Name
18-33		<b>FNME</b>	Format Name

**FGNME** Provides the name of the group containing the failing format.

**FNME** Provides the name of the first format that has an exception condition.

---

## 6.2.7 Format Status Self-Defining Parameter

The Format Status self-defining parameter must be present if, and only if, it has been requested by a previous Load Format Storage structured field (OPRND = Request Summary Status, X'05').

The following table shows the format of the Format Status self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'0C'</b> Length of this self-defining parameter
1		<b>SDPID</b>	<b>X'04'</b> Identifies this self-defining parameter as Format Status
2-3		<b>NGL</b>	Number of groups currently assigned
4-5		<b>NML</b>	Number of formats currently loaded
6-7		<b>NLML</b>	Number of local names used
8-11		<b>STOR</b>	Amount of format storage space available

**NGL** Provides the number of Format Groups that are currently loaded in format storage.

**NML** Provides the number of local formats that are currently loaded in format storage.

**NLML** Provides the number of local formats that are currently loaded in format storage.

**STOR** Provides the amount of remaining format storage space available in K bytes.

---

## 6.2.8 Group Status Self-Defining Parameter

The Group Status self-defining parameter must be present if, and only if, it has been requested by a previous Load Format Storage structured field (OPRND = Request Group Status X'06').

The following table shows the format of the Group Status self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'14'</b> Length of this self-defining parameter
1		<b>SDPID</b>	<b>X'05'</b> Identifies this self-defining parameter as Group Status
2-17		<b>FGNME</b>	Group Name
18-19		<b>NML</b>	Number of formats currently loaded

**FGNME** Provides the name of the group that the NML parameter applies to.

**NML** Provides the number of formats currently loaded in the group specified in FGNME.

---

## 6.3 Inbound Text Header

This section describes the Inbound Text Header structured field.

Subtopics:

- [6.3.1 Function](#)
- [6.3.2 Format](#)
- [6.3.3 Additional Content Description](#)

---

### 6.3.1 Function

The Inbound Text Header structured field is transmitted inbound from a text partition. It can be followed by a Type 1 Text Inbound structured field.

A single Inbound Text Header structured field is transmitted inbound from a text partition (A-MODE = X'2') under the following conditions:

- In reply to a Read Modified command when the device is in a retry state and INPID identifies the text partition
- For an operator enter action when the active partition is a text partition.

---

### 6.3.2 Format

The following table shows the format of the Inbound Text Header structured field:

---

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0016'</b> Length of this structured field
2-3	<b>SFID</b>	<b>X'0FB1'</b> Identifies this structured field as Inbound Text Header
4	<b>PID</b>	Partition identifier
5	<b>AID</b>	Attention identifier
6	<b>RES</b>	Reserved
7	<b>RES</b>	Reserved
8	<b>RES</b>	Reserved
9	<b>LVL</b>	Cursor level
10-11	<b>CRO</b>	Cursor row offset
12-13	<b>CC</b>	Cursor column offset
14-15	<b>RW</b>	Row offset of window origin
16-17	<b>CW</b>	Column offset of window origin
18-19	<b>HW</b>	Window height
20-21	<b>WW</b>	Window width

---

### 6.3.3 Additional Content Description

The content of the Inbound Text Header structured field is further described as follows:

**LVL** The LVL parameter specifies the level of the element at the presentation space position CRO,CC that is identified. (The first element is at level 1.)

When the element level can be determined by the device, it should return a value for LVL in the range 1 through 254. If the device does not identify the level explicitly, it returns value X'FF' for LVL, meaning all elements at position CRO,CC.

It is not required that a device have the capability of distinguishing between multiple elements at the same position. That is, a device could always return LVL = X'FF' for a non-empty presentation position.

**CRO, CC** The CRO,CC parameters specify the presentation space position of the cursor. These values can range from 0 through 65534 (that is, the value X'FFFF' is not returned inbound).

If position CRO,CC is empty, LVL is X'00'.

**RW, CW** RW and CW give the origin of the window on presentation space.

**HW, WW** HW and WW give the height and width of this window. In a text partition, the operator can change the size of this window from the initial values established by the Create Partition structured field.

---

## 6.4 Inbound 3270DS

This section describes the Inbound 3270DS structured field.

Subtopics:

- [6.4.1 Function](#)
- [6.4.2 Format](#)

---

### 6.4.1 Function

The Inbound 3270DS structured field is used to transmit 3270 orders and data inbound from a named alphanumeric partition.

This structured field is transmitted inbound from an alphanumeric partition (Create Partition Structured Field - A-MODE = B'0000' or B'0001') under the following conditions:

- In reply to a Read Modified, Read Modified All (SNA only), or Read Buffer command when the inbound partition has a nonzero PID, and the device is in Retry state
- In reply to a Read Partition structured field, type X'F6', X'6E' or X'F2' addressed to a partition with a nonzero PID
- For an operator enter action when the Active partition has a nonzero PID.

Orders and data from partition zero (whether implicitly or explicitly created) are transmitted in 3270 format, not in this structured field.

---

### 6.4.2 Format

The following table shows the format of the Inbound 3270DS structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'80'</b> Identifies this structured field as Inbound 3270DS
3	<b>PID</b>	Partition identifier

4	<b>AID</b>	Attention identifier
5-6	<b>CURSOR</b>	Cursor address
7-n	<b>DATA</b>	Orders and data

## 6.5 Recovery Data

This section describes the Recovery Data inbound structured field.

Subtopics:

- [6.5.1 Function](#)
- [6.5.2 Format](#)
- [6.5.3 Additional Content Description](#)

### 6.5.1 Function

The Recovery Data structured field is sent from the SLU to the PLU in response to a Request Recovery Data structured field. It contains the recovery data needed by the PLU to recover from the error.

### 6.5.2 Format

The following table shows the format of the Recovery Data structured field:

Byte	Bit	Content	Content Description
0 - 1		<b>L</b>	<b>X'0031'</b> Length of this structured field
2 - 3		<b>SFID</b>	<b>X'1031'</b> Identifies this structured field as Recovery Data
4		<b>RES</b>	Reserved
5		<b>FLAGS</b>	
	0	VFORMAT	<b>B'0'</b> Vertical format not needed on Restart <b>B'1'</b> Vertical format needed on Restart
	1	HFORMAT	<b>B'0'</b> Horizontal format not needed on Restart <b>B'1'</b> Horizontal format needed on Restart
	2 - 7	RES	Reserved

6		<b>SLD</b>	SLD -- Set line density parameter in effect at the checkpoint
7		<b>CHAR. SET</b>	Character set parameter of Set Attribute control in effect at the checkpoint
8 - 11		<b>VERTICAL</b>	Byte offset from Checkpoint Interval structured field to the Set Vertical Format control in effect for the checkpoint
12 - 13		<b>V-OFFSET</b>	Byte offset within the string control byte string or the SVF character
14 - 15		<b>V-SEQUEN</b>	ERU sequence number
16 - 17		<b>V-LENGTH</b>	Length of the SVF character string required for restart
18 - 19		<b>SPD</b>	Set Primary Density parameter in effect at the checkpoint
20 - 23		<b>HORIZON</b>	Same as VERTICAL for SHF
24 - 25		<b>H-OFFSET</b>	Same as V-OFFSET for SHF
26 - 27		<b>H-SEQUEN</b>	ESame as V-SEQUENCE for SHF
28 - 29		<b>H-LENGTH</b>	Same as V-LENGTH for SHF
30		<b>COLOR</b>	Color parameter of SA in effect at the checkpoint
31		<b>HILITE</b>	Highlight parameter of SA in effect at the checkpoint
32 - 33		<b>PAGES</b>	Number of pages printed since the checkpoint
34 - 35		<b>LINES</b>	Number of lines printed on the page with the error
36 - 39		<b>CHECKPOI</b>	TByte offset from Set Checkpoint Interval structured field to the first character after the code point or character that caused an eject to the checkpointed page
40 - 41		<b>C-OFFSET</b>	Byte offset within the String Control Byte string or structured field of the checkpointed character
42 - 43		<b>C-SEQUEN</b>	ERU sequence number of the RU containing the checkpoint character
44 - 45		<b>C-SEQOFF</b>	Byte offset within the RU of the checkpointed character
46 - 47		<b>C-SCSOFF</b>	Byte offset within the parameterized SCS control code (for example, TRN) of the checkpointed character.
48		<b>PRIME</b>	Prime compression character

---

### 6.5.3 Additional Content Description

**FLAGS** Bit 0 of the FLAG field indicates whether the printer requires the SVF format to be present in the Restart structured field. If bit 0 is 1, the primary must use the VERTICAL field and V-OFFSET to locate the SVF format and resend it in the Restart structured field chain. If bit 0 is zero, the primary does not have to send the SVF format as the printer can assure that the formats are still set properly. Bit 1 of the FLAGS field is the same as bit 0, but for SHF.

**SLD** SLD is a 1-byte parameter field of the SLD in effect at the checkpoint. The default value (X'00') is used if SLD is not supported or has changed since the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.

**CHAR. SET**

Character Set is the 1-byte parameter field of the SA value for character set in effect at the checkpoint. The default value (X'00') is used if SA for character set is not supported or has not changed since the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.

**VERTICAL**

VERTICAL is a 4-byte counter that indicates the byte offset from the Set Checkpoint Interval structured field to the SVF code point in effect for this checkpoint. If no SVF code has been received, this field is set to zero. If the SVF is inside a parameterized string, the counter points to the control code of the parameters (for example, the immediately preceding SCB control code).

**V-OFFSET**

V-OFFSET is a 2-byte counter that indicates the position within an SCB string or structured field of the SVF code. It points to the uncompact/uncompressed position of the SVF. This field is set to zero if the vertical field points to the actual SWF code or if the SVF character is not within a SCB string or structured field.

**V-SEQUENCE**

V-SEQUENCE contains the RU sequence number of the RU that contains the SVF control code in effect at the checkpoint.

**V-LENGTH**

V-LENGTH contains the length of the SVF parameter string that must be returned in the Restart structured field. It includes the SVF control code and all parameters following.

**SPD**

SPD is the 2-byte parameter field of the SPD in effect at the checkpoint. The default value (X'0000') is used if SPD is not supported or has not changed since the checkpoint. The default value (X'0000') indicates that the parameter is not to be reinitialized at restart.

**HORIZON**

This is a 4-byte binary counter. It indicates that the byte offset

from the Set Checkpoint Interval structured field to the SHF code point is in effect for this checkpoint. If no SHF has been received, this field is set to zero. If the SHF is inside a parameterized string, the counter points to the control code of the parameter (for example, the immediately preceding SCB control code).

#### **H-OFFSET**

H-OFFSET is a 2-byte binary counter that indicates the position within an SCB string or structured field of the SHF code. It points to the uncompact/uncompressed position of the SHF. This field is set to zero if the Horizontal Field points to the actual SHF code or if the SHF character is not within an SCB string or structured field.

#### **H-SEQUENCE**

H-SEQUENCE contains the RU sequence number of the RU that contains the SHF control code in effect at the checkpoint.

#### **H-LENGTH**

H-LENGTH contains the length of the SHF parameter string that must be returned in the Restart structured field. It includes the SHF control code and all parameters following.

**COLOR** COLOR is the 1-byte parameter field of the SA (set attribute) value for color in effect at the checkpoint. The default value (X'00') is used if SA for color is not supported or has not changed since the checkpoint. The default value (X'00') indicates the parameter is not to be initialized at restart.

**HILITE** This is the 1-byte parameter field of the SA value for highlighting in effect at the checkpoint. The default SA value (X'00') is not supported or has not changed since the checkpoint. The default value (X'00') indicates the parameter is not to be initialized at restart.

**PAGES** PAGES indicates the number of pages that the SLU has printed since the checkpoint that is being sent to the PLU was taken.

**LINES** LINES indicates the number of lines that the SLU has printed since the beginning of the page on which the error was detected.

#### **CHECKPOINT**

CHECKPOINT is a 4-byte binary counter that indicates a certain number of FM data bytes. These bytes are from the Set Checkpoint Interval structured field to the first code point after the one causing a page eject to the page for which a checkpoint is to be taken. That is, the counter counts all FM data bytes in the RUs following the latest Set Checkpoint Interval structured field. It then points to the first code point processed after the printer ejected to the top of the page for which the interval count requires a checkpoint. If the eject was caused by a parameterized string, the counter points to the control code of the parameters (for example, the immediately preceding SCB control code).

#### **C-OFFSET**

C-OFFSET is a 2-byte binary counter that indicates the position within a compression/compaction SCB string or structured field of the actual checkpoint character. It points to the uncompact/uncompressed position of the character, not the

compressed/compacted position. If the character is within a parameterized SCS control code, this is the offset to the SCS control code. This field is zero if the checkpoint field points to the exact checkpointed character or if the checkpointer is not within an SCB string or structured field.

**C-SEQ** C-SEQ contains the RU sequence number of the RU containing the checkpointed character.

**C-SEQOFF**

C-SEQOFF contains the byte offset within the RU of the checkpointed character.

**C-SCSOFF**

This is a 2-byte binary counter that indicates the position within a parameterized SCS code of the checkpointed character. This field is set to zero if the checkpointed character is not within a parameterized SCS code.

**PRIME** PRIME is the prime compression character in effect at the time of checkpoint.

---

## 6.6 Type 1 Text Inbound

This section describes the Type 1 Text Inbound structured field.

Subtopics:

- [6.6.1 Function](#)
- [6.6.2 Format](#)
- [6.6.3 Additional Content Description](#)

---

### 6.6.1 Function

The Type 1 Text Inbound structured field transmits data from a text partition.

The structured field is transmitted inbound from a text partition (A-MODE = X'2') following an Inbound Text Header structured field if both of the following are true:

- The AID corresponds to a read modified operation.
- The MDT is set because the operator modified data.

The Type 1 Text Inbound structured field is not transmitted:

- For a "short read" (PA key or Clear Partition)
- If the MDT is not set.

---

### 6.6.2 Format

The following table shows the format of the Type 1 Text Inbound structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2-3	<b>SFID</b>	<b>X'0FC1'</b> Identifies this structured field as Type 1 Text Inbound
4	<b>PID</b>	Partition identifier
5-6	<b>RES</b>	Reserved
7-n	<b>DATA</b>	Text data

---

### 6.6.3 Additional Content Description

The content of the Type 1 Text Inbound structured field is further described as follows:

**DATA** The text DATA consists of the entire contents of the text buffer, with an embedded insert cursor control to identify the cursor position. The data stream position of insert cursor should be interpreted in combination with the LVL parameter in the Inbound Text Header, as follows:

- If LVL is X'01' through X'FE', the insert cursor control immediately precedes the element at position [CRO, CC, LVL].
- If LVL = X'FF' (that is, all elements at [CRO,CC]), the insert cursor control immediately precedes the first element in the data stream that is at a (row, column) at or beyond (CRO,CC).
- If LVL = X'00' (that is, [CRO,CC] is empty), the insert cursor control immediately precedes the first element in the data stream (if any) that is at a (row, column) beyond (CRO,CC). If there is no such element, insert cursor is at the end of the data stream.

---

## 6.7 Query Replies

The Query Reply structured field is used to report to the host application the functions supported by the 3270 data stream devices and information about these functions.

All Query Reply structured fields have a value of X'81' in byte 2 that identifies the structured field as a Query Reply. The QCODE parameter (byte 3) is a unique value assigned to each Query Reply.

The Read Partition structured field is used for soliciting Query Replies when the Type parameter has a value of either X'02' (Query) or X'03'

(Query List). The PID parameter (byte 3) has a value of X'FF' (PID not applicable). If the Type parameter value is X'03' (Query List), the Read Partition structured field contains an additional parameter, REQ TYP (Request Type), bits 0-1 of byte 5 and, optionally, a list of QCODES starting at byte 6. The Request Type parameter is used to indicate the type of Query List operation:

- B'00' = QCODE List (QCODES are listed for the Query Replies requested)
- B'01' = Equivalent (with or without a QCODE List)
- B'10' = All.

The Query Replies are returned in response to a Query or Query List as indicated in Table 6-1 in topic 6.7.3.

Subtopics:

- 6.7.1 Query
- 6.7.2 Query List
- 6.7.3 Query Response Table

## 6.7.1 Query

A Read Partition structured field with a Type parameter (byte 4) value of X'02' is defined as a Query. See Table 6-1 in topic 6.7.3 for the Query Replies that a 3270 data stream device or workstation supports that must be returned in reply to a Query. Those not indicated are not returned in reply to a Query.

## 6.7.2 Query List

A Read Partition structured field with a Type parameter (byte 5) value of X'03' is defined as a Query List. There are three versions of Query List. The version is designated by the Request Type parameter (bits 0-1, byte 5) value, as follows:

Version	Request Type Parameter	Description
QCODE List	B'00'	<p>Contains a list of one or more Query Reply QCODES. The 3270 data stream device or workstation returns all the requested Query Replies (QCODES listed) that are supported. If none of the requested Query Replies are supported, a Null Query Reply is returned. There is no requirement as to the order of the QCODES in the Read Partition (Query List) structured field or the order that the requested Query Replies are returned by the 3270 data stream device or workstation.</p> <p>It is not invalid for a particular QCODE to appear more than once in the list. However, regardless of how many times it appears, the 3270 device or workstation does not return duplicate Query Replies.</p>

EQUIVALEN	B'01'	Requests the 3270 device or workstation to return the same Query Replies that would be returned in reply to a Query. Optionally, a list of QCODES can also be included. Requests for the same QCODE are made within the list, or a QCODE within the list is returned for the Query Equivalent. No duplicate Query Replies are returned by the 3270 data stream device or workstation.
ALL	B'10'	Requests the 3270 data stream device or workstation to return all the Query Replies supported. The Query List = All can contain a QCODE list. However, the QCODE list is ignored by the 3270 data stream device or workstation.

Therefore, although duplicate Query Replies may be requested by the Query List = QCODE List, Equivalent, or All, the 3270 data stream device or workstation does not return duplicate Query Replies. There are cases where a single request (for example, QCODE in Query List) can result in multiple Query Replies being returned. For example, a 3270 data stream workstation could support two OEM auxiliary devices such as plotters. A Query List request for the OEM Auxiliary Device Query Reply would result in two OEM Auxiliary Device Query Replies being returned, one for each device. However, although the two Query Replies are of the same type (OEM Auxiliary Device), the contents of the Query Replies differ (at least in the DOID parameter) and hence are not duplicate Query Replies.

All 3270 data stream devices or workstations must support the Summary Query Reply, QCODE = X'80'. This Query Reply is returned in reply to a Query or Query List (QCODE List, Equivalent, or All). The Summary Query Reply provides a list of the QCODES of all the Query Replies supported by the 3270 data stream device or workstation. (The QCODE for the Summary Query Reply itself is also included in the list.) The Summary Query Reply provides the host with the only indication of support of functions where the associated Query Reply is returned in reply to a Query List = QCODE List or All.

### 6.7.3 Query Response Table

Table 6-1 briefly describes each Query Reply.

Table 6-1. Query Responses				
Query Reply	Returned in Response to a Read Partition:			
	Query	Query List		
		List	Equivalen	All
Alphanumeric Partitions	Yes	X'84'	Yes	Yes
Auxiliary Device	Yes	X'99'	Yes	Yes
Begin/End of File	No	X'9F'	No	Yes
Character Sets	Yes	X'85'	Yes	Yes
Color	Yes	X'86'	Yes	Yes

Cooperative Proc. Requestor	Yes	X'AB'	Yes	Yes
Data Chaining	Yes	X'98'	Yes	Yes
Data Streams	Yes	X'A2'	Yes	Yes
DBCS-Asia	Yes	X'91'	Yes	Yes
Device Characteristics	Yes	X'A0'	Yes	Yes
Distributed Data Management	Yes	X'95'	Yes	Yes
Document Interchange Architecture	Yes	X'97'	Yes	Yes
Extended Drawing Routine	No	X'B5'	No	Yes
Field Outlining	Yes	X'8C'	Yes	Yes
Field Validation	Yes	X'8A'	Yes	Yes
Format Presentation	Yes	X'90'	Yes	Yes
Format Storage Auxiliary Device	Yes	X'94'	Yes	Yes
Graphic Color	No	X'B4'	No	Yes
Graphic Symbol Sets	No	X'B6'	No	Yes
Highlighting	Yes	X'87'	Yes	Yes
IBM Auxiliary Device	Yes	X'9E'	Yes	Yes
Image	No	X'82'	No	Yes
Implicit Partition	Yes	X'A6'	Yes	Yes
IOCA Auxiliary Device	No	X'AA'	No	Yes
Line Type	No	X'B2'	No	Yes
MSR Control	Yes	X'8B'	Yes	Yes
Null	No	X'FF'	No	No
OEM Auxiliary Device	Yes	X'8F'	Yes	Yes
Paper Feed Techniques	Yes	X'A7'	Yes	Yes
Partition Characteristics	No	X'8E'	No	Yes
Port	No	X'B3'	No	Yes
Procedure	No	X'B1'	No	Yes
Product Defined Data Stream	No	X'9C'	No	Yes
Reply Modes	Yes	X'88'	Yes	Yes
RPQ Names	Yes	X'A1'	Yes	Yes
Save/Restore Format	No	X'92'	No	Yes
Segment	No	X'B0'	No	Yes
Settable Printer Characteristics	No	X'A9'	No	Yes
Storage Pools	No	X'96'	No	Yes

Summary	Yes	X'80'	Yes	Yes
Text Partitions	Yes	X'83'	Yes	Yes
Transparency	No	X'A8'	No	Yes
Usable Area	Yes	X'81'	Yes	Yes
3270 IPDS	Yes	X'9A'	Yes	Yes

## 6.8 Query Reply

This section describes the Query Reply structured field.

Subtopics:

- [6.8.1 Function](#)
- [6.8.2 Format](#)

### 6.8.1 Function

The Query Reply structured field is a response to a Query request. Query requests are contained in the Read Partition structured field.

In reply to a Query, the device transmits inbound a set of structured fields that describe the device features. Only the Query Replies that reflect the functions and features supported by the device are transmitted inbound.

Each Query Reply has an ID of X'81nn', where nn is the QCODE. When a Query Reply is used in the 3270 data stream, it is preceded by an AID of X'88'. If the structured field is one of a set of Query Reply structured fields, only the first is preceded by an AID of X'88'.

### 6.8.2 Format

The following table shows the format of the Query Reply structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	The code identifying the feature being described
4-n	<b>PLIST</b>	A variable length parameter list. See the specific Query Reply in this chapter for the parameter list for each QCODE.

---

## 6.9 Query Reply (Alphanumeric Partitions)

This section describes the Query Reply (Alphanumeric Partitions) structured field.

Subtopics:

- [6.9.1 Function](#)
- [6.9.2 Format](#)
- [6.9.3 Additional Content Description](#)
- [6.9.4 Buffer Allocation Self-Defining Parameter](#)

---

### 6.9.1 Function

The Query Reply (Alphanumeric Partitions) structured field transmits the maximum number of partitions that can be supported on this device and the device's ability to support partition related functions.

When this function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE = X'84', Equivalent, or All).

---

### 6.9.2 Format

The following table shows the format of the Query Reply (Alphanumeric Partitions) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'84'</b> Identifies this Query Reply as Alphanumeric Partitions
4		<b>NA</b>	Max number of alphanumeric partitions
5-6		<b>M</b>	Total available partition storage
7		<b>FLAGS</b>	
	0	VERTWIN	<b>B'0'</b> Vertical scrolling not supported <b>B'1'</b> Vertical scrolling supported
	1	HORWIN	<b>B'0'</b> Horizontal scrolling not supported <b>B'1'</b> Horizontal scrolling supported
	2	RES	Reserved

	3	APA FLG	<b>B'0'</b> All Points addressability not supported <b>B'1'</b> All Points addressability supported
	4	PROT	<b>B'0'</b> Partition protection not supported <b>B'1'</b> Partition protection supported
	5	LCOPY	<b>B'0'</b> Presentation space local copy not supported <b>B'1'</b> Presentation space local copy supported
	6	MODPART	<b>B'0'</b> Modify Partition not supported <b>B'1'</b> Modify Partition supported
	7	RES	Reserved

### 6.9.3 Additional Content Description

The content of the Query Reply (Alphanumeric Partitions) structured field is further described as follows:

**NA** NA specifies the number of alphanumeric partitions supported.

If NA>0, the alphanumeric partitions can be named 0 through N-1, where N is the total number of partitions supported.

If NA=0, only one alphanumeric partition can be created with Create Partition. The identifier of this partition must be zero. All outbound and inbound operations to this partition must be through 3270 commands (that is, EW, EWA, EAU, RM, RMA [SNA only], RB).

**M** M defines the total number of bytes of storage available for creating alphanumeric partitions. The value of X'FFFF' is reserved.

Refer to CO, RO, CM, and FO in the Buffer Allocation self-defining parameter for allocation of M.

**FLAGS** The flag bits specify device support of partition related functions and are associated with parameters in the Create Partition structured field. A zero value indicates that the associated function is not supported.

**VERTWIN/HORWIN**

Indicates whether the device supports vertical and horizontal scrolling. Support of scrolling implies the following:

- Support for Create Partition with presentation space

height greater than viewport height, or width greater than viewport width

- Support for Set Window Origin with a change of the column (row)
- Possible support of local window movement by the operator; thus, the operator can move the window from the host-specified position.

**APA** Indicates that the Create Partition structured field parameters can be specified in terms of addressable points.

**LCOPY** Indicates whether the device supports the Presentation Space Local Copy function.

**MODPART** Indicates whether the device supports the Modify Partition structured field. If the vertical scroll flag is also set, Modify Partition can be used to change the window row.

---

## 6.9.4 Buffer Allocation Self-Defining Parameter

The Buffer Allocation self-defining parameter must be included as part of the Alphanumeric Partitions Query Reply unless bytes 3-6 (RO, CO, FO) are all zeros and byte 2 (CM) = 1. The default value for parameters not sent is zero.

A partition size, as defined by (HxW) of the Create Partition structured field, requires storage of:

$CM(H \times W) + (RO \times H) + (CO \times W) + FO$  bytes.

The total storage for all created partitions must not exceed M. A Create Partition structured field that causes the total storage used to exceed M will be rejected.

The following table shows the format of the Buffer Allocation self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	Length of this structured field
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Buffer Allocation Parameters
2	<b>CM</b>	Character multiplier - number of storage bytes required for each position defined by (HxW) of the Create Partition structured field
3	<b>RO</b>	Row overhead - bytes of storage required per partition row
4	<b>CO</b>	Column overhead - bytes of storage required per partition column

5-6	FO	Fixed overhead - fixed storage bytes required per partition in addition to RO and CO
-----	----	--

## 6.10 Query Reply (Auxiliary Device)

This section describes the Query Reply (Auxiliary Device) structured field.

Subtopics:

- [6.10.1 Function](#)
- [6.10.2 Format](#)

### 6.10.1 Function

The Query Reply (Auxiliary Device) structured field indicates direct access support of one or more auxiliary devices.

When a 3270 Data Stream Work Station function is supported (that is, support of one or more auxiliary devices) this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'99', Equivalent, or All) or to a Query.

This Query Reply indicates support of the following:

- Destination/Origin structured field
- Read Partition structured field specifying Query List
- One or more auxiliary devices.

### 6.10.2 Format

The following table shows the format of the Query Reply (Auxiliary Device) structured field:

Byte	Content	Content Description
0-1	L	X'0006' Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'99' Identifies this Query Reply as Auxiliary Device
4-5	FLAGS	Reserved

## 6.11 Query Reply (Begin/End of File)

This section describes the Query Reply (Begin/End of File) structured field.

Subtopics:

- [6.11.1 Function](#)
- [6.11.2 Format](#)

---

## 6.11.1 Function

This Query Reply (Begin/End of File) structured field indicates that the SLU supports receiving Beginning-of-file and End-of-file from the PLU to delineate a file or job. The Begin/End of File Query Reply must always be sent inbound in reply to a Read Partition Query List (QCODE List=X'9F', or All) by any device which supports Beginning of File and End of File indication to delineate a file or job.

---

## 6.11.2 Format

The following table shows the format of the Query Reply (Begin/End of File) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0005'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'9F'</b> Identifies this Query Reply as Begin/End of File
4-5	<b>FLAGS</b>	Reserved; must be set to zeros

---

## 6.12 Query Reply (Character Sets)

This section describes the Query Reply (Character Sets) structured field.

Subtopics:

- [6.12.1 Function](#)
- [6.12.2 Requirements](#)
- [6.12.3 Character Sets Query Reply Base](#)
- [6.12.4 Character Set Descriptors](#)

---

### 6.12.1 Function

The Query Reply (Character Sets) structured fields transmits information about the following:

- Each character set supported
- The ability of the device to support the Load PS structured field and the GE character.

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

This form of the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'85', Equivalent, or All).

---

## 6.12.2 Requirements

This Query Reply consists of at least two parts: the base and a descriptor. Each is discussed in the following sections.

The GF and CF flags in the base must be set to B'1'. The Coded Graphic Character Set Global Identifier (CGCSGID) and Coded Character Set Identifier (CCSID) fields must be present.

---

## 6.12.3 Character Sets Query Reply Base

This section describes the Character Sets Query Reply Base.

Subtopics:

- 6.12.3.1 Function
- 6.12.3.2 Format
- 6.12.3.3 Additional Content Description

---

### 6.12.3.1 Function

This portion of the Character Sets Query Reply tells the host application which loadable and/or nonloadable character sets are available at the device. The base is always required and has requirements of its own.

---

### 6.12.3.2 Format

The following table shows the format of the Character Sets Query Reply Base:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'85'</b> Identifies this Query Reply as Character Sets Query Reply
4		<b>FLAGS</b>	
	0	ALT	<b>B'0'</b> Graphic Escape not supported <b>B'1'</b> Graphic Escape supported
	1	MULTID	<b>B'0'</b> Multiple LCIDs are not supported <b>B'1'</b> Multiple LCIDs are supported
	2	LOADABLE	<b>B'0'</b> Load PSSF is not supported <b>B'1'</b> Load PSSF is supported
	3	EXT	<b>B'0'</b> Load PS EXTENDED is not supported <b>B'1'</b> Load PS EXTENDED is supported
	4	MS	<b>B'0'</b> Only one character slot size is supported <b>B'1'</b> More than one size of character slot is supported
	5	CH2	<b>B'0'</b> Two-byte coded character sets are not supported <b>B'1'</b> Two-byte coded character sets are supported
	6	GF	<b>B'0'</b> CGCSGID is not present <b>B'1'</b> CGCSGID is present
	7	RES	Reserved
5		<b>FLAGS</b>	
	0	RES	Reserved
	1	PSCS	Programmed Symbols Character Slot
			<b>B'0'</b> Load PS slot size match required <b>B'1'</b> Load PS slot size match not required

	2	RES	Reserved
	3	CF	<b>B'0'</b> CCSID not present <b>B'1'</b> CCSID present
	4-7	RES	Reserved
6		<b>SDW</b>	Default character slot width
7		<b>SDH</b>	Default character slot height
8-11		<b>FORM</b>	Supported Load PS format types, bit encoded. Bit i = 1 means type i is supported.
12		<b>DL</b>	Length of each descriptor

### 6.12.3.3 Additional Content Description

The content of the Character Sets Query Reply Base is further described as follows:

- ALT** Indicates that the device can support an alternate character set through the use of the GE control character.
- MULTID** Indicates that the device supports multiple graphic symbol sets with the same LCID (see the MULTID parameter of the Load PS structured field on page [5.13.2](#)).
- LOADABLE** Indicates that the Load Programmed Symbol Set structured field is supported. In some implementations, the Load PS structured field may be supported while one or more of the attached devices do not support any loadable character sets (for example, a controller with multiple devices attached). Consequently, the Character Sets Query Reply associated with a particular device may indicate support of the Load PS structured field, but only nonloadable character sets are supported. Therefore, to indicate support of loadable character sets, the Character Sets Query Reply for a device must indicate the support of both the Load PS structured field and support of one or more loadable character sets.
- EXT** Indicates that the device supports the Load PS extensions. Thus EXT can be set only if Loadable is set.
- MS** Indicates whether or not the device supports a character set related character slot size. This parameter can take one of two of the following values:
- B'0'** The character slot size for all character sets is given by the parameters SDW and SDH. If any values are specified in the character set descriptors for SW or SH, they are ignored.
- B'1'** Each character set has its own character slot size, given by the parameters SW and SH in the descriptor for that

character set. The parameters SW and SH are present in all descriptors. However, if the value of either SW or SH for a given character set is zero, the values given by SDW and SDH are used.

If the device supports non-matrix characters (as indicated in byte 5, bit 1 of the Usable Area Query Reply), bytes 6 and 7 (SDW and SDH) are not applicable and must be set to zero.

**CH2** Indicates whether or not 2-byte character sets are supported. In a 2-byte character set, the characters are represented by 2-byte codes in the data stream. This parameter can take one of two of the following values:

B'0' Two-byte character sets are not supported and the SUBSN parameters are not present in the descriptors.

B'1' Two-byte character sets are supported and the SUBSN parameters are present in the descriptors.

**GF** Indicates that the device returns Coded Graphic Character Set Global identifiers in this Query Reply. This parameter remains to accommodate some existing implementations that support the Character Sets Query Reply but not CGCSGID. All new implementations must have the CGCSGID present and have GF set to B'1'.

**PSCS** Indicates whether or not the character size (LW/LH) specified in the Load PS structured field must exactly match the character slot size (SDW/SDH or SW/SH) specified in the Character Sets Query Reply. This parameter can take one of two of the following values:

B'0' A Load PS structured field specifying anything other than an exact match is rejected.

B'1' A Load PS structured field specifying LW equal to or less than SW/SDW and an LH equal to or less than SH/SDH is accepted.

Regardless of the PSCS flag setting, if either or both the width or height of the character size specified in the Load PS Sets structured field exceeds the corresponding dimension of the character slot specified in the Character Sets Query Reply, the Load PS structured field is rejected. Support of a Load PS character size smaller than the character slot size (bit = B'1') is an implementation/requirement option. The PSCS flag has no meaning for an implementation that does not support Load PS Loadable Symbol Sets. The flag is set to B'0'.

When supported, a valid Load PS structured field specifying a character size smaller (in either or both dimensions) loads the character starting at the top left of the designated character slot. If the width (LW) is less than the width of the character slot, then the remaining dots in each row of the character slot are cleared to zero. The height (LH) is processed in a similar manner when it is less than the height

of the character slot. The effect of this processing is as if the character slot were cleared to zero immediately before the character that was transmitted (in the Load PS Sets structured field) was loaded into the character slot.

**CF** Indicates that the device returns the Coded Character Set Identifier (CCSID), which identifies the character set, code page, and encoding scheme.

**SDW** and **SDH** Define the default size of all the characters in the character set.

**FORM** In this 32-bit field, each bit corresponds to a format type that can be specified in the Load PS structured field. Thus, if bit *i* is set, the device supports format type *i*.

Currently, Load PS format types 1 - 6 and 8 are defined. The remaining values are reserved. This means that bit 0 and bits 7 through 32 (with the exception of bit 8) are reserved. The remaining bits are set depending on the formats supported.

**DL** Defines the length of each descriptor.

---

## 6.12.4 Character Set Descriptors

This section describes the Character Set Query Reply descriptors.

Subtopics:

- [6.12.4.1 Function](#)
- [6.12.4.2 Format](#)
- [6.12.4.3 Additional Content Description](#)

---

### 6.12.4.1 Function

The Character Set Query Reply descriptors define the characteristics of the character sets. Descriptors follow the base portion of a Character Sets Query Reply. At least one descriptor is required for each Character Sets Query Reply. Each descriptor's length is determined by the DL parameter; the number of bytes assigned to this parameter is the length of the descriptor.

---

### 6.12.4.2 Format

The following table shows the format of the Character Set Query Reply descriptors:

Byte	Bit	Content	Content Description
1		<b>SET</b>	Device Specific Character Set ID (PS store

			No.)
2		<b>FLAGS</b>	
	0	LOAD	<b>B'0'</b> Non-loadable character set <b>B'1'</b> Loadable character set
	1	TRIPLE	<b>B'0'</b> Single-plane character set <b>B'1'</b> Triple-plane character set
	2	CHAR	<b>B'0'</b> Single-Byte coded character set <b>B'1'</b> Double-Byte coded character set
	3	CB	<b>B'0'</b> LCID compare <b>B'1'</b> No LCID compare
	4-7	RES	Reserved
3		<b>LCID</b>	Local character set ID (alias)
4		<b>SW</b>	Width of the character slots in this character set. Present only if MS flag set to B'1'.
5		<b>SH</b>	Height of the character slots in this character set. Present only if MS flag set to B'1'.
6		<b>SUBSN(1)</b>	Starting subsection. Present only if CH2 flag = 1.
7		<b>SUBSN(1)</b>	Ending subsection. Present only if CH2 flag = 1.
8-11		<b>CGCSGID ( )</b>	Coded Graphic Character Set Identifier.
12-13		<b>CCSID</b>	Coded Character Set Identifier.
<b>Note:</b>			
(1) The position of the CGCSGID and SUBSN parameters may vary. For example, if SUBSNs and SW/SH were not present, the CGCSGID parameter would start at byte 4.			

### 6.12.4.3 Additional Content Description

The content of the Character Sets Query Reply descriptors is further described as follows:

**SET** Defines the device-specific character set ID. For a loadable character set, this value is specified in the RWS parameter of the Load PS structured field. It also defines the key that the operator can use to select this character set. The character set with device specific ID equal to X'01' is defined as the alternate character page.

If multiple base character sets are supported, the device returns multiple descriptors with SET = X'00' (one descriptor for each supported base character set). The first such descriptor defines the device default character set.

The BASE parameter in the Create Partition structured field or the base parameters in the Select Base Character Set SDP of the Set Partitions Characteristics structured field are used to select the appropriate base character set for a partition by indexing down the base character set descriptors in the order returned in this Query Reply.

The device default character set is the one associated with the implicit partition and with the keyboard.

**LOAD** Indicates whether or not this character set is loadable.

**TRIPLE** Indicates whether or not this device set has three planes: red, blue, and green.

**CHAR** Indicates whether the character set is a 1-byte or a 2-byte character set. It can take one of two values:

B'0' The characters are represented by 1-byte codes in the data stream. The character slot maps into a display cell of the size equal to AW x AH (AW/AH are defined in the Usable Area Query Reply).

B'1' The characters are represented by 2-byte codes in the data stream. The character slot maps into a display cell of size equal to 2AW x AH.

**CB** Indicates whether this character set can be compared for copy. See the section "Load Programmed Symbols (Load PS)" in topic 5.13.

**LCID** Identifies the Local ID currently connected to this character set; it does not necessarily uniquely identify the contents. A value of X'FF' indicates that this character set is free. (It cannot be accessed using an LCID in an SA, SFE, or MF order, and it cannot be selected by the operator using the PS [A-F] selection keys.) If the set is loadable, X'FF' also means that the set is available for local copy. The Load PS structured field allows the LCID (alias) to be connected to a loadable character set. This is the data stream value used in the SA, SFE, or MF orders to refer to characters from this set. Nonloadable character sets have default LCIDs assigned to them (X'F0' to X'FE'), where:

X'F0' to X'F7' = 1-byte character sets  
X'F8' to X'FE' = 2-byte character sets.

**SW and SH**

Defines the size of all the character slots in the character set. The parameters are present only if character sets with different slot sizes are supported (that is, when the parameter MS = B'1'). If SW and SH are present but are set to zero in any descriptor, the character slot size for that character set is given by SDW and SDH, respectively.

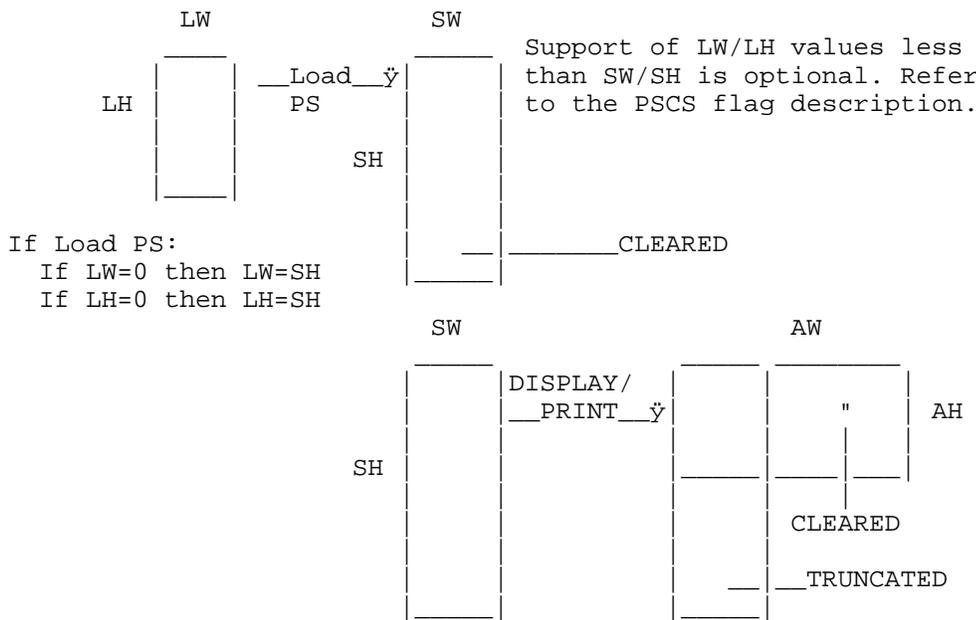
When a character is presented in the usable area, a dot matrix SW by SH is placed in the top left of the appropriate cell.

In a data type that is addressed in cells (for example, alphanumeric), the cell size, (X by Y) is defined when the partition is created, or it defaults to the values associated with the usable area (namely AW by AH). Thus, if SW < X, the remaining cells in each row become background and similarly for SH < Y. If SW > X, each row of the dot matrix is truncated on the right and similarly for SH > Y.

The rules for interpreting the Load PS structured field data stream are summarized below. For FORMAT types 3 through 6, one of the following occurs:

- If LW and LH are present in the Load PS structured field extension, then LW and LH are used.
- If SW and SH are present in the Character Set Descriptor and are non-zero, SW and SH are used.
- SDW and SDH are used.

This operation is illustrated in Figure 6-1.



SW and SH returned in CHAR SET Query Reply.

Defaults, SDW and SDH also returned in CHAR SET Query Reply.

If MS=0, or SW=SH=0, then SW=SDW and SH=SDH.

AW and AH are returned in Usable Area query reply.

Figure 6-1. Steps Used in Interpreting Load PS Data Stream

Bytes 6 and 7 are not present when bit 2 of byte 2 is B'0' (1-byte coded character set). A 2-byte coded character set can consist of a nonloadable portion or both a nonloadable and a loadable portion. If a character set consists of both a nonloadable and loadable portion, then bit 0 of byte 0 (Load flag) is set to B'1' in the descriptor for that character set.

**SUBSN** Bytes 6 and 7 of each descriptor define the starting and ending subsection identifiers supported by the device for the PS store being described.

The starting subsection (byte 6 of each descriptor) defines the starting or lowest subsection identifier allowed for the 2-byte coded character set being described. The default value for a loadable 2-byte character set is X'41', but it can be set by the ST.SUBS of Load PS.

For a nonloadable 2-byte character set, the starting subsection is set to the lowest subsection identifier accepted by the device for that device character set. The ending subsection identifier is set to the highest subsection identifier accepted by the device. Supported subsection identifiers must be consecutive within the range X'41' to X'FE'. For a loadable 2-byte character set, the starting subsection can be set by the ST.SUBSN parameter of Load PS.

The ending subsection is set by the device based on two factors. The first is the starting subsection identifier; the second is the number of physical subsections supported (indicating the range of consecutive subsection identifiers accepted by the device for this loadable device set). The difference between the ending and starting subsection identifiers is one less than the number of subsections supported for that device set. Supported subsection identifiers must be within the range X'41' to X'FE'.

The starting and ending subsection identifiers for loadable 2-byte character sets are reset to their device default values by power-on and test mode. The device default values are X'41' for the starting subsection identifier and the appropriate value for the ending subsection identifier (indicating the number of consecutive subsections supported by the device set).

Subsection parameters (bytes 6 and 7) are present in all descriptors if and only if flag CH2 is set to B'1'. If they are present but the specified device set represents a 1-byte coded character set (CHAR = B'0'), then both SUBSN parameters are set to X'00'.

**CGCSGID**

The CGCSGID consists of a 2-byte character set identifier followed by a 2-byte code page identifier. The CGCSGID is included for each descriptor in the Character Set Query Reply. If the character set being defined in the descriptor has no associated CGCSGID (for instance, PS Sets, which are user defined), the 4 bytes will be set to zeros.

**Note:** The CGCSGID is made up of the 2-byte character set number and the 2-byte code page number. For more information about

CGCSGID values, see the 3174 *Character Set Reference*.

**CCSID** The Coded Character Set Identifier indicates the character set, code page, and encoding scheme. Information on valid parameter values and aspects of Character Data Representation Architecture can be found in the following documents:

- *Character Data Representation Architecture - Level 1, Reference*
- *Character Data Representation Architecture - Level 1, Executive Overview*
- *Character Data Representation Architecture - Level 1, Registry*.

---

## 6.13 Query Reply (Color)

This section describes the Query Reply (Color) structured field.

Subtopics:

- [6.13.1 Function](#)
- [6.13.2 Format](#)
- [6.13.3 Additional Content Description](#)
- [6.13.4 Query Reply \(Color\) Example](#)
- [6.13.5 Default Background Color Self-Defining Parameter](#)

---

### 6.13.1 Function

The Query Reply (Color) structured field transmits information about the color features of the device. The structured field is sent if the device is capable of interpreting 3270 field attributes as color attributes or accepts some subset of the color attribute values.

When this function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'86', Equivalent, or All).

---

### 6.13.2 Format

The following table shows the format of the Query Reply (Color) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply

3		<b>QCODE</b>	<b>X'86'</b> Identifies this Query Reply as Color
4		<b>FLAGS</b>	
	0	RES	Reserved
	1	PRTBLK	Printer Ribbon:
			<b>B'0'</b> Printer only - black ribbon is not loaded
			<b>B'1'</b> Printer only - black ribbon is loaded
	2-7	RES	Reserved
5		<b>NP</b>	Length of color attribute list (NP = number of CAV/COLOR pairs)
n		<b>CAV(n)</b>	Color attribute value accepted by the device
	n+1	CI(n)	Color identifier of the color displayed/printed for CAV(n)

### 6.13.3 Additional Content Description

The content of the Query Reply (Color) structured field is further described as follows:

**PRTBLK** This flag informs the host application if a black ribbon is loaded or not.

**CAV(n)** The parameters CAV(n) are all those color attribute values that are accepted by the device without causing a negative response. Bytes n and n+1 are repeated for each of the data stream values accepted by the device.

**CI(n)** The parameters CI(n) identify the colors that are displayed or printed by the device for each of the accepted color attribute values. The device must either display the color whose color identifier is the same as the color attribute value or display the device default color.

The following table shows the colors and their architected color identifications as they are currently defined:

Color	I.D.	Defined
Neutral	X'F0'	(black on displays, white on printers)
Blue	X'F1'	

Red	X'F2'	
Pink	X'F3'	
Green	X'F4'	
Turquoise	X'F5'	
Yellow	X'F6'	
Neutral	X'F7'	(white on displays, black on printers)
Black	X'F8'	
Deep Blue	X'F9'	
Orange	X'FA'	
Purple	X'FB'	
Pale Green	X'FC'	
Pale Turquoise	X'FD'	
Grey	X'FE'	
White	X'FF'	

The 'color' listed above as Neutral with a color identification of X'F7' is defined as White for a display and Black for a printer.

The color associated with the CAV(n) value of X'F7' defines the default color that is displayed or printed when a single plane character set is referenced; the associated CI(n) value can be any of the values in CAV(n), including X'00'.

The CAV(n) value of X'00' can have an associated CI(n) value of any of the defined values except X'00'.

All devices that send Query Reply (Color) are required to have the values CAV1 = X'00', CI1 = value associated with the device default color, as the first entry in the CAV/CI pairs list.

---

### 6.13.4 Query Reply (Color) Example

The following table shows examples of the values returned by the Query Reply (Color) structured field:

CI(n) Color Identifier	CAV(n) Attribute Value			
	Color Display	Mono Display	Color Printer	3290 Display
X'00'	X'F4'	X'F4'	X'F7'	X'FA'
X'F1'	X'F1'	X'00'	X'F1'	X'00'
X'F2'	X'F2'	X'00'	X'F2'	X'00'

X'F3'	X'F3'	X'00'	X'00'	X'00'
X'F4'	X'F4'	X'00'	X'F4'	X'00'
X'F5'	X'F5'	X'00'	X'00'	X'00'
X'F6'	X'F6'	X'00'	X'00'	X'00'
X'F7'	X'F7'	X'00'	X'00'	X'00'

### 6.13.5 Default Background Color Self-Defining Parameter

The Default Background Color self-defining parameter is present only if the device supports background color. There is only one of this type of self-defining parameter. All color attribute values supported by (foreground) color are supported by background color with the same color identifiers. Therefore, only the background color default is identified by the CAVDEF and CIDEF parameters. The following table shows the format of the Default Background Color self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	Length of this structure
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Default Background Color
2	<b>CAVDEF</b>	<b>X'00'</b> Default color attribute value
3	<b>CIDEF</b>	Default background color identifier

## 6.14 Query Reply (Cooperative Processing Requestor)

This section describes the Query Reply (Cooperative Processing Requestor) structured field.

Subtopics:

- [6.14.1 Function](#)
- [6.14.2 Format](#)
- [6.14.3 Additional Content Description](#)
- [6.14.4 Direct Access ID Self-Defining Parameter](#)

### 6.14.1 Function

The Query Reply (Cooperative Processing Requestor) structured field indicates that Cooperative Processing Requestor (CPR) functions are supported.

When this function is supported, the Query Reply is transmitted inbound in

### 6.14.2 Format

The following table shows the format of the Query Reply (Cooperative Processing Requestor) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'AB'</b> Identifies this Query Reply as the Cooperative Processing Requestor
4-5	<b>FLAGS</b>	Reserved
6-7	<b>LIMIN</b>	Maximum CPR bytes/transmission allowed inbound
8-9	<b>LIMOUT</b>	Maximum CPR bytes/transmission allowed outbound
10	<b>FEATL</b>	Length (in bytes) of feature information that follows
11-12	<b>FEATS</b>	CPR length and feature flags
13 to (N×2)	<b>FEATs</b> 12	Additional flags

### 6.14.3 Additional Content Description

The content of the Query Reply (cooperative Processing Requestor) structured field is further described as follows:

**LIMIN** Certain implementations can have a limit on the number of bytes of CPR information that are allowed inbound following an AID88. The number of bytes in an inbound CPR transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on inbound CPR data.

**LIMOUT** The sum of bytes contained in all the CPR structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission is rejected. The data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound CPR data.

The LIMOUT parameter applies only to the CPR structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by a CPR SF (100 bytes), 3270 SF (500 bytes) and a CPR SF (300 bytes) would be accepted.

**FEATL** Indicates the number of bytes of feature information that follow. The minimum value is X'02'.

**FEATS** The content of the feature bytes is defined in related publications.

---

## 6.14.4 Direct Access ID Self-Defining Parameter

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Parameter Length
1	SDPID	X'01' Identifies this self-defining parameter as Direct Access ID
2-3	DOID	Destination/Origin Identification

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** The presence of the Direct Access ID self-defining parameter indicates that the CPR device can be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate that the destination or origin of the data following it is the CPR device. When more than one CPR device is supported, each one has a separate CPR Query Reply and a separate ID.

For the Query Reply to be valid, this self-defining parameter must be present.

---

## 6.15 Query Reply (Data Chaining)

This section describes the Query Reply (Data Chaining) structured field.

Subtopics:

- [6.15.1 Function](#)
- [6.15.2 Format](#)

---

### 6.15.1 Function

The Query Reply (Data Chaining) structured field indicates that data chaining is supported in the non-SNA environment.

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE=X'98', Equivalent, or All).

## 6.15.2 Format

The following table shows the format of the Query Reply (Data Chaining) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	<b>X'0006'</b> Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'98'</b> Identifies this Query Reply as Data Chaining
4	0-1	<b>DIR</b>	Indicates which direction can use the Data Chain structured field.
			<b>B'00'</b> Both <b>B'01'</b> From device only <b>B'10'</b> To device only
	2-7	<b>SFID</b>	<b>B'11'</b> Reserved
5		<b>FLAGS</b>	Reserved

## 6.16 Query Reply (Data Streams)

This section describes the Query Reply (Data Streams) structured field.

Subtopics:

- [6.16.1 Function](#)
- [6.16.2 Format](#)

### 6.16.1 Function

The Query Reply (Data Streams) structured field indicates which data streams are supported by the device. The positional relationship of the data within the structured field indicates the default data stream.

The following list describes the valid data stream identifiers:

X'00' SCS Base Data Stream with extensions as specified in the BIND

X'01' Document Content Architecture Level 2

X'02' IPDS as defined in related documentation.

All other values are reserved. As noted, the first identifier to appear in the subfield (for example, at byte 4 of the structured field) is assumed to be the default data stream.

When the function is supported, this Query Reply is transmitted inbound in response to a Read Partition structured field specifying Query, or Query List (QCODE List=X'A2', Equivalent, or All).

---

## 6.16.2 Format

The following table shows the format of the Query Reply (Data Streams) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'A2' Identifies this Query Reply as Data Streams
4	SFID	X'nn...' List of (L-4) data stream identifiers, the first of which is the default data stream

---

## 6.17 Query Reply (DBCS-Asia)

This section describes the Query Reply (DBCS-Asia) structured field.

Subtopics:

- [6.17.1 Function](#)
- [6.17.2 Format](#)
- [6.17.3 SO/SI Self-Defining Parameter](#)
- [6.17.4 Input Control Self-Defining Parameter](#)

---

### 6.17.1 Function

This Query Reply indicates the support of the DBCS-Asia node as defined in [Chapter 12, "Double-Byte Coded Character Set \(DBCS\)-Asia."](#)

When this function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query

---

## 6.17.2 Format

The following table shows the format of the Query Reply (DBCS-Asia) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'91'</b> Identifies this Query Reply as DBCS-Asia
4		<b>FLAGS</b>	Reserved

The DBCS-Asia Query Reply can accept one or both of the following self-defining parameters:

- SO/SI
- Input Control.

---

## 6.17.3 SO/SI Self-Defining Parameter

The presence of the SO/SI self-defining parameter indicates that the host can send SO/SI to the device. Also, if previously written by the host, SO/SI can be read back. However, unless indicated in the Input Control self-defining parameter, SO/SI cannot be generated by the operator.

The following table shows the format of the SO/SI self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'03'</b> Parameter length
1		<b>SDPID</b>	<b>X'01'</b> Identifies this self-defining parameter as SO/SI
2		<b>SOSET</b>	Set ID of the Shift Out (SO) character set

**SOSET** Designates the Set ID of the character set associated with SO. The LCID that the implementation assigned to the character set designated in SOSET is reported in the Character Sets Query Reply.

---

## 6.17.4 Input Control Self-Defining Parameter

The presence of the Input Control self-defining parameter indicates support of the extended field attribute type, Input Control.

The following table shows the format of the Input Control self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'03'</b> Parameter length
1		<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Input Control
2		<b>FUNC</b>	Functions supported
	0-6	RES	Reserved
	7	CREATE	Operator creation of SO/SI
			<b>B'0'</b> SO/SI Creation not supported
			<b>B'1'</b> SO/SI Creation supported

The content of the Input Control self-defining parameter is further described as follows:

**CREATE** Indicates whether the Input Control self-defining parameter can be used to enable or disable the operator creation of SO/SI. This parameter can have one of two values:

**B'0'** Indicates creation of SO/SI by the operator is not supported.

**B'1'** Indicates creation of SO/SI by the operator is supported and may be enabled or disabled using the Input Control extended attribute. The default (for example, POR) is disabled.

**Note:** When CREATE = B'1' (operator creation of SO/SI supported), the SO/SI self-defining parameter must be present.

---

## 6.18 Query Reply (Device Characteristics)

This section describes the Query Reply (Device Characteristics) structured field.

Subtopics:

- 6.18.1 Function
- 6.18.2 Format
- 6.18.3 Function Descriptors
- 6.18.4 Set Print Density (SPD) Descriptor
- 6.18.5 Horizontal Dimensional Parameters Descriptor

- 6.18.6 Vertical Dimensional Parameters Descriptor
- 6.18.7 Page Presentation Media (PPM) Descriptor
- 6.18.8 Set Text Orientation (STO) Descriptor

## 6.18.1 Function

The Query Reply (Device Characteristics) structured field transmits the device's ability to support SNA SCS functions.

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'A0', Equivalent, or All). The function descriptor is included in the Device Characteristics query reply to indicate how an SCS function is supported.

As a general rule, any of the SCS control codes reported in the Device Characteristics Query Reply which are received with invalid parameters, are rejected. However, there are some situations where there is a relationship between parameters in which a normally valid parameter value becomes invalid in combination with another parameter value. For example, a device supporting character densities of 10 and 15 cpi could support a maximum MPP of 160 only when the character density was 15 cpi. The descriptors describe the action taken for the sets of parameters that are interrelated.

## 6.18.2 Format

The following table shows the format of the Query Reply (Device Characteristics) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as Query Reply
3	<b>QCODE</b>	<b>X'A0'</b> Identifies this Query Reply as Device Characteristics

## 6.18.3 Function Descriptors

The base part is followed by one or more of the following function descriptors:

- Set Print Density
- Horizontal Dimensional Parameters
- Vertical Dimensional Parameters
- Page Presentation Media
- Set Text Orientation.

Each descriptor defines the characteristics of one SCS control.

---

## 6.18.4 Set Print Density (SPD) Descriptor

This section describes the Set Print Density (SPD) descriptor.

Subtopics:

- [6.18.4.1 Function](#)
- [6.18.4.2 Format](#)

---

### 6.18.4.1 Function

The SPD descriptor is included for historical purposes. The Horizontal Dimensional Parameters descriptor should be used in its place.

---

### 6.18.4.2 Format

The following table shows the format of the SPD descriptor:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2-3	<b>SCS2B</b>	<b>X'1104'</b> SCS X'2B' identifier
4-5	<b>SPDDES</b>	<b>X'D229'</b> Identifies this descriptor as Set Print Density
6	<b>SPDSUP</b>	<b>X'00'</b> SPD is supported. Default value is used.
7-8	<b>SPDDEF</b>	<b>X'00nn'</b> Default value for print density (for example, X'000A' is 10 characters per inch)
9	<b>CDLEN</b>	<b>X'02'</b> Character density parameter length
10	<b>CDPRES</b>	<b>X'60'</b> The character density is present sometimes. It takes only discrete values, and it appears only one time.
11	<b>CDVALS</b>	<b>X'mmm'</b> Number of discrete values supported by the device
12-n	<b>CDPARM</b>	The discrete character density values supported

---

## 6.18.5 Horizontal Dimensional Parameters Descriptor

This section describes the Horizontal Dimensional Parameters descriptor.

Subtopics:

- 6.18.5.1 Function
- 6.18.5.2 Format
- 6.18.5.3 Additional Content Description

### 6.18.5.1 Function

With Horizontal Dimensional Parameters, character density and maximum print position (MPP) occur as 2-byte pairs, one for each density. The character density is the number of characters per inch, rounded down to the next integer. The first character density/MPP pair is the device default setting. The other character density/MPP pairs must appear in the ascending order of MPP values.

### 6.18.5.2 Format

The following table shows the format of the Horizontal Dimensional Parameters descriptor:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>HDPDES</b>	<b>X'FF01'</b> Identifies this descriptor as Horizontal Dimensional Parameters
4		<b>FLAGS</b>	
	0	SPDSUP	<b>B'0'</b> SPD not supported <b>B'1'</b> SPD supported
	1-7	RES	Reserved
4+2n-		<b>CD(n)</b>	Character Density
4+2n		<b>MPP(n)</b>	Maximum print position for character density

### 6.18.5.3 Additional Content Description

The content of the Horizontal Dimensional Parameters descriptor is further described as follows:

#### Character density/MPP

Character density is given preference. That is, it takes effect, regardless of whether the maximum MPP in effect is valid for that character density. If the maximum MPP in effect becomes invalid when a new character density is set, the maximum MPP becomes the highest value valid for the new character density. If a new MPP is set using the SHF control that is invalid for the character density in effect, the MPP is rejected.

## 6.18.6 Vertical Dimensional Parameters Descriptor

This section describes the Vertical Dimension Parameters descriptor.

Subtopics:

- [6.18.6.1 Function](#)
- [6.18.6.2 Format](#)
- [6.18.6.3 Additional Content Description](#)

---

### 6.18.6.1 Function

With Vertical Dimensional Parameters, typographic points and maximum page length (MPL) occur as 2-byte pairs, one for each density. The typographic points are expressed in 1/72nds of an inch and are rounded down to the next integer. The first typographic point/MPL pair is the device default setting. The other typographic point/MPL pairs must appear in the ascending order of MPL values.

---

### 6.18.6.2 Format

The following table shows the format of the Vertical Dimensional Parameters descriptor:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2-3		<b>VDPDES</b>	<b>X'FF02'</b> Identifies this descriptor as Vertical Dimensional Parameters
4		<b>FLAGS</b>	
	0	SLDSUP	<b>B'0'</b> Set Line Density not supported <b>B'1'</b> Set Line Density supported
	1-7	RES	Reserved
4+2n-		<b>PNTS(n)</b>	Typographic points
4+2n		<b>MPL(n)</b>	Maximum page length for PNTS

---

### 6.18.6.3 Additional Content Description

The content of the Vertical Dimensional Parameters descriptor is further described as follows:

## PNTS/MPL

Treated the same as character density/MPP, the line density PNTS is given preference, and a valid PNTS is accepted whether or not the MPL in effect is valid for that PNTS. Also, if a new MPL is set with the SVF control that is invalid for the PNTS in effect, the MPL is rejected.

---

## 6.18.7 Page Presentation Media (PPM) Descriptor

This section describes the Page Presentation Media (PPM) descriptor.

Subtopics:

- [6.18.7.1 Function](#)
- [6.18.7.2 Format](#)
- [6.18.7.3 Additional Content Description](#)

---

### 6.18.7.1 Function

The PPM descriptor indicates whether the device provides support for such functions as forms control, paper source and destination drawers, print quality level, and duplex printing.

On certain devices, PS sets have only the Quality levels inherent in the characteristics in which they were loaded into the device. On these devices, quality cannot be changed on PS sets. Other devices have the capability of changing quality for both PS sets as well as nonloadable character sets by not requiring a downstream load to change the quality level. Devices set the Quality flag byte, bit 1, to designate their capability to change quality on PS sets.

**Note:** On certain devices, changes in print quality can inherently result in a change in cell geometry. Following a change in the Quality parameter, the Primary should issue a Query List with the Usable Area Query Reply specified to determine the effects of the quality change upon the cell geometry.

For some devices, certain combinations of supported quality and character density are invalid. The existence of this condition is indicated by setting bit 3 of byte m+1 to B'1' and including a count of the number of invalid Q,character density pairs. This is followed by a list of the invalid Q,character density combinations (2 bytes per pair). Devices that can support all combinations of supported quality and character density set bit 3 of byte (m+1) to B'0' and do not include the above fields.

---

### 6.18.7.2 Format

The following table shows the format of the PPM descriptor:

Byte	Bit	Content	Content Description
------	-----	---------	---------------------

0-1		<b>L</b>	Length of this structured field
2-3		<b>PPMDES</b>	<b>X'FF03'</b> Identifies this descriptor as PPM
4		<b>FLAGS</b>	
	0	PPMSUP	<b>B'0'</b> PPM not supported <b>B'1'</b> PPM supported
	1-7	RES	Reserved
5		<b>FC</b>	Parameter Forms Control (FC)
	0	FCSUP	<b>B'0'</b> FC not supported <b>B'1'</b> FC supported
	1-7	RES	Reserved
6		<b>FCNUMB</b>	<b>X'nn'</b> Number of discrete Forms Control (FC) parameter values supported by the device
7 to j		<b>FCVALS</b>	The discrete FC values supported. The first value is the default value. Each value occupies 1 byte.
j+1		<b>SD</b>	Parameter Source Drawer (SD)
	0	SDSUP	<b>B'0'</b> SD not supported <b>B'1'</b> SD supported
	1	SDOPER	<b>B'0'</b> This parameter is not operator-selectable at the device. <b>B'1'</b> This parameter is operator-selectable at the device. Its current value can be invoked by the Primary setting SD = X'FF'.
	2-7	RES	Reserved
j+2		<b>SDNUMB</b>	<b>X'nn'</b> Number of discrete SD parameter values supported by the device
j+3 to k		<b>SDPARM</b>	The discrete SD values supported. The first value is also the default value. Each value occupies 1 byte.
k+1		<b>DDO</b>	Parameter Destination Drawer Offset (DDO)
	0	DDOSUP	<b>B'0'</b> DDO not supported <b>B'1'</b> DDO supported

	1-7	RES	Reserved

Byte	Bit	Content	Content Description
k+2		<b>DD</b>	Parameter Destination Drawer (DD)
	0	DDSUP	<b>B'0'</b> DD not supported <b>B'1'</b> DD supported
	1	DDOPER	<b>B'0'</b> This parameter is not operator-selectable at the device. <b>B'1'</b> This parameter is operator-selectable at the device. Its current value can be invoked by the Primary setting DD=X'FF'.
	2-7	RES	Reserved
k+3		<b>DDNUMB</b>	<b>X'nn'</b> Number of discrete DD parameter values supported by the device.
k+4 to m		<b>DDVALS</b>	The discrete values supported. The first value is also the default value.
m+1		<b>QUAL</b>	Parameter Quality (Q)
	0	QSUP	<b>B'0'</b> Q not supported <b>B'1'</b> Q supported
	1	QOPER	<b>B'0'</b> This parameter is not operator-selectable at the device. <b>B'1'</b> This parameter is operator-selectable at the device. Its current value can be invoked by the Primary specifying Q=X'FF'.
	2	QPS	<b>B'0'</b> Q is not applicable to PS sets. <b>B'1'</b> Q is applicable to PS sets.
	3	QALL	<b>B'0'</b> All supported quality levels are allowed at all supported character densities. <b>B'1'</b> Some combinations of supported quality levels/character densities are not valid.
	4-7	RES	Reserved
m+2		<b>QNUMB</b>	<b>X'nn'</b> Number of discrete Q parameter values supported by the device.
m+3		<b>QPARMS</b>	The discrete Q values supported. The first

to n			value is also the default value (1 byte per value).
n+1		<b>QCDNUMB</b>	<b>X'PP'</b> Number of discrete Q/character density pair parameter values that are not allowed by the device.
n+2 to p		<b>Q/character density()</b>	<b>eQ/character density pairs of valid values that are not allowed as a combination. Two bytes per pair.</b>

Byte	Bit	Content	Content Description
p+1		<b>DUP</b>	Parameter Duplex (D)
	0	<b>DUPSUP</b>	<b>B'0'</b> D is not supported. <b>B'1'</b> D is supported.
	1	<b>DUPOPER</b>	<b>B'0'</b> This parameter is not operator-selectable at the device. <b>B'1'</b> This parameter is operator-selectable at the device. Its current value can be invoked by the Primary setting D = X'FF'.
	2-7	<b>RES</b>	Reserved

### 6.18.7.3 Additional Content Description

The content of the PPM descriptor is further described as follows:

#### Character density/Q

Character density is given preference. A valid character density takes effect whether or not the quality (Q) level in effect is valid for that character density. When a new character density selection makes the current Q level invalid, the Q level defaults to the next highest valid Q level. If a new valid Q level is set with the PPM that is invalid because of the character density in effect, it is rejected.

**FCPARAM** Valid values for the FC mechanisms are:

X'01' Paper Source Drawer  
X'02' Envelope Source Drawer  
X'03' Manual Paper Drawer  
X'04' Manual Envelope Drawer.

**SDPARAM** Valid values for the Paper SDs are:

X'01' Select paper from bin 1  
X'02' Select paper from bin 2  
X'03' Select paper from bin 3

X'04' to X'FE'  
Select paper from respective drawer.

**DDPARM** Valid values for the Paper DDs are:

X'01' Primary destination drawer  
X'02' to X'FE'  
Secondary destination drawers.

**QPARM** The number of print quality levels supported are listed in order of worst to best print quality.

For interrelated parameters (for example, character density/MPP), when both parameters are changed in the same transmission, the results obtained can be different, depending on which parameter appears first.

---

## 6.18.8 Set Text Orientation (STO) Descriptor

This section describes the Set Text Orientation (STO) descriptor.

Subtopics:

- [6.18.8.1 Function](#)
- [6.18.8.2 Format](#)

---

### 6.18.8.1 Function

The STO descriptor indicates that the Set Text Orientation SCS control code is supported by the device.

---

### 6.18.8.2 Format

The following table shows the format for the STO descriptor:

Byte	Bit	Content	Content Description
0 - 1		<b>L</b>	<b>X'0005'</b> Length of this structured field
2 - 3		<b>STODES</b>	<b>X'FF04'</b> Identifies this descriptor as Set Text Orientation
4		<b>FLAGS</b>	
	0	STOSUP	<b>B'0'</b> STO not supported <b>B'1'</b> STO supported
	1 - 7	RES	Reserved

--	--	--	--

## 6.19 Query Reply (Distributed Data Management)

This section describes the Query Reply (Distributed Data Management) structured field.

Subtopics:

- [6.19.1 Function](#)
- [6.19.2 Format](#)
- [6.19.3 Additional Content Description](#)
- [6.19.4 Direct Access ID Self-Defining Parameter](#)
- [6.19.5 DDM Application Name Self-Defining Parameter](#)
- [6.19.6 PCLK Protocol Controls Self-Defining Parameter](#)

### 6.19.1 Function

The Query Reply (Distributed Data Management) indicates the Distributed Data Management (DDM) subsets supported.

When this function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'95', Equivalent, or All).

### 6.19.2 Format

The following table shows the format of the Query Reply (DDM) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'95'</b> Identifies this Query Reply as DDM
4-5	<b>FLAGS</b>	Reserved
6-7	<b>LIMIN</b>	Maximum DDM bytes/transmission allowed inbound
8-9	<b>LIMOUT</b>	Maximum DDM bytes/transmission allowed outbound
10	<b>NSS</b>	Number of subsets supported
N	<b>DDMSS</b>	DDM subset identifier

### 6.19.3 Additional Content Description

The content of the Query Reply (DDM) structured field is further described as follows:

**LIMIN** Certain implementations can have a limit on the number of bytes of DDM information that are allowed inbound following an AID88. The number of bytes in an inbound DDM transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on inbound DDM data.

**LIMOUT** The sum of bytes contained in all the DDM structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission is rejected. The data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound DDM data.

The LIMOUT parameter applies only to the DDM structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by a DDM structured field (100 bytes), 3270 structured field (500 bytes), and a DDM structured field (300 bytes) would be accepted.

**NSS** Indicates the number of different subsets that are supported, that is, the number of DDMSS parameters present.

**DDMSS** Indicates the DDM subset that is supported. Valid values are as follows:

- X'01' = DDM Copy Subset 1
- All other values reserved.

The DDM Query Reply supports three self-defining parameters:

- Direct Access ID
- DDM Application Name
- GDDM-PCLK Protocol Controls.

---

## 6.19.4 Direct Access ID Self-Defining Parameter

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Parameter Length
1	SDPID	X'01' Identifies this self-defining parameter as Direct Access ID
2-3	DOID	Destination/Origin Identification

The content of the Direct Access ID self-defining parameter is further

described as follows:

**DOID** The presence of the Direct Access ID self-defining parameter indicates that the DDM device can be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate the destination or origin of the following data is the DDM device.

---

## 6.19.5 DDM Application Name Self-Defining Parameter

This parameter provides the host with the name of the application containing control of the DDM auxiliary device that is identified by the DOID in the Direct Access self-defining parameter.

This parameter is optional unless the host application must identify distinct DDM auxiliary devices when more than one application at the remote workstation supports DDM auxiliary devices.

The following table shows the format of the DDM Application Name self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	Parameter Length
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as DDM Application Name
2-3	<b>NAME</b>	Name of the remote application program

The content of the DDM Application Name self-defining parameter is further described as follows:

**NAME** This value is a name of eight characters or less by which a host application can relate to an application in a remote workstation. It is the responsibility of the host and the remote application users to ensure that the name is understood by the applications on each end.

---

## 6.19.6 PCLK Protocol Controls Self-Defining Parameter

The presence of the PCLK Protocol Controls self-defining parameter indicates that the GDDM-PCLK Protocol Controls structured field (X'1013') can be used in both inbound and outbound DDM Auxiliary Device data streams.

The following table shows the format of the PCLK Protocol Controls self-defining parameter:

Byte	Content	Content Description
------	---------	---------------------

0	<b>L</b>	<b>X'04'</b> Parameter Length
1	<b>SDPID</b>	<b>X'03'</b> Identifies this self-defining parameter as PCLK Protocol Controls
2-3	<b>VERS</b>	Protocol version

The content of the PCLK Protocol Controls self-defining parameter is further described as follows:

**VERS** This value indicates the version of GDDM-PCLK installed in the terminal at the time when the Query Reply is returned. X'0001' indicates GDDM-PCLK Version 1.1.

---

## 6.20 Query Reply (Document Interchange Architecture)

This section describes the Query Reply (Document Interchange Architecture) structured field.

Subtopics:

- [6.20.1 Function](#)
- [6.20.2 Format](#)
- [6.20.3 Additional Content Description](#)
- [6.20.4 Direct Access ID Self-Defining Parameter](#)

---

### 6.20.1 Function

The Query Reply (Document Interchange Architecture) structured field indicates the Document Interchange Architecture (DIA) function sets supported.

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query, or Query List (QCODE List=X'97', Equivalent, or All).

---

### 6.20.2 Format

The following table shows the format of the Query Reply (Document Interchange Architecture) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'97'</b> Identifies this Query Reply as Document Interchange Architecture

4-5	<b>FLAGS</b>	Reserved
6-7	<b>LIMIN</b>	Maximum DIA bytes/transmission allowed inbound
8-9	<b>LIMOUT</b>	Maximum DIA bytes/transmission allowed inbound
10	<b>NFS</b>	Number of 3-byte function set identifiers that follow
11-13	<b>DIAFS</b>	DIA function set identifier
N-N=2	<b>DIAFSs</b>	Additional DIA function set identifiers

## 6.20.3 Additional Content Description

The content of the Query Reply (Document Interchange Architecture) structured field is further described as follows:

**LIMIN** Certain implementations can have a limit on the number of bytes of DIA information that are allowed inbound following an AID X'88'. The number of bytes in an inbound DIA transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on inbound DIA data.

**LIMOUT** The sum of bytes contained in all the DIA structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission is rejected. The data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on DIA data outbound.

The LIMOUT parameter applies only to the DIA structured fields. For example, if LIMOUT=400 bytes, a WSF followed by a DIA structured field (100 bytes), an Outbound 3270DS structured field (500 bytes), and a DIA structured field (300 bytes) would be accepted.

**NFS** The number of different function sets that are supported, that is, the number of 3-byte DIAFS parameters present, as follows: A valid Query Reply must have at least one DIAFS.

X'01' File Server  
X'02' File Requestor  
X'03' Both File Server and File Requestor  
All other values reserved.

The second and third bytes give the function set number in hexadecimal. For example, to indicate the role of the File Server with support of function set 11, the value of bytes 11 through 13 would be X'0100B'.

For a description of the Document Interchange Architecture functions refer to the following publications:

- *Document Interchange Architecture Technical Reference*
- *Document Interchange Architecture Document Profile Reference*

- *Document Interchange Architecture Transaction Programmer's Guide.*

See "Where To Find More Information" in topic PREFACE.4 .

---

## 6.20.4 Direct Access ID Self-Defining Parameter

For the Query Reply to be valid, the Direct Access ID self-defining parameter must be present.

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Parameter length
1	SDPID	X'01' Identifies this self-defining parameter as Direct Access ID
2-3	DOID	Destination/Origin Identification

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** The presence of the Direct Access ID self-defining parameter indicates that the DIA device can be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate that the destination or origin of the following data is the DIA device. When more than one DIA device is supported, each one has a separate DIA Query Reply and a separate ID.

---

## 6.21 Query Reply (Extended Drawing Routine)

This section describes the Query Reply (Extended Drawing Routine) structured field.

Subtopics:

- 6.21.1 Function
- 6.21.2 Format

---

### 6.21.1 Function

The Query Reply (Extended Drawing Routine) structured field indicates at which graphic subset level extended drawing routines are supported.

When this function is supported, it is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B5' or All).

---

## 6.21.2 Format

The following table shows the format of the Query Reply (Extended Drawing Routine) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'B5'</b> Identifies this Query Reply as Extended Drawing Routine
4-n	<b>DATA</b>	Data

**Note:** For information regarding the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

---

## 6.22 Query Reply (Field Outlining)

This section describes the Query Reply (Field Outlining) structured field.

Subtopics:

- [6.22.1 Function](#)
- [6.22.2 Format](#)
- [6.22.3 Additional Content Description](#)

---

### 6.22.1 Function

The Query Reply (Field Outlining) structured field specifies the details of the field outlining supported by the device.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'8C', Equivalent, or All).

---

### 6.22.2 Format

The following table shows the format of the Query Reply (Field Outlining) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	<b>X'000A'</b> Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'8C'</b> Identifies this Query Reply as Field Outlining
4	0-7	<b>FLAG</b>	Reserved
5	0	<b>SEP</b>	Support of separation of underlining and overlining:  <b>B'0'</b> Separation not supported <b>B'1'</b> Separation supported
	1-7	<b>RES</b>	Reserved
6		<b>VPOS</b>	Location of vertical line
7		<b>HPOS</b>	Location of overline/underline
8		<b>HPOS0</b>	Location of overline in case of separation
9		<b>HPOS1</b>	Location of underline in case of separation

### 6.22.3 Additional Content Description

The content of the Query Reply (Field Outlining) structured field is further described as follows:

**SEP** Indicates that the separation of underline and overline is supported in printers. When the function is not supported (SEP = B'0'), an underline and the next overline are always printed as one line. When the function is supported (SEP = B'1'), an underline and the next overline are separated when specified.

**VPOS** Indicates the horizontal location of the vertical line.

**HPOS** Indicates the vertical location of the horizontal line. In the printer, the location of the horizontal line when the separation is not specified is indicated here. When the overline is drawn above the cell, the location of the underline, which is the same as the location of the overline in the next row, is used.

**HPOS0 and HPOS1**

Indicate the vertical locations of the overline and the underline respectively when Separation is specified. In displays or printers where the device does not support Separation, zero values are set. When the overline is drawn above the character cell, the location of the overline in the next row is used.

Each of VPOS, HPOS, HPOS0, and HPOS1 is a 1-byte binary number, and

they indicate the location of the Field Ruling lines against the top left corner of the character cell, where the value of the location is zero vertically and horizontally. These values are measured in the same unit as that in cell units of the Usable Area Query Reply and measured when the Skip Suppression is specified.

## 6.23 Query Reply (Field Validation)

This section describes the Query Reply (Field Validation) structured field.

Subtopics:

- [6.23.1 Function](#)
- [6.23.2 Format](#)

### 6.23.1 Function

The Query Reply (Field Validation) structured field specifies that the device supports field validation and indicates the types of validation the device supports.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'8A', Equivalent, or All).

### 6.23.2 Format

The following table shows the format of the Query Reply (Field Validation) structured field:

Byte	Bit	Content	Content Description
0-1		L	X'0005' Length of this structured field
2		SFID	X'81' Identifies this structured field as a Query Reply
3		QCODE	X'8A' Identifies this Query Reply as Field Validation
4		TYPES	Types supported:
	0-4	RES	Reserved
	5	MANDFILL	B'0' Mandatory fill not supported B'1' Mandatory fill supported
	6	MANDENTR	B'0' Mandatory entry not supported B'1' Mandatory entry supported

	7	TRIGGER	<b>B'0'</b> Trigger not supported <b>B'1'</b> Trigger supported

---

## 6.24 Query Reply (Format Presentation)

This section describes the Query Reply (Format Presentation) structured field.

Subtopics:

- [6.24.1 Function](#)
- [6.24.2 Format](#)

---

### 6.24.1 Function

The Query Reply (Format Presentation) structured field specifies that the device supports Format Presentation.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'90', Equivalent, or All).

---

### 6.24.2 Format

The following table shows the format of the Query Reply (Format Presentation) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0004'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'90'</b> Identifies this Query Reply as Format Presentation

---

## 6.25 Query Reply (Format Storage Auxiliary Device)

This section describes the Query Reply (Format Storage Auxiliary Device) structured field.

Subtopics:

- 6.25.1 Function
- 6.25.2 Format
- 6.25.3 Additional Content Description
- 6.25.4 Direct Access ID Self-Defining Parameter

## 6.25.1 Function

The Format Storage Auxiliary Device Query Reply indicates that the device supports the loading of format storage by the Load Format Storage structured field. This Query Reply also provides the DOID to be used in the Destination/Origin structured field, which precedes data to and from format storage.

The Format Storage Auxiliary Device Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'94', Equivalent or All) by any device which supports the Load Format Storage (LFS) node.

The minimum support for the Format Storage Auxiliary Device query reply consists of the Format Storage Auxiliary Device Query Reply Base and the Direct Access ID self-defining parameter. The Format Storage Auxiliary Device Query Reply has one self-defining parameter: Direct Access ID.

The Format Storage Auxiliary Device Query Reply precedes the Direct Access ID self-defining parameter and must be present when the Format Storage Auxiliary Device Query Reply is supported.

## 6.25.2 Format

The following table shows the format of the Query Reply (Format Storage Auxiliary Device) structured field:

Byte	Bit	Content	Content Description
0-1		L	Length of this structured field
2		SFID	X'81' Identifies this structured field as a Query Reply
3		QCODE	X'94' Identifies this Query Reply as Format Storage Auxiliary Device
4		FLAGS	
	0	F/T	Formats per transmission
			B'0' One format per transmission allowed B'1' Multiple formats/transmission allowed
	1	CSTAT	Current Status
			B'0' No formats currently loaded

			<b>B'1'</b> One or more formats currently loaded
	2	OPSEL	User selected format default
			<b>B'0'</b> Disabled <b>B'1'</b> Enabled
	3	HMGMT	Host Format Management
			<b>B'0'</b> Non-shared <b>B'1'</b> Shared
	4-7	RES	Reserved, must be set to zeros
5		<b>FLAGS</b>	Reserved, must be set to zeros
6-7		<b>LIMIN</b>	Reserved for LIMIN parameter. Must be set to zeros.
8-9		<b>LIMOUT</b>	Maximum bytes of format storage data per transmission allowed outbound.

### 6.25.3 Additional Content Description

The content of the Query Reply (Format Storage Auxiliary Device) structured field is further described as follows:

**QCODE** The X'94' is the code for the Format Storage Auxiliary Device Query Reply.

**F/T** This flag indicates whether the host application is limited to one format per transmission or if more than one format per transmission is allowed if the function is supported in non-SNA, as follows:

**B'0'** Only one format per transmission or per Data Chain if the function is supported in non-SNA. The next format cannot be sent until the Format Storage Auxiliary Device sends in an Exception/Status structured field indicating either Acknowledge or Reject.

**B'1'** More than one format is allowed per transmission (or per Data Chain). After sending the transmission (or Data Chain), no further formats can be sent until the Format Storage Auxiliary Device sends in an Exception/Status structured field indicating either Acknowledge or Reject. If rejected, the Exception/Status structured field must include the Failing Format self-defining parameter to indicate the name of the first format encountered with an exception condition.

Regardless of the setting of the F/T flag, if the host application does not wait for the required Acknowledge/Reject, the result is unpredictable. A transmission cannot contain a

partial format. Therefore, a format cannot span a transmission unless Data Chaining is being used in a non-SNA environment. If Data Chaining is being used, a format cannot span Data Chains, as follows:

- CSTAT** This indicates whether or not there are any formats loaded at the time the Query Reply is sent.
- B'0' No formats were loaded when the query was sent.
- B'1' One or more formats were present in format storage when the Query Reply was sent.
- OPSEL** This indicates the default state for user-selected formats. The default state is entered after a reset condition established by configuration settings or host control, as follows:
- B'0' Indicates that the default state for user-selected formats is disabled.
- B'1' Indicates that formats presented by this host application can be loaded by another host application.
- LIMIN** This field is reserved for the LIMIN parameter and should be set to all zeros.
- LIMOUT** The sum of the bytes contained in all the Load Format Storage structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission is rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on format storage data in an outbound transmission.

The LIMOUT parameter applies only to the Load Format Storage structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by a Load Format Storage structured field (100 bytes), Outbound 3270DS structured field (500 bytes) and a Load Format Storage structured field (300 bytes) would be accepted.

---

## 6.25.4 Direct Access ID Self-Defining Parameter

The Direct Access ID self-defining parameter provides the DOID value to be used in the Destination/Origin structured field preceding information to and from format storage. The Direct Access self-defining parameter must be included whenever the Format Storage Auxiliary Device Query Reply is sent.

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Bit	Content	Content Description
0		L	X'06' Length of this self-defining parameter

1		<b>SDPID</b>	<b>X'01'</b> Identifies this self-defining parameter as Direct Access ID
2-3		<b>DOID</b>	Destination/Origin Identification
4-5		<b>SIZE</b>	Size of the format storage space.

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** Provides the value to use in the DOID field of the Destination/Origin structured field which precedes information to and from format storage.

**SIZE** Provides the total size of the format storage space in K bytes. For example, the Size parameter for a format storage space of 32 K bytes would be X'0020'.

---

## 6.26 Query Reply (Graphic Color)

This section describes the Query Reply (Graphic Color) structured field.

Subtopics:

- [6.26.1 Function](#)
- [6.26.2 Format](#)

---

### 6.26.1 Function

The Query Reply (Graphic Color) structured field indicates the support for color available in graphics.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B4', or All).

---

### 6.26.2 Format

The following table shows the format of the Query Reply (Graphic Color) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply

3	QCODE	X'B4'	Identifies this Query Reply as Graphic Color
4-n	DATA	Data	

**Note:** For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

## 6.27 Query Reply (Graphic Symbol Sets)

This section describes the Query Reply (Graphic Symbol Sets) structured field.

Subtopics:

- [6.27.1 Function](#)
- [6.27.2 Format](#)

### 6.27.1 Function

The Query Reply (Graphic Symbol Sets) structured field reports all of the symbol set stores that are available for use in graphics.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B6', or All).

### 6.27.2 Format

The following table shows the format of the Query Reply (Graphic Symbol Sets) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'B6' Identifies this Query Reply as Graphic symbol Sets
4-n	DATA	Data

**Note:** For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

## 6.28 Query Reply (Highlighting)

This section describes the Query Reply (Highlighting) structured field.

Subtopics:

- 6.28.1 Function
- 6.28.2 Format

---

## 6.28.1 Function

The Query Reply (Highlighting) structured field transmits the types of highlighting supported by the device.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'87', Equivalent, or All).

If a device accepts the highlight attribute, then it must accept attribute value X'00' (default specification). It can, optionally, accept other attribute values. The attribute values listed in the Query Reply are accepted by the device. The following attribute values are the only valid values:

X'00' Default  
X'F0' Normal highlight  
X'F1' Blink  
X'F2' Reverse video  
X'F4' Underscore  
X'F8' Intensify.

This structured field indicates that the device supports highlighting on an exclusive basis. That is, one and only one of the highlight values can be applied to a field or character location.

The code X'00' indicates that the device action for the corresponding attribute value is the same as the action for the attribute value X'00' (the default action of the device).

---

## 6.28.2 Format

The following table shows the format of the Query Reply (Highlighting) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'87' Identifies this Query Reply as Highlighting

4	<b>NP</b>	Number of attribute-value/action pairs
n	<b>Vi</b>	Data stream attribute value accepted
n+1	<b>Ai</b>	Data stream action

The following example illustrates the Query Reply (Highlighting) response:

Byte	Content Description
0-1	<b>X'00C'</b> Length of this structured field
2	<b>X'81'</b> Identifies this structured field as Query Reply
3	<b>X'87'</b> Identifies this query as Highlighting
4	<b>X'04'</b> Number of pairs
5-6	<b>X'00'</b> Attribute value (default) (pair 1) <b>X'F0'</b> Action - normal
7-8	<b>X'F1'</b> Attribute value (pair 2) <b>X'F1'</b> Action - blink
9-10	<b>X'F2'</b> Attribute value (pair 3) <b>X'F2'</b> Action - reverse video
11-12	<b>X'F4'</b> Attribute value (pair 4) <b>X'F4'</b> Action - underscore

## 6.29 Query Reply (IBM Auxiliary Device)

This section describes the Query Reply (IBM Auxiliary Device) structured field.

Subtopics:

- [6.29.1 Function](#)
- [6.29.2 Format](#)
- [6.29.3 Additional Content Description](#)
- [6.29.4 Direct Access ID Self-Defining Parameter](#)
- [6.29.5 Printer Name Self-Defining Parameter](#)
- [6.29.6 PCLK Protocol Controls Self-Defining Parameter](#)

### 6.29.1 Function

This Query Reply structured field tells the host application about the characteristics of IBM auxiliary devices. This Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'9E', Equivalent, or All).

To use this Query Reply, you must have a device that:

- Can be addressed with the Destination/Origin structured field
- Uses a data stream defined by an IBM product or one of the IBM Object Content Architectures.

## 6.29.2 Format

The following table shows the format of the Query Reply (IBM Auxiliary Device) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'9E'</b> Identifies this Query Reply as IBM Auxiliary Device
4		<b>FLAGS</b>	
	0	<b>QUERY</b>	Read Partition (Query or Query List)
			<b>B'1'</b> Device supports Query
	1-7	<b>RES</b>	Reserved; must be zeros
5		<b>FLAG</b>	Reserved; must be zeros
6-7		<b>LIMIN</b>	Inbound message size limit
8-9		<b>LIMOUT</b>	Outbound message size limit
10		<b>TYPE</b>	Type of IBM auxiliary device
			<b>X'01'</b> Display
			<b>X'02'</b> Printer
			<b>Others</b> Reserved
<p><b>Note:</b> All bytes of the IBM Auxiliary Device structured field must be present. If not used, they are set to X'00'.</p>			

## 6.29.3 Additional Content Description

The content of the Query Reply (IBM Auxiliary Device) structured field is further described as follows:

**QUERY** When this bit is set to B'1', the device supports receiving a Read Partition (Query or Query List), so the host application can use a Read Partition to determine the device's characteristics. The Read Partition must be directed to the auxiliary device with the Destination/Origin structured field.

This bit must be set to B'1' for all IBM auxiliary devices. The minimum support for this structured field is to return the Null Query Reply in response to the Read Partition.

**LIMIN** LIMIN is the maximum number of bytes in an inbound transmission. A value of X'0000' indicates no implementation limit on the number of bytes transferred.

**LIMOUT** LIMOUT is the maximum number of bytes in an outbound transmission. A value of X'0000' indicates no implementation limit on the number of bytes transferred.

**TYPE** Type identifies the auxiliary device that is supported. A value of X'01' identifies the device as a display. A value of X'02' identifies that device as a printer. All other values are reserved.

The IBM Auxiliary Device Query Reply accepts three self-defining parameters:

- Direct Access ID
- Printer Name
- PCLK Protocol Controls.

---

## 6.29.4 Direct Access ID Self-Defining Parameter

For the Query Reply to be valid, the Direct Access ID self-defining parameter must be present.

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'04'</b> Parameter length
1	<b>SDPID</b>	<b>X'01'</b> Identifies this self-defining parameter as Direct Access ID
2-3	<b>DOID</b>	Destination/Origin Identification

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** The presence of the Direct Access ID self-defining parameter indicates that the IBM auxiliary device can be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate that the destination or origin of the

following data is the auxiliary device. When more than one auxiliary device is supported, each one has a separate IBM Auxiliary Device Query Reply and a separate ID.

---

## 6.29.5 Printer Name Self-Defining Parameter

The Printer Name self-defining parameter provides a name that uniquely identifies the specified printer. The parameter is valid only if TYPE in the IBM Auxiliary Device Query Reply Base is set to X'02' (IBM Auxiliary Device Printer).

If more than one IBM Auxiliary Device Query Reply for a printer is returned by a device, then each Printer Name self-defining parameter must provide a unique name.

The following table shows the format of the Printer Name self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'0E'</b> Parameter length
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Printer Name
2-3	<b>CSET</b>	Character set used to define PNAME
4-5	<b>CPAGE</b>	Code page used to define PNAME
6-13	<b>PNAME</b>	Printer name

The content of the Printer Name self-defining parameter is further described as follows:

**CSET** A 2-byte Global Graphic Character Identifier (GGID) that identifies the character set used to define PNAME.

**CPAGE** A 2-byte Global Code Page Identifier (GPID) that identifies the code page used to define PNAME.

**PNAME** This parameter contains an 8-byte character string to define the name of the printer.

---

## 6.29.6 PCLK Protocol Controls Self-Defining Parameter

The presence of the PCLK Protocol Controls self-defining parameter indicates that the PCLK Protocol Controls structured field (X'1013') can be used in both inbound and outbound IBM Auxiliary Device data streams.

The following table shows the format of the PCLK Protocol Controls self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'04'</b> Parameter length
1	<b>SDPID</b>	<b>X'03'</b> Identifies this self-defining parameter as PCLK Protocol Controls
2-3	<b>VERS</b>	Protocol version

The content of the PCLK Protocol Controls self-defining parameter is further described as follows:

**VERS** This value indicates the version of GDDM-PCLK installed in the terminal at the time the Query Reply is returned. X'0001' indicates GDDM-PCLK Version 1.1.

---

## 6.30 Query Reply (Image)

This section describes the Query Reply (Image) structured field.

Subtopics:

- [6.30.1 Function](#)
- [6.30.2 Format](#)

---

### 6.30.1 Function

The Query Reply (Image) structured field provides specific information about the device support of image parameter sets and image order sets.

This Query Reply is returned to the host in reply to a Read Partition structured field specifying Query List (QCODE List=X'82' or All).

---

### 6.30.2 Format

The following table shows the format of the Query Reply (Image) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'82'</b> Identifies this Query Reply as Image
4-n	<b>COMMAND</b>	Command

**Note:** For information about the format and operation of the COMMAND parameter, refer to related imaging documentation.

---

## 6.31 Query Reply (Implicit Partition)

This section describes the Query Reply (Implicit Partition) structured field.

Subtopics:

- [6.31.1 Function](#)
- [6.31.2 Format](#)
- [6.31.3 Implicit Partition Sizes for Display Devices Self-Defining Parameter](#)
- [6.31.4 Implicit Partition Sizes for Printer Devices Self-Defining Parameter](#)
- [6.31.5 Implicit Partition Sizes for Character Cell Dimensions Self-Defining Parameter](#)

---

### 6.31.1 Function

The Query Reply (Implicit Partition) structured field defines unique implicit partition characteristics.

This field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

The Implicit Partition Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'A6', Equivalent, or All) by any device supporting EBASE.

---

### 6.31.2 Format

The following table shows the format of the Query Reply (Implicit Partition) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'A6'</b> Identifies this Query Reply as Implicit Partition
4-5	<b>FLAGS</b>	<b>X'0000'</b> Reserved

The Implicit Partition Query Reply accepts the following self-defining parameters:

- Implicit Partition Sizes for Display Devices

This parameter is required for all display devices.

- Implicit Partition Sizes for Printer Devices

This parameter is required for all printer devices.

- Implicit Partition Sizes for Character Cell Dimensions.

This parameter is not required for all Implicit Partition Query Replies. When this parameter is not present, the character cell size for the Implicit Partition is specified in the Usable Area Query Reply.

### 6.31.3 Implicit Partition Sizes for Display Devices Self-Defining Parameter

The Implicit Partition Sizes for Display Devices self-defining parameter informs the host of the default and alternate screen sizes of the Implicit Partition. The sizes are specified in character cells. This parameter is required for all display devices.

When the default and alternate screen sizes are specified, two things must be true:

- Default and alternate values must be nonzero.
- If the device does not have an alternate screen size, the value for the alternate screen size must be that of the default screen size.

The Implicit Partition default and alternate screen sizes are established differently in SNA and non-SNA systems. In SNA systems, the default and alternate sizes returned in this reply are those established at BIND. In non-SNA systems, the default and alternate sizes returned in this reply are those in effect at the time the Query Reply is generated.

The following table shows the format of the Implicit Partition Sizes for Display Devices self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'0B'</b> Length of this self-defining parameter
1	<b>SDPID</b>	<b>X'01'</b> Identifies this self-defining parameter as Implicit Partition Sizes
2	<b>FLAGS</b>	<b>X'00'</b> Reserved
3-4	<b>WD</b>	Width of the Implicit Partition default screen size (in character cells)
5-6	<b>HD</b>	Height of the Implicit Partition default screen size

7-8	<b>WA</b>	Width of the Implicit Partition alternate screen size
9-10	<b>HA</b>	Height of the Implicit Partition alternate screen size

## 6.31.4 Implicit Partition Sizes for Printer Devices Self-Defining Parameter

The Implicit Partition Sizes for Display Devices self-defining parameter informs the host of the default and alternate printer buffer sizes. The sizes are specified in character cells. This parameter is required for all printer devices.

When the default and alternate printer buffer sizes are specified, two things must be true:

- Default and alternate values must be nonzero.
- If the device does not have an alternate print buffer size, the value for the alternate size must be that of the default size.

The Implicit Partition default and alternate printer buffer sizes are established differently between SNA and non-SNA systems. In SNA systems, the default and alternate sizes returned in this reply are those established at BIND. In non-SNA systems, the default and alternate sizes returned in this query reply are those in effect at the time that the Query Reply is generated.

The buffer size defines the following printer buffer restrictions:

- The maximum linear character buffer address that can be explicitly specified in 3270 orders. (The maximum buffer address is one less than the buffer size in character cells.)
- The wrapping point for the transmitted data.

If the implied address for the data being loaded into the character buffer exceeds the maximum address allowed by the buffer size, then the implied address is reset to zero and loading continues from the first buffer location.

The following table shows the format of the Implicit Partition Sizes for Printer Devices self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'0B'</b> Length of this self-defining parameter
1	<b>SDPID</b>	<b>X'03'</b> Identifies this self-defining parameter as Implicit Partition Sizes
2	<b>FLAGS</b>	<b>X'00'</b> Reserved

3-6	<b>DPBS</b>	Default printer buffer size (in character cells)
7-10	<b>APBS</b>	Alternate printer buffer size

### 6.31.5 Implicit Partition Sizes for Character Cell Dimensions Self-Defining Parameter

The Implicit Partition Sizes for Character Cell Dimensions self-defining parameter informs the host system of the character cell sizes associated with the default and alternate Implicit Partition sizes. This parameter is not required for all Implicit Partition Query Replies. Use this parameter only if the following conditions exist:

- The cell size associated with either the default or alternate screen size for the Implicit Partition is different from the cell size that is reported in the Usable Area Query Reply parameters (AW/AH).
- The device supports the Load PS structured field.

Character Cells are measured in this Query Reply in the same way as in the Usable Area Query Reply parameters (UNITS, Xr, and Yr).

When this parameter is present, the default or alternate character cell sizes are determined by the following three factors:

- The Implicit Partition default screen size uses the default character cell size.
- The Implicit Partition alternate screen size uses the alternate character cell size.
- The default character cell size for the Explicit Partition is stated in the Usable Area Query Reply parameters (AW/AH). (There is no alternate size in Explicit Partition state.)

The following table shows the format of the Implicit Partition Sizes for Character Cell Dimensions self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'0B'</b> Length of this self-defining parameter
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Implicit Partition Sizes
2	<b>FLAGS</b>	<b>X'00'</b> Reserved
3-4	<b>WCD</b>	Width of character cell for the Implicit Partition default screen size
5-6	<b>HCD</b>	Height of character cell for the Implicit Partition default screen size
7-8	<b>WCA</b>	Width of character cell for the Implicit Partition alternate screen size

9-10	HCA	Height of character cell for the Implicit Partition alternate screen size
------	-----	---

## 6.32 Query Reply (IOCA Auxiliary Device)

This section describes the Query Reply (IOCA Auxiliary Device) structured field.

Subtopics:

- [6.32.1 Function](#)
- [6.32.2 Format](#)
- [6.32.3 Additional Content Description](#)

### 6.32.1 Function

The Query Reply (IOCA Auxiliary Device) structured field indicates support of a 3270 workstation auxiliary device that uses the Image Object Content Architecture (IOCA) data stream.

This Query Reply is returned to the host as a result of receiving a Read Partition structured field specifying Query List (QCODE List=X'AA' or All).

When a workstation supports multiple IOCA auxiliary devices, an IOCA Auxiliary Device Query Reply must be sent for each of them.

### 6.32.2 Format

The following table shows the format of the Query Reply (IOCA Auxiliary Device) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'AA' Identifies this Query Reply as IOCA Auxiliary Device
4-5	FLAGS	Reserved
6-7	LIMIN	Max IOCA bytes/inbound transmission
8-9	LIMOUT	Max IOCA bytes/outbound transmission
10	IOTYPE	X'00' Type of device X'01' Input device X'02' Output device

---

**Note:** For information about the format and operation of the COMMAND parameter, refer to related imaging documentation.

---

### 6.32.3 Additional Content Description

The content of the Query Reply (IOCA Auxiliary Device) structured field is further described as follows:

**LIMIN** The sum of bytes contained in all of the structured fields that are associated with the IOCA Auxiliary Device following an AID 88, is equal to or less than the value specified (in hexadecimal) in LIMIN. A LIMIN value of X'0000' indicates no implementation limit on inbound IOCA data.

**LIMOUT** The sum of bytes contained in all the IOCA structured fields associated with the IOCA auxiliary device-- following a WSF command-- must not exceed the LIMOUT value. If this limit is exceeded, the transmission must be rejected. The data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound IOCA data.

**IOTYPE** This specifies the type of device. There are three types of devices: input, output, and input/output.

For information about self-defining parameters, refer to related image documentation.

---

## 6.33 Query Reply (Line Type)

This section describes the Query Reply (Line Type) structured field.

Subtopics:

- [6.33.1 Function](#)
- [6.33.2 Format](#)

---

### 6.33.1 Function

The Query Reply (Line Type) structured field indicates which line type attribute values are supported and the corresponding device action. Also, for loadable line types, it indicates which formats of line type definition are supported, as well as an indication of which, if any, code points are already loaded.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B2' or All).

---

## 6.33.2 Format

The following table shows the format of the Query Reply (Line Type) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'B2'</b> Identifies this Query Reply as Line Type
4-n	<b>DATA</b>	Data

**Note:** For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

---

## 6.34 Query Reply (MSR Control)

This section describes the Query Reply (MSR Control) structured field.

Subtopics:

- [6.34.1 Function](#)
- [6.34.2 Format](#)

---

### 6.34.1 Function

The Query Reply (MSR Control) structured field defines the number of magnetic slot reader (MSR) devices that are attached and specifies the magnetic slot reader type that the application program can use to determine the control requirements of the device.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'8B', Equivalent, or All).

---

### 6.34.2 Format

The following table shows the format of the Query Reply (MSR Control) structured field:

Byte	Content	Content Description
------	---------	---------------------

0-1	<b>L</b>	<b>X'0007'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'8B'</b> Identifies this Query Reply as MSR Control
4	<b>FLAGS</b>	Reserved
5	<b>ND</b>	Number of MSR device types
6	<b>TYPE</b>	MSR type:
		<b>X'01'</b> Magnetic reader
		<b>Other</b> Reserved

---

## 6.35 Query Reply (Null)

This section describes the Query Reply (Null) structured field.

Subtopics:

- [6.35.1 Function](#)
- [6.35.2 Format](#)

---

### 6.35.1 Function

The Query Reply (Null) structured field informs the host that the device does not support any of the features or functions that the host inquired about with the Read Partition structured field specifying Query List (QCODE List). If the host queries the device and the device supports at least one feature or function that was queried about, then the Null Query Reply is not sent to the host.

This Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'FF'), as shown in the following examples:

**Example 1:**

- A device supports features A, B, and C.
- The host queries for features A, X, and Z.

The device does not send the Null Query Reply. It sends the Query Reply for feature A only. Therefore, the host knows that the device does not support features X and Z.

**Example 2:**

- A device supports features A, B, and C.
- The host queries for features X, Y, and Z.

The device sends the Null Query Reply because the device does not support any of the requested features.

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

## 6.35.2 Format

The following table shows the format of the Query Reply (Null) structured field:

Byte	Content	Content Description
0-1	L	X'0004' Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'FF' Identifies this Query Reply as Null

## 6.36 Query Reply (OEM Auxiliary Device)

This section describes the Query Reply (OEM Auxiliary Device) structured field.

Subtopics:

- 6.36.1 Function
- 6.36.2 Format
- 6.36.3 Additional Content Description
- 6.36.4 Direct Access ID Self-Defining Parameter
- 6.36.5 LIMIN/LIMOUT OEM Auxiliary Controls Self-Defining Parameter
- 6.36.6 PCLK Protocol Controls Self-Defining Parameter

### 6.36.1 Function

The Query Reply (OEM Auxiliary Device) structured field indicates support of an OEM auxiliary device (see Chapter 11, "Auxiliary Devices and Workstations"). An OEM device is defined here as a device that is manufactured outside of IBM and does not use an IBM-defined data stream. The device may carry either an outside manufacturer logo or an IBM logo.

When the function is supported, the OEM Auxiliary Device Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'8F', Equivalent, or All).

When a workstation supports multiple OEM devices, an OEM Auxiliary Device Query Reply must be sent for each of the devices.

All parameters shown in the base part of the OEM Auxiliary Device Query Reply must be present.

---

### 6.36.2 Format

The following table shows the format of the Query Reply (OEM Auxiliary Device) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'8F'</b> Identifies this Query Reply as OEM Auxiliary Device
4	<b>FLAGS</b>	Reserved
5	<b>DSREF</b>	Data stream reference identifier
6-13	<b>DTYPE</b>	Device type
14-21	<b>UNAME</b>	User assigned name

---

### 6.36.3 Additional Content Description

The content of the Query Reply (OEM Auxiliary Device) structured field is further described as follows:

**DTYPE** This is an 8-byte character string used to send the device type to the host application. An understanding must exist between the workstation and the host on the relationship of the DTYPE value and the device characteristics (for example, what data the device accepts or sends).

**DSREF** This parameter is used to indicate what data is contained in an OEM Data structured field directed to or from the OEM auxiliary device.

A value of X'00' indicates the data in the OEM Data structured field must be the data stream the device recognizes or sends as derived from the DTYPE parameter.

A nonzero value indicates that the OEM Data structured field contains a value added data stream. That is, the OEM Data structured field contains controls and other data, in addition to the data recognized or sent by the device. Further workstation processing of this value added data stream is required before sending the data to the OEM auxiliary device. Refer to your related product documentation for a description of the value added

data stream associated with the DSREF parameter.

Valid values for the DSREF parameter (in addition to X'00') are as follows:

X'01' Plotter Type 1  
All other values reserved.

**UNAME** This parameter contains an 8-byte character string name provided by the workstation. The purpose is to provide a user friendly name (for example, plotter1, plotter2) for use by the application. This name will not appear in the data stream between the host and the workstation. A value of all zeros indicates no name assigned. The value X'FF FF' is reserved.

The Query Reply (OEM Auxiliary Device) structured field accepts three self-defining parameters:

- Direct Access ID
- LIMIN/LIMOUT OEM Auxiliary Controls
- PCLK Protocol Controls.

---

### 6.36.4 Direct Access ID Self-Defining Parameter

The Direct Access ID self-defining parameter provides the ID for use in the Destination/Origin structured field in the direct access of the OEM auxiliary device. It must be present.

The following table shows the format of the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Length of this parameter
1	SDPID	X'01' Identifies this self-defining parameter as Direct Access ID
2-3	DOID	Destination/Origin ID

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** The value in this byte is used in the ID field of the Destination/Origin structured field to identify the auxiliary device as the destination or origin of the data that follows.

---

### 6.36.5 LIMIN/LIMOUT OEM Auxiliary Controls Self-Defining Parameter

The LIMIN/LIMOUT OEM Auxiliary Controls self-defining parameter indicates the maximum number of bytes allowed in an OEM Data structured field to or

from the OEM auxiliary device.

The following table shows the format of the LIMIN/LIMOUT OEM Auxiliary Controls self-defining parameter:

Byte	Content	Content Description
0	L	X'06' Parameter Length
1	SDPID	X'02' Identifies this self-defining parameter as LIMIN/LIMOUT OEM Auxiliary Controls
2-3	LIMIN	Maximum OEM dsf bytes/transmission allowed inbound
4-5	LIMOUT	Maximum OEM dsf bytes/transmission allowed outbound

The content of the LIMIN/LIMOUT OEM Auxiliary Controls self-defining parameter is further described as follows:

**LIMIN** Certain implementations may have a limit on the number of bytes of OEM information that are allowed inbound following an AID88. The number of bytes in an inbound OEM transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on inbound OEM data.

**LIMOUT** The sum of bytes contained in all the OEM structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission is rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound OEM data.

The LIMOUT parameter applies only to the OEM structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by an OEM sf (100 bytes), 3270E sf (500 bytes), and an OEM sf (300 bytes) would be accepted.

---

## 6.36.6 PCLK Protocol Controls Self-Defining Parameter

The presence of the PCLK Protocol Controls self-defining parameter indicates that the GDDM-PCLK Protocol Controls structured field (X'1013') can be used in both inbound and outbound OEM Auxiliary Device data streams.

The following table shows the format of the PCLK Protocol Controls self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Parameter Length
1	SDPID	X'03' Identifies this self-defining parameter as GDDM-PCLK Protocol Controls

2-3	<b>VERS</b>	Protocol version
-----	-------------	------------------

The content of the PCLK Protocol Controls self-defining parameter is further described as follows:

**VERS** This value indicates the version of GDDM-PCLK installed in the terminal at the time the Query Reply is returned. X'0001' indicates GDDM-PCLK version 1.1.

## 6.37 Query Reply (Paper Feed Techniques)

This section describes the Query Reply (Paper Feed Techniques) structured field.

Subtopics:

- [6.37.1 Function](#)
- [6.37.2 Format](#)
- [6.37.3 Additional Content Description](#)

### 6.37.1 Function

The Query Reply (Paper Feed Techniques) structured field transmits the currently installed and active paper feed technique (for example, cut sheet automatic). It transmits the size of the restricted print areas (if any) at the top and/or bottom of the presentation surface (form). If any area of the presentation surface (form) is restricted, a device must support this Query Reply.

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'A7', Equivalent, or All).

The top and bottom margin offsets define the areas at the top and bottom of the presentation surface (form) where printing should not occur. The top and bottom margin offsets can range from 0 (meaning there is no restricted area) to 65 535/1440ths of an inch (45.51 inches).

### 6.37.2 Format

The following table shows the format of the Query Reply (Paper Feed Techniques) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	<b>X'0009'</b> Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply

3		<b>QCODE</b>	<b>X'A7'</b> Identifies this Query Reply as Paper Feed Techniques
4		<b>FLAGS</b>	
	0-2	<b>PRMECH</b>	<b>B'000'</b> Unknown <b>B'001'</b> Cut sheet manual <b>B'010'</b> Continuous form <b>B'011'</b> Cut sheet automatic <b>B'100'</b> Document on demand <b>Other</b> Values reserved
	3-7	<b>RES</b>	<b>B'00000'</b> Reserved
5-6		<b>TMO</b>	<b>X'nnnn'</b> Top margin offset in 1/1440ths of an inch
7-8		<b>BMO</b>	<b>X'mmmmm'</b> Bottom margin offset in 1/1440ths of an inch

### 6.37.3 Additional Content Description

**FLAGS** When this bit is set to B'100', then document on demand is similar to continuous forms except that printing stops automatically after the current page is ejected. The operator can tear off the current form any time after printing stops. Printing resumes when the operator presses the Enable (Start) key.

**TMO** The top margin offset is the top margin that must be used if the restricted area at the top of the presentation surface is to be avoided. The top margin offset is measured from the top edge of the presentation surface to the base line of the first allowable print line.

Except when the printer is operating in the 3270 mode, the action taken when printing is attempted in the top restricted area is device dependent (that is, actions could include a forced movement to the first safe line, an attempt to print, or a rejected request). In effect, to obtain predictable results, the application must avoid attempting to print in the top restricted area.

Other than lines of nulls or lines of spaces, the application can use any of the means provided by the data stream to move through the top restricted area without printing; for example:

- Set the TM parameter of SVF control to a value equal to or greater than the first allowable print line.
- Move from top of the presentation surface to the first allowable print line (or greater) using new lines (NLs), line feeds (LFs), or vertical tabs (VTs).

When the printer is operating in the 3270 mode (LU3 or non-SNA) the printer must protect against printing in the top restricted area.

An application attempt to print in the top restricted area results in a forced move to the first safe print line. The number of lines moved equal the line equivalent of the distance given in the top offset field of the Paper Feed Technique Query Reply; the line counter is incremented for each line of forced movement.

**BMO** The bottom margin offset is the bottom margin that must be used if the restricted area at the bottom of the presentation surface is to be avoided. The bottom margin offset is measured from the bottom edge of the presentation surface to the base line of the last allowable print line.

The action taken when an application attempts to print in the bottom restricted area is device dependent (that is, actions could include a forced move to the next form, an attempt to print, or a rejected request). In effect, to obtain predictable results, an application must avoid printing in the bottom restricted area.

Other than lines of nulls and lines of spaces, the application can use any of the means provided by the data stream to avoid printing in the bottom restricted area; for example:

- Use of a form feed (FF) prior to reaching the bottom restricted area
- Use of new lines (NLs), line feeds (LFs), or vertical tabs (VTs) to move through the bottom restricted area.

When operating in 3270 mode, the means of avoiding the bottom restricted area are somewhat limited. VTs and LFs are not valid, and the FF/NL controls are not always valid. Therefore, the user should ensure that the paper loaded in the printer is long enough to accommodate the complete buffer contents.

---

## 6.38 Query Reply (Partition Characteristics)

This section describes the Query Reply (Partition Characteristics) structured field.

Subtopics:

- 6.38.1 Function
- 6.38.2 Format
- 6.38.3 Viewport Outline Parameters Self-Defining Parameters
- 6.38.4 Enable User Call-up Self-Defining Parameter

---

### 6.38.1 Function

The Query Reply (Partition Characteristics) structured field specifies that the device supports the Partition Characteristics Structured Field. When the function is supported, this Query Reply is transmitted inbound in reply to a Query List specifying a QCODE of X'8E' or All.

The Partition Characteristics Query Reply has the following self-defining parameters:

- Viewport Outline Parameters
- Enable User Call-up.

---

### 6.38.2 Format

The following table shows the format of the Query Reply (Partition Characteristics) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structure
2	<b>SFID</b>	<b>X'81'</b> Query Reply
3	<b>QCODE</b>	<b>X'8E'</b> Identifies this Query Reply as Partition Characteristics

---

### 6.38.3 Viewport Outline Parameters Self-Defining Parameters

The Viewport Outline Parameters self-defining parameter indicates that viewport outlining is supported by the device.

The following table shows the format of the Viewport Outline Parameters self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'03'</b> Length of this structure
1	<b>SDPID</b>	<b>X'01'</b> Identifies this self-defining parameter as Viewport Outline Parameters
2	<b>THICKNESS</b>	Maximum outline thickness supported

---

### 6.38.4 Enable User Call-up Self-Defining Parameter

The Enable User Call-up self-defining parameter indicates that the Enable User Call-up function of the Set Partition Characteristics structured field is supported by the device.

The following table shows the format of the Enable User Call-up self-defining parameter:

Byte	Content	Content Description
------	---------	---------------------

0	L	X'02'	Length of this self-defining parameter
1	SDPID	X'03'	Identifies this self-defining parameter as Enable User Call-up

## 6.39 Query Reply (Port)

This section describes the Query Reply (Port) structured field.

Subtopics:

- [6.39.1 Function](#)
- [6.39.2 Format](#)

### 6.39.1 Function

The Query Reply (Port) structured field defines which ports are supported.

For each port supported, an appropriate, separate Query Reply is returned in response to a Read Partition structured field specifying Query List (QCODE List=X'B3' or All).

### 6.39.2 Format

The following table shows the format of the Query Reply (Port) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'B3' Identifies this Query Reply as Port
4-n	DATA	Data

**Note:** For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

## 6.40 Query Reply (Procedure)

This section describes the Query Reply (Procedure) structured field.

Subtopics:

- [6.40.1 Function](#)
- [6.40.2 Format](#)

## 6.40.1 Function

The Query Reply (Procedure) structured field indicates at which graphic subset level graphic procedures are supported.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B1' or All).

## 6.40.2 Format

The following table shows the format of the Query Reply (Procedure) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'B1' Identifies this Query Reply as Procedure
4-n	DATA	Data

**Note:** For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

## 6.41 Query Reply (Product Defined Data Stream)

This section describes the Query Reply (Product Defined Data Stream) structured field.

Subtopics:

- [6.41.1 Function](#)
- [6.41.2 Format](#)
- [6.41.3 Additional Content Description](#)
- [6.41.4 Direct Access ID Self-Defining Parameter](#)

### 6.41.1 Function

This Query Reply indicates support of a 3270DS workstation auxiliary device that uses an IBM product defined data stream (PDDS).

When an auxiliary device supports an IBM product defined data stream, the Query Reply is transmitted inbound in reply to a Query List (QCODE List=X'9C' or All).

All bytes of this structured field and the Direct Access ID parameter must be present.

---

## 6.41.2 Format

The following table shows the format of the Query Reply (Product Defined Data Stream) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'000C'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'9C'</b> Identifies this Query Reply as Product Defined Data Stream
4-5	<b>FLAGS</b>	Reserved; must be zeros
6	<b>REFID</b>	Reference identifier
7	<b>SSID</b>	Subset Identifier

---

## 6.41.3 Additional Content Description

The content of the Query Reply (Product Defined Data Stream) structured field is further described as follows:

**REFID** This byte contains a reference number that indicates the particular IBM PDDS and the documentation describing that data stream.

**SSID** This byte contains a reference number that indicates which subset of the PDDS identified by REFID is supported by the auxiliary device. The subset must be defined in the product documentation. If no subsets exist inside the PDDS, the value of SSID is X'00'.

The values of REFID and SSID can be any of the combinations described in the following table:

REFID	SSID	Product and Data Stream Documentation
X'01'		This reference ID indicates that the auxiliary device supports the 5080 Graphics System. The 5080 Graphics Architecture, structured fields, subset

		IDs, DOID, and associated function sets are described in <i>IBM 5080 Graphics System Principles of Operation</i> .
	X'01'	5080 HFGD Graphics Subset
	X'02'	5080 RS232 Ports Subset
X'02'		This reference ID indicates that the auxiliary device supports the WHIP API data stream. The WHIP API architecture is described in <i>IBM AIX Workstation Host Interface Program User's Guide and Reference Manual, Version 1.1</i> .
	X'01'	WHIP Subset 1
X'03' to X'FF'		Reserved

## 6.41.4 Direct Access ID Self-Defining Parameter

The Direct Access ID parameter provides the ID for use in the Destination/Origin structured field in the direct access of the auxiliary device. It must be present.

When multiple auxiliary devices that use a product defined data stream are supported, a separate Product Defined Data Stream Query Reply with a unique DOID must be provided.

The following table shows the format for the Direct Access ID self-defining parameter:

Byte	Content	Content Description
0	L	X'04' Length of this parameter
1	SDPID	X'01' Identifies this self-defining parameter as Direct Access ID
2-3	DOID	Destination/Origin ID

The content of the Direct Access ID self-defining parameter is further described as follows:

**DOID** The value in this byte is used in the ID field of the Destination/Origin structured field to identify the auxiliary device as the destination or origin of the data that follows.

## 6.42 Query Reply (Reply Modes)

This section describes the Query Reply (Reply Modes) structured field.

Subtopics:

- [6.42.1 Function](#)
- [6.42.2 Format](#)
- [6.42.3 Additional Content Description](#)

---

## | 6.42.1 Function

| The Query Reply (Reply Modes) is used to inform the host of the following:

- | ° The 3270 Data Stream attribute orders that the device supports, in both the inbound and the outbound directions.
- | ° The modes that can be set by the Set Reply Modes structured field.

| When this function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE list=X'88', Equivalent, or All). This query reply is not transmitted if *only* the Field Mode is supported.

| For a description of reply modes, see ["Set Reply Mode" in topic 5.30](#).

---

## 6.42.2 Format

The following table shows the format of the Query Reply (Reply Modes) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'88'</b> Identifies this Query Reply as Reply Modes
4-n	<b>MODES</b>	Modes supported:  <b>X'00'</b> Field mode <b>X'01'</b> Extended field mode <b>X'02'</b> Character mode <b>Other</b> Reserved

---

## | 6.42.3 Additional Content Description

| The content of the Query Reply (Reply Modes) structured field is further

described as follows:

**Modes** This field contains one byte for each of the modes supported. The minimum entry is X'0001' (X'00' for Field and X'01' for Extended Field) because this query reply is first used when extended fields are supported.

The following is a list of modes and the Orders that they use:

- The Field mode indicates support of the Start Field (SF) order. See "Start Field (SF)" in topic 4.3.1.
- The Extended Field mode indicates support of the Start Field Extended (SFE) and the Modify Field (MF) orders. See "Start Field Extended (SFE)" in topic 4.3.2 and "Modify Field (MF)" in topic 4.3.5.
- The Character mode indicates support of the Set Attribute order. See "Set Attribute (SA)" in topic 4.3.4.

#### **Examples**

- If the device supports the Field and Extended Field attributes, then the device sets the MODES field to X'0001'.
- If the device supports the Field, the Extended Field, and the Character attributes, then the device sets the MODES field to X'000102'.

---

## 6.43 Query Reply (RPQ Names)

This section describes the Query Reply (RPQ Names) structured field.

Subtopics:

- 6.43.1 Function
- 6.43.2 Format

---

### 6.43.1 Function

The Query Reply (RPQ NAMES) structured field tells the application which RPQs are initialized for use in the display. If appropriate, RPQ-dependent information is supplied for each.

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'A1', Equivalent, or All).

---

### 6.43.2 Format

The following table shows the format of the Query Reply (RPQ Names) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'A1'</b> Identifies this Query Reply as RPQ Names
4-7	<b>DEVICE</b>	Device type identifier
8-11	<b>MODEL</b>	Model type identifier (X'00000000' = all models)
12	<b>RPQL</b>	RPQ length (length of RPQ Name + RPQ)
13-n	<b>RPQID</b>	RPQ name

For example, the following table describes the RPQ Query Reply sent by device type 8775 supporting RPQ SU0183 only:

Byte	Content	Content Description
0-1		<b>X'0013'</b> Length of this structured field
2		<b>X'81'</b> Identifies this structured field as Query Reply
3		<b>X'A1'</b> Identifies this query as RPQ Names
4-7		<b>C'8775'</b> Device type identifier
8-11		<b>AL4(0)</b> Model type identifier (X'00000000' = all models)
12		<b>X'07'</b> Length of RPQ name (including this byte)
13-18		<b>C'SU0183'</b> RPQ name

---

## 6.44 Query Reply (Save/Restore Format)

This section describes the Query Reply (Save/Restore Format) structured field.

Subtopics:

- [6.44.1 Function](#)
- [6.44.2 Format](#)

---

## 6.44.1 Function

The Query Reply (Save/Restore Format) structured field indicates that the secondary device supports the Save/Restore Format structured field.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'92' or All).

---

## 6.44.2 Format

The following table shows the format of the Query Reply (Save/Restore Format) structured field:

Byte	Content	Content Description
0-1	L	X'0006' Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'92' Identifies this Query Reply as Save/Restore Format
4-5	FPCBL	Format parameter control block length: a binary count that designates the length of the Format Parameter Control Block (FPCB). The FPCB length is fixed for a given implementation, that is, length does not vary with the specific SCS control codes used within a session.

---

## 6.45 Query Reply (Segment)

This section describes the Query Reply (Segment) structured field.

Subtopics:

- [6.45.1 Function](#)
- [6.45.2 Format](#)

---

### 6.45.1 Function

The Query Reply (Segment) structured field indicates the graphic subset level at which graphic segments are supported.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'B0' or All).

---

## 6.45.2 Format

The following table shows the format of the Query Reply (Segment) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'B0'</b> Identifies this Query Reply as Segment
4-n	<b>DATA</b>	Data

For a description of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

---

## 6.46 Query Reply (Settable Printer Characteristics)

This section describes the Query Reply (Settable Printer Characteristics) structured field.

Subtopics:

- [6.46.1 Function](#)
- [6.46.2 Format](#)
- [6.46.3 Early Print Complete \(EPC\) Self-Defining Parameter](#)

---

### 6.46.1 Function

The Query Reply (Settable Printer Characteristics) structured field indicates support of one or more characteristics such as printer functions and modes that can be set or reset by use of the SPC structured field. Self-defining parameters are used to describe each characteristic supported. This structured field flows inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'A9' or All).

---

### 6.46.2 Format

The following table shows the format of the Query Reply (Settable Printer Characteristics) structured field:

Byte	Content	Content Description
------	---------	---------------------

0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'A9'</b> Identifies this Query Reply as Settable Printer Characteristics
4-5	<b>FLAGS</b>	Reserved

The Early Print Complete (EPC) self-defining parameter must be present. See "Early Print Complete Self-Defining Parameter" in topic 5.29.3

### 6.46.3 Early Print Complete (EPC) Self-Defining Parameter

The EPC parameter supports the early print complete mode of operation. When the early print complete mode is set, it allows a 3270 printer to indicate that it is ready for more data before printing is actually completed. The overlapping of load and print can improve throughput for certain situations involving high-speed printers.

The following table shows the format of the EPC self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'03'</b> Length of this parameter
1		<b>SDPID</b>	<b>X'01'</b> EPC
2		<b>FLAGS</b>	
	0 - 1	POC	Printer operator control
			<b>B'00'</b> No printer operator control <b>B'01'</b> Printer operator control EPC set off <b>B'10'</b> Printer operator control EPS set on <b>B'11'</b> Reserved
	2 - 7	RES	Reserved

The content of the EPC self-defining parameter is further described as follows:

**POC** Printer Operator Control indicates whether the implementation provides printer operator control of the set/reset of the EPC mode. It also indicates the operator selection when the query list was received.

### 6.47 Query Reply (Storage Pools)

This section describes the Query Reply (Storage Pools) structured field.

Subtopics:

- [6.47.1 Function](#)
- [6.47.2 Format](#)
- [6.47.3 Storage Pool Characteristics Self-Defining Parameter](#)

---

## 6.47.1 Function

The Query Reply (Storage Pools) structured field identifies the storage pools in the device. For each storage pool, there is a self-defining parameter describing the characteristics of that storage pool. These characteristics are the total size of that storage pool when empty, the amount of that storage pool available for additional objects, and a list of identifiers of the types of objects housed in that storage pool.

When the function is supported, this form of Query Reply structured field is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'96' or All).

---

## 6.47.2 Format

The following table shows the format of the Query Reply (Storage Pools) structured field:

Byte	Content	Content Description
0-1	L	Length of this structured field
2	SFID	X'81' Identifies this structured field as a Query Reply
3	QCODE	X'96' Identifies this Query Reply as Storage Pools

---

## 6.47.3 Storage Pool Characteristics Self-Defining Parameter

The following table shows the format of the Storage Pool Characteristics self-defining parameter:

Byte	Content	Content Description
0	L	Length of this self-defining parameter
1	SDPID	X'01' Identifies this self-defining parameter as Storage Pool Characteristics
2	SPID	Storage pool identity

3-6	<b>SIZE</b>	Size of this storage pool when empty
7-10	<b>SPACE</b>	Space available in this storage pool
11-n	<b>OBJLIST</b>	Identifiers of objects housed in this storage pool

The content of the Storage Pool Characteristics self-defining parameter is further described as follows:

**SIZE** Size is the total size of the storage pool in bytes; for example, the size when it contains no objects.

**SPACE** Space is the amount of storage in the pool remaining in bytes; that is, the amount available to house additional objects.

**OBJLIST** OBJLIST is a list of the 2-byte identifiers of the objects that are housed in this storage pool. The following identifiers are from the registry of object identifiers:

Object	Identifier
Segment	X'0001'
Procedure	X'0002'
Extended drawing routine	X'0003'
Data unit	X'0004'
Temporary	X'0005'
Line type	X'0006'
Symbol set	X'0007'.

---

## 6.48 Query Reply (Summary)

This section describes the Query Reply (Summary) structured field.

Subtopics:

- [6.48.1 Function](#)
- [6.48.2 Format](#)

---

### 6.48.1 Function

The Query Reply (Summary) structured field provides a list of the device-supported Query Replies (QCODES) that can be used by the host in a Read Partition Query List (QCODE List). All of the QCODES that are supported by the device are included in the Summary Query Reply (except the Null Query Reply QCODE).

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and [Appendix E, "Functions Required for Systems Application Architecture \(SAA\) Support."](#)

The Summary Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query, or Query List (QCODE List=X'80', Equivalent, or All).

---

## 6.48.2 Format

The following table shows the format of the Query Reply (Summary) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'80'</b> Identifies this Query Reply as Summary Query Reply
4-N	<b>LIST</b>	List of supported QCODES

---

## 6.49 Query Reply (Text Partitions)

This section describes the Query Reply (Text Partitions) structured field.

Subtopics:

- [6.49.1 Function](#)
- [6.49.2 Format](#)
- [6.49.3 Additional Content Description](#)

---

### 6.49.1 Function

The Query Reply (Text Partitions) structured field defines the text partition support.

The maximum partition size is the guaranteed size of the text buffer. This means that if the host restricts an outbound text structured field to this size, it is guaranteed to fit in the buffer. However, if the outbound text structured field exceeds this size, it does *not* necessarily mean that the transmission will fail.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'83', Equivalent, or All).

---

### 6.49.2 Format

The following table shows the format of the Query Reply (Text Partitions) structured field:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'83'</b> Identifies this Query Reply as Text Partitions
4		<b>NT</b>	Maximum number of text partitions
5-6		<b>M</b>	Maximum partition size
7		<b>FLAGS</b>	
	0	VERTSCR	<b>B'0'</b> Vertical scrolling not supported <b>B'1'</b> Vertical scrolling supported
	1	RES	Reserved
	2	RES	Reserved
	3	APADDR	<b>B'0'</b> AP addressability not supported <b>B'1'</b> AP addressability supported
	4	PARPROT	<b>B'0'</b> Partition protection not supported <b>B'1'</b> Partition protection supported
	5	RES	Reserved
	6	MODPROT	<b>B'0'</b> Modify partition not supported <b>B'1'</b> Modify partition supported
	7	RES	Reserved
8		<b>NTT</b>	Number of text types supported
9-N		<b>TLIST</b>	List of types supported:
			<b>X'01'</b> Type 1 Text <b>Others</b> Reserved

### 6.49.3 Additional Content Description

The content of the Query Reply (Text Partitions) structured field is further described as follows:

**NT** NT specifies the number of text partitions supported. The range of text PIDs supported is 0 to N-1, where N is the total number of partitions supported. (See also "Query Reply (Alphanumeric Partitions)" in topic 6.9.)

**FLAGS** The scrolling flags indicate whether the device supports vertical scrolling. Support of scrolling implies the following:

- Support for the Set Window Origin structured field with a change of the row.
- Possible support of local window movement by the operator. The operator can move the window from the host-specified position.

The Modify Partition flag indicates whether the device supports the Modify Partition structured field. If the vertical scrolling flag is also set, the Modify Partition structured field can be used to change the window row.

**TLIST** TLIST is a list of 1-byte values showing which types of text are supported. It has NTT entries, where NTT is the number of text types supported.

---

## 6.50 Query Reply (Transparency)

This section describes the Query Reply (Transparency) structured field.

Subtopics:

- 6.50.1 Function
- 6.50.2 Format
- 6.50.3 Additional Content Description

---

### 6.50.1 Function

The Query Reply (Transparency) structured field identifies the type of transparency supported by the device.

When the function is supported, this form of the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'A8' or All).

---

### 6.50.2 Format

The following table shows the format of the Query Reply (Transparency) structured field:

---

Byte	Content	Content Description
------	---------	---------------------

---

0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'A8'</b> Identifies this Query Reply as Transparency
4	<b>NP</b>	Number of pairs
5	<b>V(i)</b>	Data stream attribute value accepted
6	<b>A(i)</b>	Associated action value
7-N	<b>SFID</b>	Additional V(i)/A(i) pairs as needed

### 6.50.3 Additional Content Description

The content of the Query Reply (Transparency) structured field is further described as follows:

**NP** The number of value/action pairs in this Query Reply.

**V(i)** The background transparency attribute values valid for this device. For each accepted value, the Query Reply returns an action value that indicates the device action associated with that value.

**A(i)** The valid action values accepted by the device. The action values currently defined are listed in the A(i) parameter. The device actions currently defined for each action value are as follows:

Value	Device Action
X'00'	Default
X'F0'	Normal (background transparent) - OR
X'F1'	Background is transparent - XOR
X'FF'	Non-transparent (opaque).

An action value of X'00' indicates that the device action for the corresponding attribute value is the same as for the default attribute value.

## 6.51 Query Reply (Usable Area)

This section describes the Query Reply (Usable Area) structured field.

Subtopics:

- [6.51.1 Function](#)
- [6.51.2 Usable Area Query Reply Base](#)
- [6.51.3 On Pels Limit Self-Defining Parameter](#)
- [6.51.4 Multiple Usable Area Self-Defining Parameter](#)
- [6.51.5 Alternate Usable Area Self-Defining Parameter](#)

---

## 6.51.1 Function

The Query Reply (Usable Area) structured field defines the size and other characteristics of the display surface that can be used for partition viewports (*usable area*) independent of the data type. For example, this structured field defines default size of a character cell. It also defines the following:

### Character Slot Size

The character slot size, specified in the Character Sets Query Reply, is mapped into the character cell size that is defined in the Usable Area Query Reply.

### Variable Character Cell Size

If a device supports variable character cell sizes, an explicit partition's character cell size can be defined by the host in the Create Partition structured field. In this case, the host-specified value overrides the default character cell size described in the Usable Area Query Reply.

### Printers

Printers use the fields in the Usable Area Query Reply differently than displays. When a printer uses a field differently than displays, there will be a note to define how the printer uses the field.

If a printer is a page printer, consider the following:

- Print data is not immediately placed on the paper. It resides in a volatile internal storage area, and unexpected losses of power can cause loss of printed data that would print on non-page printers.
- Page printers may have recovery resources that eliminate the need to resend print data after an intervention-required condition.

The Usable Area Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'81', Equivalent, or All).

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "Functions Required for Systems Application Architecture (SAA) Support."

The minimum support for the Usable Area Query Reply consists of the Usable Area Query Reply Base and the self-defining parameters that are appropriate to the device.

The Usable Area Query Reply has three self-defining parameters:

- On Pels Limit
- Multiple Usable Area
- Alternate Usable Area.

## 6.51.2 Usable Area Query Reply Base

This section describes the Usable Area Query Reply Base.

Subtopics:

- [6.51.2.1 Format](#)
- [6.51.2.2 Additional Content Description](#)

### 6.51.2.1 Format

The following table shows the format of the Usable Area Query Reply Base:

Byte	Bit	Content	Content Description
0-1		<b>L</b>	Length of this structured field
2		<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3		<b>QCODE</b>	<b>X'81'</b> Identifies this Query Reply as Usable Area
4		<b>FLAGS</b>	
	0	RES	<b>B'0'</b> Reserved
	1	PP	Page printer  <b>B'0'</b> Non-page printer <b>B'1'</b> Page printer
	2	RES	<b>B'0'</b> Reserved(1)
	3	HC	Hard copy:  <b>B'0'</b> Not a hard copy device <b>B'1'</b> A hard copy device
	4-7	ADDR	Allowable addressing modes for alphanumeric support of the 3270 data stream  <b>X'0'</b> Reserved <b>X'1'</b> 12/14-bit addressing allowed <b>X'2'</b> Reserved <b>X'3'</b> 12/14/16- bit addressing allowed <b>X'F'</b> Unmapped (no explicit address)

5		<b>FLAGS</b>	
	0	VCP	Variable cells:  <b>B'0'</b> Variable cells not supported <b>B'1'</b> Variable cells supported
	1	CHAR	Characters:  <b>B'0'</b> Matrix character <b>B'1'</b> Non-matrix character
	2	CELLUNIT	Cell Units:  <b>B'0'</b> Value in bytes 6 and 7, 8 and 9; cells <b>B'1'</b> Value in bytes 6 and 7, 8 and 9; pels
	3-7	RES	<b>B'00000'</b> Reserved
6-7		<b>W</b>	Width of usable area in cells/pels
8-9		<b>H</b>	Height of usable area in cells/pels
10		<b>UNITS</b>	Units of measure of pels / in. or mm.  <b>X'00'</b> Inches <b>X'01'</b> Millimeters
11-14		<b>Xr</b>	Distance between points in X direction as a fraction, measured in UNITS, with 2-byte numerator and 2-byte denominator
15-18		<b>Yr</b>	Distance between points in Y direction as a fraction, measured in UNITS, with 2-byte numerator and 2-byte denominator
19		<b>AW</b>	Number of X units in default cell
20		<b>AH</b>	Number of Y units in default cell
21-22		<b>BUFFSZ</b>	Character buffer size (bytes)
23		<b>XMIN</b>	Minimum number of X units in variable cell (present if VCP flag set)
24		<b>YMIN</b>	Minimum number of Y units in variable cell (present if VCP flag set)
25		<b>XMAX</b>	Maximum number of X units in variable cell (present if VCP flag set)
26		<b>YMAX</b>	Maximum number of Y units in variable cell

(present if VCP flag set)

(1) A few early products set this bit to B'1' to indicate that the Outbound 3270DS and Erase/Reset structured fields were not supported.

Bytes 0 through 20 must always be included in the Usable Area Query Reply Base. Bytes 21 through 26 (parameters BUFFSZ and XMIN, YMIN, XMAX, YMAX) must be present if any self-defining parameters are included. If the parameters are not applicable, then zero values are used for the parameters.

---

### 6.51.2.2 Additional Content Description

The content of the Query Reply (Usable Area) structured field is further described as follows:

**PP** Indicates whether the device is a page printer. A *page printer* is a device that can print only one page at a time. This flag is applicable only if the device is a printer (that is, HC = B'1'). If HC is set to B'0', this flag must be set to B'0'.

**HC** Indicates that this device is a hard copy device (a printer).

**ADDR** Indicates the addressing modes that the device can support. Following are the valid values and a description of each value. All other values are reserved:

X'0' Reserved

This value was used by the 8100 to indicate 12-bit addressing. All displays and printers must support 12/14-bit addressing mode.

X'1' 12/14-bit addressing allowed

When the 12/14-bit addressing mode (outbound) is specified for a partition (Implicit Partition zero is always set to 12/14-bit addressing), bits 0 and 1 of the first address byte following the 3270 order are flag bits and have the following significance:

B'00' 14-bit binary address follows

The next 14 bits (the remainder of this byte and the full 8 bits of the next byte) contain the buffer address in binary form. No address translation is necessary.

B'01' 12-bit coded address follows

The next 14 bits are to be interpreted as a 3270 coded address (6 bits in each byte). The second byte has the first 2 bits as B'01' or B'11'. Otherwise, the data stream is rejected.

B'10' Reserved

Receipt of a buffer address beginning with the flag bits of B'10' cause the data stream to be rejected.

B'11' 12-bit coded address follows

The next 14 bits are to be interpreted as a 3270 coded address (6 bits in each byte). The second byte will have the first 2 bits as B'01' or B'11'. Otherwise, the data stream is rejected.

The rules for an inbound data stream with the 12/14-bit addressing mode are a function of the partition size and are as follows:

- If the partition size is greater than 4096 characters, then all of the addresses are in the 14-bit form.
- If the partition size is equal to or less than 4096 characters, then all of the addresses are in the 12-bit form.

X'2' Reserved

This value indicates 16-bit addressing only.

X'3' 12/14/16-bit addressing allowed. This value is used to support partitions that use the 16-bit addressing. The 16-bit addressing is interpreted as binary values.

For 12/14-bit addressing see X'1' above.

X'F' Unmapped (no explicit address). This value is used to indicate a non-3270 data stream addressing mode (for example, SCS for printers).

**VCP** Indicates that the device can support different character cell sizes in each partition. Thus, VCP must be B'0' unless the device also supports the Create Partition structured field. If VCP is set to B'1', a character cell size can be specified in the Create Partition structured field when each partition is created. The XMIN, YMIN, XMAX and YMAX parameters of this Query Reply provide the minimum and the maximum valid values for the character cell sizes.

**CHAR** Indicates whether matrix or non-matrix type characters are supported. (An example of a non-matrix type character would be found on a belt printer.) If this bit is set to B'1', bytes 10 through 20 are not applicable and must be set to zeros.

**CELLUNITS**

Indicates the unit of measure (character cells or pels) of the W and H parameters, as follows:

B'0' The values in W and H indicate Usable Area width and height in units of character cells.

B'1' The values in W and H indicate Usable Area width and height in units of pels.

#### **W and H**

Define the size of the usable area for the base display surface.

Thus, the size of the Usable Area in pels is (W x AW) by (H x AH) when Byte 5, bit 2 is B'0' (character cells), or W x H when Byte 5, bit 2 is B'1' (PELs).

Thus, W x H defines the maximum number of characters that can be presented in the Usable Area when characters from a nonloadable alphanumeric character set are written with the 3270 data stream.

For different types of printers, W and H have different meanings:

- For an LU Type 1 printer, W and H are the MPP and the MPL (respectively) supported by the implementation. These values indicate the maximum MPP and MPL that the host can set using the SHF(MPP) and the SVF(MPL) SCS controls. The values reported in the W and H fields are not affected by operator set values on printers that provide operator control (for example, switches). A host selection using SHF or SVF overrides an operator selection.
- For LU Type 3 and non-SNA 3270 printers, the W and H fields have no meaning and are ignored by the host. The W and H fields should be set to zeros.

**UNITS** Indicates the unit of measure (inches or millimeters) of the Xr and Yr parameters.

#### **Xr and Yr**

Give the distance between the pel centers in fractions of the units specified by the UNITS parameter (inches or millimeters). When determined, the numerator and denominator values for Xr and Yr are expressed in a 16-bit (2-byte), positive, hexadecimal number.

If a value for Xr or Yr cannot be determined, then Xr and Yr are set to X'FFFFFFFF' (Xr = X'FFFF' and Yr = X'FFFF'). If the device supports the Load PS Set structured field, then the values for Xr and Yr must not be X'FFFFFFFF'.

For example, if the device has 72.5 pels/inch horizontally, and 69 PELs/inch vertically, then the values returned would be as follows:

```
UNITS X'00'  
Xr   X'00020091' (2/145 inch)  
Yr   X'00010045' (1/69 inch).
```

#### **AW and AH**

Define the default character cell size for the device. If the

device supports the Load Programmed Symbol Set structured field and the size of the character cell used for either the default or alternate size of Implicit Partition Zero is different than (AW, AH), then the Character Cell Dimensions parameter must be returned in the Implicit Partitions Query Reply. If a partition is created without specifying the character cell dimensions (PW, PH), the character cell used by the display must be of size (AW, AH).

**BUFFSZ** Defines the amount of character buffer available. It applies only to a device that does not support partitions. If the device supports partitions, the Alphanumeric Partitions Query Reply defines the buffer size, and the BUFFSZ parameter is set to zero.

If there are no self-defining parameters, no variable character cell parameters, and the BUFFSZ parameter is not applicable, then both the BUFFSZ and the XMIN, YMIN, XMAX, YMAX can be omitted.

For different types of printers, BUFFSZ has different meanings:

- For LU Type 1, the BUFFSZ parameter is not applicable and should be set to X'0000'.
- For LU Type 3 and non-SNA 3270 Printers, the Implicit Partition Query Reply provides the buffer wrap points associated with the default and alternate buffer sizes. The BUFFSZ parameter is redundant. The value used in BUFFSZ is the same as that in the Implicit Partition Query Reply for Alternate Printer Buffer Size (APBS).

**XMIN, YMIN, XMAX, YMAX**

Define the limits of character cell size that can be specified in the Create Partition structured field when the device supports different size cells in each partition.

If there are no self-defining parameters and no variable character cell parameters, then the XMIN, YMIN, XMAX, YMAX parameters can be omitted. The parameters XMIN, YMIN, XMAX, YMAX are treated as one parameter (that is, if there is a value for XMIN, then there are values for YMIN, XMAX, YMAX).

---

### 6.51.3 On Pels Limit Self-Defining Parameter

The On Pels Limit self-defining parameter informs the host application of the percentage of pels that can be on at the same time.

If the device has no limit (100% of the pels on at the same time), then the device does not have to send this self-defining parameter.

The On Pels Limit self-defining parameter is required when the device has a limit for how many pels may be on at the same time.

The following table shows the format of the On Pels Limit self-defining parameter:

---

Byte	Content	Content Description
0	L	X'03' Length of this self-defining parameter
1	SDPID	X'01' Identifies this self-defining parameter as On Pels Limit
2	PERCENT	Percent of pels allowed on at a time

The content of the On Pels Limit self-defining parameter is further described as follows:

#### PERCENT

Indicates the percentage of the available pels, reported in the Usable Area Query Reply Base, that can be on at the same time. The PERCENT parameter gives the percentage as a hexadecimal value. Valid values are X'01' through X'64' (1 through 100 percent). For example, if the limit is 31%, the value of the parameter is X'1F'.

Exceeding the reported limit can result in the loss of displayed information.

---

## 6.51.4 Multiple Usable Area Self-Defining Parameter

The Multiple Usable Area self-defining parameter informs the host application of the presence of multiple (two or more), separate and independent display surfaces and what data types are to be viewed on each surface. If the device has only a single display surface, then the Multiple Usable Area self-defining parameter is not returned to the host application, and all of the data types supported by the device are displayed on the single display surface.

When the Multiple Usable Area self-defining parameter is included in the Usable Area Query Reply, the following apply:

- The existence of multiple display surfaces has no effect on the host data stream.
- The Usable Area Query Reply Base reports the usable area for alphanumeric (A/N) data and every data type not reported in a Multiple Usable Area self-defining parameter.
- Each instance of the Multiple Usable Area self-defining parameter must have only one unique data type (DATATYPE) parameter.
- The routing of data to the multiple display surfaces is done by data type. Therefore, each data type can only occur once per logical terminal.
- When a partition is created in the Multiple Usable Area environment, the device creates the viewports for each of the display surfaces. The viewport for the Alphanumeric partition is defined (size and position) in the Create Partition structured field. The size and position of the viewports for the other display surfaces is determined by the device using the data in the Create Partition structured field and the data in the Usable Area Query Reply.

The Multiple Usable Area self-defining parameter is always required when the device supports multiple, separate, and independent display surfaces.

**Note:** Printers cannot use the Multiple Usable Area self-defining parameter.

The following table shows the format of the Multiple Usable Area self-defining parameter:

Byte	Bit	Content	Content Description
0		<b>L</b>	<b>X'13'</b> Length of this self-defining parameter
1		<b>SDPID</b>	<b>X'03'</b> Identifies this self-defining parameter as Multiple Usable Area
2		<b>RES</b>	<b>X'00'</b> Reserved - must be set to zero
3		<b>FLAGS</b>	
	0-1	<b>RES</b>	<b>B'00'</b> Reserved
	2	<b>CELLUNIT</b>	Cell units
			<b>B'0'</b> Value in bytes 4-5, 6-7 in cells <b>B'1'</b> Value in bytes 4-5, 6-7 in pels
	3	<b>RES</b>	<b>B'0'</b> Reserved
	4-7	<b>DATATYPE</b>	Data types presented on Usable Area
			<b>B'0001'</b> Reserved (1) <b>B'0010'</b> Graphic data type <b>Others</b> Reserved
4-5		<b>WG</b>	Width of Multiple Usable Area
6-7		<b>HG</b>	Height of Multiple Usable Area
8		<b>UNITS</b>	Units of measure of pels/in or mm
			<b>X'00'</b> Inches <b>X'01'</b> Millimeters
9-12		<b>Xr</b>	Distance between points in X direction as a fraction measured in UNITS, with 2-byte numerator and 2-byte denominator
13-16		<b>Yr</b>	Distance between points in Y direction as a fraction measured in UNITS, with 2-byte numerator and 2-byte denominator
17		<b>AW</b>	Number of X units in default cell

18		<b>AH</b>	Number of Y units in default cell
(1) Previously, this value was used to indicate an alphanumeric data type.			

The content of the Multiple Usable Area self-defining parameter is further described as follows:

**CELLUNITS**

Indicates the unit of measure (character cells or pels) of the WG and HG parameters (see the Usable Area Query Reply Base CELLUNITS parameter).

**DATATYPE**

Indicates which type of data is presented on this multiple display surface.

**WG and HG**

Define the size of the usable area for this multiple display surface (see the Usable Area Query Reply Base W and H parameters).

**UNITS**

Indicates the unit of measure (inches or millimeters) of the Xr and Yr parameters.

**Xr and Yr**

Give the distance between the pel centers in fractions of the units specified by the UNITS parameter (inches or millimeters). The numerator and denominator values for Xr and Yr are expressed in a 16-bit (2-byte), positive, hexadecimal number.

If Xr and Yr cannot be determined, X'FFFFFFFF' is returned for both. Xr and Yr must not be X'FFFFFFFF' for a device that supports the Load Programmed Symbol Set structured field. (See the Usable Area Query Reply Base Xr and Yr parameters.)

**AW and AH**

Define the default character cell size for this multiple display surface. (See the Usable Area Query Reply Base AW and AH parameters.)

### 6.51.5 Alternate Usable Area Self-Defining Parameter

The Alternate Usable Area self-defining parameter is present only for devices that support more than one usable area size on one display surface.

For example, a physical device such as the 3180 has the capability to display up to 43 rows and up to 132 columns, but not concurrently. The device could report 43 rows and 80 columns (Model 4) in the Usable Area Query Reply Base and 27 rows and 132 columns (Model 5) in the Alternate Usable Area self-defining parameter.

The Alternate Usable Area self-defining parameter is limited to a single partition implementation. When the Alternate Usable Area self-defining

parameter is used and the device receives a Create Partition structured field, the following apply:

- If the defined viewport fits within the usable area defined in the Usable Area Query Reply Base, then that usable area is used.
- If the defined viewport does not fit in the usable area defined in the Usable Area Query Reply Base, then the usable area defined in the Alternate Usable Area self-defining parameter is tried. If the viewport fits, that usable area is used.
- If the viewport does not fit in either usable area, then the Create Partition structured field is rejected.

**Note:** The Alternate Usable Area self-defining parameter cannot be used for printers.

The Alternate Usable Area self-defining parameter is always required when the device supports more than one usable area size on one display surface.

For devices that support the Load PS Set structured field, bytes 8 through 18 must be supplied so that the host application can determine the correct character cell size.

The following table shows the format of the Alternate Usable Area self-defining parameter:

Byte	Content	Content Description
0	<b>L</b>	<b>X'13'</b> Length of this self-defining parameter
1	<b>SDPID</b>	<b>X'02'</b> Identifies this self-defining parameter as Alternate Usable Area
2	<b>RES</b>	<b>X'00'</b> Reserved - must be set to zero
3	<b>AUAID</b>	Alternate Usable Area ID
		<b>X'01'</b> Identifier for the Alternate Usable Area <b>X'FE'</b> Reserved values
4-5	<b>WAUAI</b>	Width of AUAi
6-7	<b>HAUAI</b>	Height of AUAi
8	<b>AUAUNITS</b>	Units of measure
		<b>X'00'</b> Inches <b>X'01'</b> Millimeters
9-12	<b>AUAXr</b>	Distance between points in X direction as a fraction, measured in AUAUNITS, with 2-byte numerator and 2-byte denominator.
13-16	<b>AUAYr</b>	Distance between points in Y direction as a

		fraction, measured in AUAUNITS, with 2-byte numerator and 2-byte denominator
17	<b>AWAUai</b>	Number of X units in default cell
18	<b>AHAUai</b>	Number of Y units in default cell

The content of the Alternate Usable Area self-defining parameter is further described as follows:

**WAUai and HAUai**

Define the size of the usable area for the alternate display surface (see the Usable Area Query Reply Base W and H parameters). The units of WAUai and HAUai are the CELLUNITS defined in the Usable Area Query Reply Base.

**AUAUNITS**

Indicates the unit of measure (inches or millimeters) of the AUAXr and AUAYr parameters.

**AUAXr and AUAYr**

Give the distance between the 'pel centers' in fractions of the units specified by the AUAUNITS parameter (inches or millimeters). The numerator and denominator values for Xr and Yr are expressed in a 16 bit (2 byte), positive, hexadecimal number.

If the device is unable to determine Xr and Yr, X'FFFFFFFF' is returned for both. Xr and Yr must not be X'FFFFFFFF' for a device that supports the Load Programmed Symbol Set structured field. (See the Usable Area Query Reply Base Xr and Yr parameters.)

**AWAUai and AHUai**

Define the default character cell size for the alternate display surface (see the Usable Area Query Reply Base AW and AH parameters).

## 6.52 Query Reply (3270 IPDS)

This section describes the Query Reply (3270 IPDS) structured field.

Subtopics:

- [6.52.1 Function](#)
- [6.52.2 Format](#)

### 6.52.1 Function

The Query Reply (3270 IPDS) structured field indicates support of the Intelligent Printer Data Stream (IPDS) through the 3270 data stream (non-SNA).

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE = X'9A', Equivalent or All).

---

## 6.52.2 Format

The following table shows the format of the Query Reply (3270 IPDS) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0008'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Identifies this structured field as a Query Reply
3	<b>QCODE</b>	<b>X'9A'</b> Identifies this Query Reply as 3270 IPDS
4-5	<b>FLAGS</b>	Reserved
6-7	<b>TRANLIM</b>	Maximum transmission size allowed outbound. It specifies the maximum number of bytes of IPDS data allowed in an outbound transmission. A value of X'0000' indicates no limit specified.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 7.0 Chapter 7. Magnetic Slot Reader, Keyboard, and Selector Pen Operations

Subtopics:

- [7.1 Introduction](#)
- [7.2 Magnetic Slot Reader Operations](#)
- [7.3 Keyboard Operations](#)
- [7.4 Selector Pen Operation](#)

---

### 7.1 Introduction

This chapter describes the magnetic slot reader (MSR), keyboard, and selector pen operations. The discussion of MSR also applies to the magnetic hand scanner (MHS).

---

### 7.2 Magnetic Slot Reader Operations

This section describes two modes of MSR operations are described: the 3275/3277-compatible mode and the numeric/alphanumeric mode.

Subtopics:

- [7.2.1 3275/3277-Compatible Operation](#)
- [7.2.2 Numeric/Alphanumeric Operation](#)
- [7.2.3 Test Card](#)

---

#### 7.2.1 3275/3277-Compatible Operation

Following is the numeric character set used for the 3275/3277-compatible mode operation:

---

Stripe Codes

Display-Generated Codes

---

Character	Hex	EBCDIC	ASCII	
0	0	F0	30	
1	1	F1	31	
2	2	F2	32	
3	3	F3	33	
4	4	F4	34	
5	5	F5	35	
6	6	F6	36	
7	7	F7	37	
8	8	F8	38	
9	9	F9	39	
	A	7A	3A	Operator ID (OID)
	B	7B	23	Start of record (SOR)
	C	7C	40	Unassigned
	D	7D	27	Field separator
	E	7E	3D	Unassigned
	F	7F	22	End of Record (EOR)

Magnetic stripes can contain up to 128 characters. These characters include the start of record (SOR), which does not appear in the input data stream. Also included are the EOR and the longitudinal redundancy check (LRC), which appear in the input data stream. The application program must verify that X'F7' appear just prior to the LRC.

The data portion of the stripe, the part between the initial start of record and the EOR, should contain only 0-9, D (field separator), or the OID (which is the A immediately following the SOR indicating secure data). The secure data is for the application program's use; the display treats all stripes as secure. Stripes containing A (other than special A), B, C, or E may be rejected. However, if accepted, the A, B, C, and E will be sent to the application program as 7A, 7B, 7C, and 7E, respectively. The graphics associated with A, B, C, and E are whatever the attaching device associates with the resulting EBCDIC or ASCII codepoint.

When the SOR character is read from the stripe, an attribute character is entered automatically into the cursor identified location of the buffer, provided the cursor is at an unprotected character location. This attribute character defines the stripe data field following it as protected, alphanumeric, nondisplay/nonprint. As the rest of the stripe data is read into the buffer, it is stored, starting at the first character location after this attribute character. As each character is stored in the buffer, the cursor advances one buffer location. The cursor may not be visible to the operator until the end of the operation. An MSR operation automatically causes the keyboard to lock and an MSR AID to become pending. At an appropriate time, the AID character and the display

buffer data are sent to the application program during a read-modified operation.

No additional keyboard data can follow an MSR operation. There are some differences in the data-stream content, depending on whether the device buffer is formatted or unformatted:

- Unformatted display. When an MSR input operation occurs, the device buffer becomes formatted because of the insertion of the protected, nondisplay attribute with the MDT bit set on. Any data that was previously displayed will no longer be displayed. (See [Figure 7-1](#).)
- Formatted display. An MSR operation may be initiated only when the cursor is located in an unprotected field. The input data stream resulting from an MSR operation will contain at least two fields with the MDT bit set on (new data field and the previous data field) because all the information from the stripe is treated as data until the information has been entered. Also, the MDT bit is set on in the attribute character created by the SOR. [Figure 7-2](#) and [Figure 7-3](#) show how to prepare an unprotected field for MSR input. In [Figure 7-2](#), the cursor follows the operator instructions. In [Figure 7-3](#), the instructions are in a protected field and the cursor follows an unprotected field attribute character.

When the extended field and character attributes are supported, an MSR operation causes an extended field attribute of X'00' to be generated in addition to the field attribute. Data characters generates character attributes of X'00'.

 PICTURE 14

Figure 7-1. Display Screen Activity Before and After Magnetic-Stripe Input (Unformatted Display)

 PICTURE 15

Figure 7-2. Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Unprotected Field Attribute)

 PICTURE 16

Figure 7-3. Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Protected Field Attribute)

---

## 7.2.2 Numeric/Alphanumeric Operation

For magnetic stripes using the numeric/alphanumeric character sets, the two character positions following the start sentinel (SS) perform a header function that designates to the display the data field contents, as follows:

```
*,* = numeric-character set, nonsecure data
'A',* = numeric-character set, secure data
'C','A' = numeric/alphanumeric-character set, nonsecure data
'A','C' = numeric/alphanumeric-character set, secure data.
```

**Note:** \* = 0 through 9 or D.

Any magnetic stripes not conforming to the above header information are rejected; the information is not sent to the application. Also, if a display does not support a particular character set, cards with magnetic stripes using that character set are rejected.

For the alphanumeric character set magnetic stripes, the data field starts after the two-character header. For the secure numeric character set magnetic stripes, the data field starts after the A character. For nonsecure numeric character set magnetic stripes, the data field starts after the SS B. For the numeric/alphanumeric character set magnetic stripes, the data field ends with (but does not include) the EOC ( F). Only the data field is sent to the application program. The LRC check is done by the display. The LRC and the end sentinel (ES) are not sent to the application program.

The data field, up to 125 characters less the header, is sent to the application program in conformance with the magnetic stripe character set designated by the header (that is, the numeric or alphanumeric character set).

Subtopics:

- 7.2.2.1 Stripe Codes and Application Program Codes
- 7.2.2.2 Secure/Nonsecure Magnetic Stripe Cards

### 7.2.2.1 Stripe Codes and Application Program Codes

The following table shows the relationship of the magnetic stripe code to the codepoint that the display generates. This relationship is independent of the graphic character set supported by the attaching display. For example, the same codepoint is generated for the U.S. EBCDIC character set displays as would be generated for a German character-set display. The graphic displayed when display is allowed depends on the graphic character sets supported by the attaching display.

Following is the magnetic-stripe numeric character set:

Stripe Codes		Display-Generated Codes	
Character	Hex	EBCDIC	ASCII
0	0	F0	30

1	1	F1	31
2	2	F2	32
3	3	F3	33
4	4	F4	34
5	5	F5	35
6	6	F6	36
7	7	F7	37
8	8	F8	38
9	9	F9	39

Control Data Characters (Hex)	
A (secure data)	Not sent in the data stream
B (SS)	Not sent in the data stream
C (reserved)	Not sent in the data stream
D (space)	40
E (reserved)	Not sent in the data stream
F (ES)	Not sent in the data stream

The control characters *A* and *B* must appear in their respective positions on the magnetic stripe, or they will be treated as errors. The control characters *C* and *E* will always be treated as an error. The *A* means secure data (protected, nondisplay, and nonprint) when located immediately after the SS. The *B* is the SS. The *F* is the ES.

Another SS, called the *reverse start sentinel (RSS)*, follows the LRC on the magnetic stripe. The RSS is for magnetic stripe card readers that can scan the card in both directions. The *B* is also used for the RSS. This type of operation is primarily for MHS operations.

Following is the alphanumeric character set:

Stripe Codes	Display-Generated Codes
Character	EBCDIC   ASCII

0A(1)	F0	30
1A(1)	F1	31
2A(1)	F2	32
3A(1)	F3	33
4A(1)	F4	34
5A(1)	F5	35
6A(1)	F6	36
7A(1)	F7	37
8A(1)	F8	38
9A(1)	F9	39
C1	C1	41
C2	C2	42
C3	C3	43
C4	C4	44
C5	C5	45
C6	C6	46
C7	C7	47
C8	C8	48
C9	C9	49
D1	D1	4A
D2	D2	4B
D3	D3	4C
D4	D4	4D
D5	D5	4E
D6	D6	4F
D7	D7	50
D8	D8	51
D9	D9	52
E2	E2	53
E3	E3	54
E4	E4	55
E5	E5	56
E6	E6	57
E7	E7	58

E8	E8	59
(1)Indicates that A is a filler character when numerics are not paired.		

Stripe Codes	Display-Generated Codes	
	EBCDIC	ASCII
E9	E9	5A
0C	4A	5B
1C	5A	5D
3C	7A	3A
4C	4C	3C
5C	5C	2A
6C	6C	25
7C	7C	40
0D	4B	2E
1D	5B	24
2D	6B	2C
3D	7B	23
4D	4D	2B
5D	5D	29
6D	6D	5F
7D	7D	27
0E	4F	21
1E	5F	5E
2E	6F	3F
3E	7F	22
4E	4E	2B
5E	5E	3B
6E	6E	3B
7E	7E	3D
E0	E0	5C
E1	61	2F

DA	50	26
EA	60	2D
CA	40	20
00	F0,F0	30,30
01	F0,F1	30,31
.	.	.
.	.	.
.	.	.
89	F8,F9	38,34
99	F9,F9	39,39
(1)Indicates that A is a filler character when numerics are not paired.		

When using the alphanumeric character set, the numerics are coded as two 4-bit numeric characters for each 8-bit byte. As a result, there must be either an even number of numerics in any continuous string of numerics or an odd number of numerics with a filler character. The code sequence (12XYZ) is represented on a card as (12E7E8E9). The code sequence (123XYZ) is represented on a card as (123AB7E8E9). This limits the number of characters on a card to 62 alphanumeric and special characters and to 124 numeric characters. The hexadecimal A (1010) is used as the filler character.

The initial B is SS. The F is the ES. The RSS follows the LRC on the magnetic stripe card. The RSS is used by devices using a magnetic stripe card reader that can scan the card in either direction.

Any stripe code not shown (except for CAEE--the test card) causes a rejection of the card when that code appears between the header and the ES. The first F encountered after the SS is treated as the ES. The code following the ES is considered the LRC.

---

### 7.2.2.2 Secure/Nonsecure Magnetic Stripe Cards

With regard to the application program, display operations that use the numeric/alphanumeric character set MSRs closely parallel the 3275/3277-compatible operation. However, secure and nonsecure cards are handled differently.

*Secure Magnetic Stripe Cards:* Handling of secure numeric/alphanumeric magnetic stripe cards is similar to the handling of cards during the 3275/3277-compatible operation. The same rules apply to the cursor when it is in an unprotected field, to the generation of a protected, nonprint/nondisplay field attribute, and to automatic entry. An AID code (E7 for EBCDIC, 58 for ASCII) is generated that indicates that the MSR is

extended. This code informs the application program that the magnetic stripe data conforms to the numeric/alphanumeric character set operation. Nothing indicates whether the data field is in the numeric or alphanumeric character set. No ES or LRC is sent with the data. When the application receives the E7,(58), AID, the data is valid; the LRC check has been completed.

*Nonsecure Magnetic Stripe Cards:* Nonsecure numeric/alphanumeric character set magnetic stripe cards appear to the application program as data entered using the keyboard. An exception is that the magnetic stripe data does not follow the keyboard rules relative to numeric field/numeric lock/keyboard type. Nonsecure magnetic stripe data can be entered in any unprotected field. The cursor must still be in an unprotected field for a successful MSR operation. However, no protected, nonprint/nondisplay field attribute is generated. No AID is generated, and no automatic entry takes place. The sending of the data relies on normal entry operation--for example, the Enter key or selector pen attention. An implementation may optionally provide for a customer selected override that causes auto entry of nonsecure data. However, the AID generated is coded as the Enter key.

---

### 7.2.3 Test Card

For displays, any nonsecure card of known content can be used as a test card. The alphanumeric test card (CAEE--) is treated as a normal nonsecure card, except that the EE is deleted. There is no automatic verification of the test data. The operator performs visual verification. All MSR data is inhibited from being sent to the application program when the display is in subsystem test mode.

---

## 7.3 Keyboard Operations

The keyboard enables the operator to change, edit, or create data except within protected fields. As data is being composed or modified by keyboard operations, the changes are inserted in the character buffer and displayed on the screen.

Subtopics:

- 7.3.1 Keys that Affect the Data Stream
- 7.3.2 Clear Partition
- 7.3.3 Clear
- 7.3.4 Error Override
- 7.3.5 Keyboard Actions with Attribute Selection Keys
- 7.3.6 Keyboard Actions in Partitions
- 7.3.7 Scrolling Partitions

---

### 7.3.1 Keys that Affect the Data Stream

The data stream is oriented toward a keyboard display, but during implementation, the user has the option of providing or not providing a keyboard. If supported, the key functions and key positions depend on the type of keyboard and the keyboard definition that the user provides. The keyboard description that follows is limited to the functions that have an

effect on the data stream.

Subtopics:

- 7.3.1.1 Cursor
- 7.3.1.2 Alphanumeric Data
- 7.3.1.3 Automatic Skip
- 7.3.1.4 Erase to End of Field (Erase EOF) Key
- 7.3.1.5 Erase Input Key
- 7.3.1.6 Insert Mode Key
- 7.3.1.7 Delete Key
- 7.3.1.8 Duplicate (DUP) Key
- 7.3.1.9 Field Mark Key
- 7.3.1.10 Program Attention Keys

---

### 7.3.1.1 Cursor

The cursor indicates where the next character entered from the keyboard will be displayed on the display surface (viewport). For example, assume the cursor character is an underscore. When the cursor is displayed under one character in a row of characters, that character can be changed or deleted by keyboard action. Also, if the cursor is displayed under a position without a character, a character can be placed in that position by keyboard action. All these operations, when performed on a field of a formatted display, cause the MDT bit (bit 7) of the field attribute for the field to be set to 1. However, when the cursor appears beneath a character in a protected field or beneath a field attribute, that position cannot be modified by keyboard action, and the MDT bit is not set.

When the display is turned on, the cursor is usually in the first location on the screen. It may be moved by an application. The cursor can be repositioned by the keyboard operator and also by the application program. The cursor is not affected by field attributes; it is displayed even when positioned in a nondisplay/nonprint field.

---

### 7.3.1.2 Alphanumeric Data

Alphabetic characters can be entered into the display buffer in the code for either uppercase or lowercase, depending on the position of the Shift key. Only uppercase alphabetic codes can be entered from some keyboards.

Keyboard entry of an alphanumeric character into the buffer occurs at the cursor location, provided that the cursor is located in a character location within an unprotected field. (An attempt to enter an alphanumeric character into a protected field or into a field attribute location is blocked.) Successful keyboard entry of the alphanumeric character causes the cursor to advance to the next character location within the unprotected field.

---

### 7.3.1.3 Automatic Skip

Upon entry of a character into the last character location of an unprotected field, the cursor is repositioned according to the field attribute describing the next field.

If the field attribute defines the next field as alphanumeric or numeric and unprotected, the field attribute is skipped and the cursor is positioned to the first character location in that field.

If the field attribute defines the field as numeric and protected, the cursor skips that field and is positioned to the first character location of the next unprotected field.

---

### 7.3.1.4 Erase to End of Field (Erase EOF) Key

If the cursor is located in an alphanumeric character location in an unprotected field, this key clears the character location occupied by the cursor and all remaining character locations in that field to nulls. Each character attribute associated with the nulled characters is set to its default value.

If the display is unformatted (that is, there are no field attributes on the display), the character buffer from the cursor address to the end of the screen is set to nulls, and the corresponding character attributes are set to their default values.

The operation can wrap from the end of the last row of the display to the beginning of the top row. The cursor does not move as a result of operating this key, and the MDT bit is set to 1.

When the cursor is in a field attribute location or is within a protected field, pressing this key locks the keyboard and notifies the terminal operator of an input inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

---

### 7.3.1.5 Erase Input Key

This key clears all unprotected character locations to nulls, resets the MDT bit to 0 in unprotected fields, sets all character attributes to their default value, and repositions the cursor to the first unprotected character location on the display.

In a buffer with only protected fields, no character locations are cleared and the cursor is repositioned to buffer address 0.

If the display contains no fields, the entire character buffer is cleared to nulls, the cursor is repositioned to character location 0, and all character attributes are set to their default value.

---

### 7.3.1.6 Insert Mode Key

This key turns on the insert mode indicator on the display (if one exists) and places the display in an insert mode of operation, regardless of the cursor location.

If the cursor is located in an unprotected field that has a null character either in the character location identified by the cursor or in any character location in the field beyond the cursor, pressing an

alphanumeric key causes that alphanumeric character to be entered at the cursor position and the MDT bit to be set to 1. The character formerly occupying the cursor location and all remaining characters within the field (except for null characters or characters to the right of null characters) are shifted one character location to the right. If more than one row of characters is contained within the field, a character occupying the last character location in the row is shifted into the first character location of the next row. If the location identified by the cursor at the time of the insert operation is a null, no character shifting occurs.

The character attributes of any shifted characters do not change. Each character attribute associated with the inserted character is set to its default value. In all cases, the PS, color, and highlighting select keys (if available on the keyboard, and allowed by the Set Reply Mode structured field) set the attribute values associated with the inserted character as it is stored.

After all null characters at or beyond the cursor location in the field have been overwritten, or if there were no null characters, pressing an alphanumeric key locks the keyboard and notifies the terminal operator of an input inhibit condition. The field attribute and the extended field attribute remain unchanged.

Pressing an alphanumeric key while in insert mode when the cursor is located in a field attribute location or is within a protected data field, locks the keyboard and notifies the operator of an input inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

Operation of the Reset key returns the keyboard from insert mode to normal mode.

---

### 7.3.1.7 Delete Key

If the cursor is located in an alphanumeric character location in an unprotected field, pressing the Delete key deletes the character from the character location occupied by the cursor and sets the MDT bit to 1. The cursor does not move. All remaining characters in the unprotected field (to the right of the cursor and on the same row) are shifted one character location to the left. The vacated character location at the end of the row is filled with a null and the default character attributes. If the unprotected field encompasses more than one row, characters in rows other than the row identified by the cursor may or may not be affected.

An implementation may choose to limit the delete to the current row or may choose to allow the delete local function to affect the whole field which may span many rows.

The character attributes of any shifted characters do not change.

Operation of this key when the cursor is located in a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

---

### 7.3.1.8 Duplicate (DUP) Key

Operation of this key causes a X'1C' code to be entered into the presentation space, a Tab key operation to be performed, and the MDT bit to be set to 1. The operator uses the Duplicate key to tell the application program that a duplicate operation is indicated for the rest of the field in which it is located. The character transferred to the application program is a X'1C' (EBCDIC) and is sent when the data is read. No duplicate operation is performed at the display. The duplicate character, when stored in the character buffer, is displayed as asterisk overscore. Display devices operating in monospace mode display this character as an asterisk.

Operation of this key when the cursor is positioned at a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

---

### 7.3.1.9 Field Mark Key

Operation of this key causes a X'1E' code to be entered into the active partition buffer and the MDT bit to be set to 1. The field mark character informs the application program of the end of a field in an unformatted buffer or subfield in a formatted buffer. The field mark character transferred to the application program is X'1E' (EBCDIC) and is sent when the data is read. The field mark character, when stored in the character buffer, is displayed as semicolon-overscore. Display devices operating in monospace mode may also display the field mark character as a semicolon.

Operation of this key when the cursor is positioned at a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

---

### 7.3.1.10 Program Attention Keys

These keys solicit application program action by causing the sending of an AID character to identify the key. The MDT bit is not affected. The program attention keys are Clear, Enter, Cancel, all program function (PF) keys, and the program access (PA) keys.

**Note:** Not all program attention keys are available on all types of keyboards.

---

## 7.3.2 Clear Partition

The effect of invoking this function depends on whether the display is in implicit partition state or in explicit partition state.

The operation of the Clear Partition function does the following:

- Sets all buffer locations for the active partition to nulls and all character attributes to X'00'.
- Sets the reply mode for the active partition to the default (field reply mode).

- Causes an AID (of value X'6A') to be transmitted inbound in a Short Read operation.
- Moves the current cursor position to the top left corner of the presentation space (buffer address 0). In a scrollable partition, this can cause automatic scrolling to occur.

Invoking this function while a previous inbound transmission is being processed locks the input device, and signal an input-inhibited condition to the operator on the indicator row of the display.

---

### 7.3.3 Clear

If the display is in implicit partition state, invoking the Clear function results in the following:

- All buffer locations are set to nulls.
- All character attributes are set to their default values.
- The MDT bit is set to 0.
- The reply mode is set to the default (field reply mode).
- An AID of X'60' is transmitted inbound in a short read operation.
- The implicit partition is either set to its default size or remains the same depending on the customization value selected or the option that has been designed into the product.
- The cursor is moved to the current cursor position, buffer address 0.

If the device is in explicit partitioned state, invoking the Clear function results in the following:

- All existing partitions are destroyed.
- A partition with a PID of 0 and default size is created.
- The device is reset to implicit partition state.
- The cursor is moved to the current position, buffer address 0.
- All buffer locations are set to nulls.
- All character attributes are set to their default values.
- The MDT bit is set to 0.
- The reply mode is set to the default (field reply mode).
- An AID of X'6D' is transmitted inbound in a short read operation.

---

### 7.3.4 Error Override

Operation of this key when the cursor is in an unprotected field causes a special substitute (SUB) character with a value X'3F' to be entered into the active partition buffer at the current cursor position. The SUB character provides a means of informing the application program that there

is an error in the field.

The cursor is advanced to the next position and the MDT bit is set to 1. The character is displayed as a solid circle.

Invoking this function when the cursor is in a protected field locks the keyboard and notifies the terminal operator of an input-inhibit condition.

With the cursor in an unprotected field of an unprotected partition, or in an unprotected partition that is unformatted, invoking this function causes the following actions:

- A special substitute (SUB) character value X'3F' is entered into the active partition buffer at the current cursor position. The SUB character provides a means of informing the host that there is an error in the field.
- The MDT bit for the field containing the cursor is set to 1.
- The cursor position is advanced by one position as for normal data entry.
- The character displays as a solid circle.
- The character attribute is updated according to the highlighting and color currently specified.

Invoking this function when the cursor is in a protected partition at an attribute location or in a protected field raises the Input-Inhibited-Wrong-Place condition.

Invoking this function when the Field Validation display function is not in use raises the Input-Inhibited-Minus Function condition.

Invoking this function while the input device is owned by the SSCP-LU session locks the input device and signal an input-inhibit condition to the operator on the indicator row of the display.

---

### 7.3.5 Keyboard Actions with Attribute Selection Keys

For those keyboards that do not have attribute selection keys for character sets, color, and highlighting, pressing an alphanumeric key causes the MDT bit to be set to 1 and the alphanumeric character to be entered into the display's character buffer at the position indicated by the cursor. The keyed character is assigned default values for programmed symbols, color, and highlighting.

For keyboards with keys that allow the operator to select or change the character attribute without application program interaction, the default settings are obtained as stated in the previous paragraph. When enabled by the application program, keying sequences are defined that allow the operator to select character sets, color, and highlighting attribute values.

Once these attribute values have been selected, pressing an alphanumeric key causes the alphanumeric character to be entered into the character buffer at the cursor position and the MDT bit to be set to 1. In addition, the character attribute associated with the entered character is modified by the attribute values selected by the operator (if operator alteration of the character attribute have previously been allowed by the Set Reply Mode structured field). The operator's selections apply until changed or until the keyboard is returned to the default setting.

---

### 7.3.6 Keyboard Actions in Partitions

As the operator enters data, the current cursor position is incremented by 1 for each character entered, and the cursor is displayed in the new character location. This cursor movement may cause automatic scrolling.

The operator may also move the screen cursor by using the Up, Down, Left, Right, Backspace, Tab, Backtab, Skip, or New Line keys. The Up, Down, Left, Right, and Backspace keys move the screen cursor within the viewport of the active partition and cause the current cursor position to be updated. Thus, the screen cursor wraps at the viewport boundaries.

The entering of data and the Tab, Backtab, Skip, and New Line keys move the current cursor position within the character buffer of the active partition. Where the buffer is larger than the viewport, automatic scrolling may occur.

If a character buffer contains at least one unprotected field, the first unprotected character location in the buffer is termed the *home position*. If the buffer contains no unprotected fields, the home position is defined to be character location zero.

The Home key resets the current cursor position in the active partition to the home position and causes the cursor to move to this position within the viewport. This may cause automatic scrolling to occur.

The effect of the Delete, Erase EOF, and Erase Input keys is constrained to the active partition; that is, no deletions or storing of nulls occurs outside the active partition. The Clear Partition key clears the active partition. The Clear key clears the entire screen by destroying all partitions, placing the device in implicit partition state, and recreating an implicit partition 0 of default size.

---

### 7.3.7 Scrolling Partitions

Different areas of the presentation space can be seen in the viewport by a technique called *scrolling*. Scrolling within a partition is possible if the presentation space is larger than the associated viewport. Scrolling can be done by the operator using the keyboard keys or by the host using a structured field.

When a partition is created, it can be set up as a scrollable partition by making the presentation space larger than the viewport. The viewport seen by the operator has a one-to-one relationship with a window on the corresponding presentation space. The position of the window on the presentation space determines what data is seen in the viewport. The initial position of the window on the presentation space is specified by the Create Partition structured field. The position of the window on the presentation space is reset using the Set Window Origin structured field.

The number of rows moved during scrolling is determined by the RS Value given in the Create Partition structured field.

Row-by-row vertical scrolling within the presentation space is provided in response to the operator's pressing the keys for the scroll up and scroll down. Vertical scrolling is achieved by moving the position of the window within the presentation space in response to the use of the keys for scrolling.

The operator interacts with the system through information displayed in the viewport. The cursor is always displayed within the viewport. Whenever an operator keystroke (of data or the field-oriented keys) causes the cursor to leave the viewport, there is an automatic scroll. Conversely, when a scrolling operation moves the window so that the cursor would no longer be within the viewport, the cursor is dragged along at the edge of the window and hence remains within the viewport.

The cursor movement keys (Up, Down, Left, Right, Backspace) wrap at the boundary of the viewport. The actions for data keystroking and the field-oriented keys are not affected by scrolling, that is, they wrap at the boundary of the character buffer. The wrap at the buffer boundary may cause an automatic scroll.

Subtopics:

- [7.3.7.1 Vertical Scrolling](#)
- [7.3.7.2 Keyboard Actions and Scrolling](#)
- [7.3.7.3 Action for Data Entry Keystrokes](#)
- [7.3.7.4 Automatic Scrolling](#)

---

### **7.3.7.1 Vertical Scrolling**

Assuming that the window is not already positioned at the bottom of the presentation space, the action in response to scroll up is to move the window down the presentation space. The effect, as seen by the operator, is to move the data up the viewport.

The previous top row is lost from the window; all other rows are moved up.

If the current cursor position is in the top row of the window, the cursor is moved down the presentation space so that it remains in the top row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

If the window is already positioned at the bottom of the presentation space, then scrolling up has no effect. There is no input-inhibit condition and no indicator.

Assuming that the window is not already positioned at the top of the presentation space, the action in response to scroll down is to move the window up the presentation space. The effect, as seen by the operator, is to move the data down the viewport.

The previous bottom row is lost from the window; all other rows are moved down.

If the current cursor position is in the bottom row of the window, the cursor is moved up the presentation space so that it remains in the bottom row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

If the window is already positioned at the top of the presentation space, then scrolling down has no effect. There is no input-inhibit condition and no indicator.

---

### **7.3.7.2 Keyboard Actions and Scrolling**

The character oriented keys (Up, Down, Left, Right, and Backspace) wrap the cursor at the viewport boundary. The field oriented keys (Tab, Backtab, Skip, and New Line) and automatic skip operate on the entire presentation space and wrap at its boundary. If the resulting current cursor position is outside the window, there is an automatic scroll.

The Home key positions the cursor at the home position within the presentation space. This key may cause an automatic scroll.

---

### **7.3.7.3 Action for Data Entry Keystrokes**

All data entry keystrokes cause normal incrementing of the cursor position, including a wrap at the boundary of the presentation space. If the resulting cursor movement places the cursor outside the window, there is an automatic scroll.

The Erase EOF, Erase Input, Clear Partition, Clear, and Delete keys, and the use of the Insert Mode key, can cause changes to the character positions anywhere in the presentation space; in particular, such changes can be outside the viewport.

---

### **7.3.7.4 Automatic Scrolling**

An automatic scroll occurs whenever a field oriented key, a data entry keystroke, the Home key, or an IC order results in a current cursor position that is outside the window.

The window is moved by the minimum number of lines necessary to put the current cursor position in a peripheral row of the window. The effect is that the cursor always remains within the window, and hence within the viewport.

---

## **7.4 Selector Pen Operation**

The selector pen is a light-sensitive pen-like device that can detect the light emitted from characters displayed on the display surface. With the selector pen, the operator can select from a list or table of displayed items and can then cause those selections to be identified to the application program.

Subtopics:

- 7.4.1 Selector Pen Field Format
- 7.4.2 Designator Characters
- 7.4.3 Selecting Fields in Partitions

---

## 7.4.1 Selector Pen Field Format

A field that is to be used for selector pen operations must be defined in the following format:

Field Attribute	The field attribute defines the field as displayed and selector pen detectable. (A field may be protected or unprotected, alphanumeric or numeric.)
Designator Character	The designator character defines the type of operation that will be performed by detection on this field.
Displayed Data	One or more displayed alphanumeric characters for sensing by the selector pen.

---

## 7.4.2 Designator Characters

Designator characters are used to define two types of selector pen fields: selection fields and attention fields. Each type of field performs a different selector pen operation. If the field attribute defines a field as selector pen detectable but the character in the designator character position is not a valid designator, a detect cannot be made on the field.

Subtopics:

- 7.4.2.1 Selection
- 7.4.2.2 Attention

---

### 7.4.2.1 Selection

The selection field is defined by a question mark (?) designator character (X'6F'). When the selector pen detects on a selection field, the MDT bit in the field attribute for the field is set to 1, and the X'6F' is changed to X'6E'. Also, the designator character is automatically changed in the character buffer and on the display surface to show to the operator that the detection was successful.

<b>If Before Detect:</b>	<b>Then After</b>
	<b>Detect:</b>
<b>Designator MDT</b>	<b>DesignatoMDT</b>

?	0	>	1
?	1	>	1

The field may be deselected by using the same technique as for selection.

If the designator is > (X'6E') and a detect is made on the field, the designator is changed to ? (X'6F') and the MDT bit is set to 0, and the X'6E' is changed back to X'6F'.

If Before Detect:		Then After	
Designator	MDT	Designator	MDT
>	0	?	0
>	1	?	0

---

### 7.4.2.2 Attention

The attention field is normally defined by a space or null designator character. In addition, an implementation can support the ampersand (&) designator character to provide an Enter key simulation.

A detection on an attention field designated by a space or null causes an AID X'7E' to be sent to the application program that identifies the source of entry to be the selector pen. Also sent are the addresses of fields with the MDT bit set to 1 and the current cursor position for the partition that contains the selected field. The application program responds, usually by issuing a Read Modified All, to obtain the contents of modified fields and the addresses.

A detection on an attention field designated by an ampersand results in an AID X'7D' being sent that identifies the source of entry to be the Enter key. Also sent are the addresses and contents of all fields with the MDT set to 1 and the current cursor position for the partition that contains the selected field. If the application defines an attention field with an ampersand designator and the implementation does not support the ampersand, the selector pen is inoperative on the field; the operator must then use the Enter key.

**Note:** The application programmer should be aware that high-intensity/unprotected fields can be modified by the operator to become selector pen detectable fields.

---

### 7.4.3 Selecting Fields in Partitions

The operator can use the selector pen to select fields in any partition, active or inactive. Selection of a field in an inactive partition causes no movement of the cursor on the display surface and the partition containing the selected field does not become active, but the operator may notice a change in the indicators. The operation performed depends on the designator character selected.

A selection outside the viewports is ignored.

Selection of an immediate detect field (designator of blank, null, or ampersand) causes data to be transmitted. This data includes the

addresses of all the modified fields and the current cursor position for the partition that contains the selected field. This need not be the active partition.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 8.0 Chapter 8. Printer Operations

Subtopics:

- 8.1 Introduction
- 8.2 Local Copy Function in an SNA Environment
- 8.3 Copy Initiation
- 8.4 Printer Availability
- 8.5 Display/Printer Compatibility
- 8.6 Partition Mode Considerations
- 8.7 Local Copy Command in the BSC Environment
- 8.8 Valid LU 1 SCS Control Codes
- 8.9 Early Print Complete (EPC) Operations
- 8.10 IPDS Data/Non-SNA

---

### 8.1 Introduction

Although oriented toward a display with a keyboard, the 3270 data stream descriptions apply, with certain exceptions, to a printer without a keyboard. The exceptions are the following:

**Commands** The read commands are not valid in the 3270 data stream received by the printer in an SNA environment. If received, the request is rejected (sense code X'1003'). See also the *Structured Field* exception below. However, the read commands are valid if received in the 3270 data stream in a non-SNA environment for locally attached 3274D models (or 3272 version) or BSC attached devices.

**Structured fields** Structured fields that are not supported and structured fields with invalid values are rejected, as follows:

- The Create Partition, Activate Partition, and Set Window Origin structured fields are not supported.
- The Read Partition structured field is limited to query operations in an SNA environment. The Read Modified command is supported only for sending Query Replies to an application.

**Selector pen** Not applicable to printer.

## WCC byte

The WCC byte for printer use is defined as follows:

### Bit Explanation

- 0, 1 In an SNA/EBCDIC environment, these bits are ignored by the printer. In an SNA/ASCII environment, these bits are set in accordance with Figure D-1 in Appendix D to make the WCC byte an EBCDIC/ASCII-translatable graphic.
- 2, 3 Defines printout format as follows:
- 00 The NL or CR orders in data stream determine print line length, and EM designates the end of the message. Provides a 132-character print line when orders are not present.
  - 01 Specifies a 40-character print line
  - 10 Specifies a 64-character print line
  - 11 Specifies a 80-character print line.
- 4 Start-printer bit. When set to 1, initiates a printout operation at the completion of the write operation.
- 5 Sound-alarm bit. When set to 1, sounds the audible alarm if an audible alarm is provided.
- 6 Keyboard restore bit.
- 7 Reset MDT bit.

**Format control orders** Unlike displays, the NL, CR, EM, and FF orders provide a print format function when received by a printer, as follows:

- NL (New Line) Moves the print position horizontally to the left margin and vertically down to the next line.
- CR (Carriage Return) Moves the print position horizontally to the left margin.
- EM (End of Medium) Terminates the print operation.
- FF (Form Feed) Moves the print position to the top and left margin of the next page.

In normal operation, when bits 2 and 3 of the WCC are 0, a new line function is performed each time a valid NL character is encountered. In addition, if no valid NL is encountered before the printer reaches the end of a line (as determined by the maximum physical carriage length), the printer automatically performs an NL and continues printing.

During a print operation, if line length format is specified in bits 2 and 3 of the WCC (bits 2 and 3 not equal to 0), data characters in the printer buffer are scanned one line at a time before they are printed. A line feed is executed after each line is printed. If a line contains only null characters and one or more space characters, a line feed is performed to cause a blank line in the printout. When null characters, field attributes, or alphanumeric characters in a nonprint field are encountered, they are treated as follows:

- If they are in a line that contains another print field, they are printed as spaces.
- If they constitute an entire line, they are ignored, and the line feed is not performed. Therefore, a blank line does not appear in the printout, and the data is compressed vertically one line.

When line length format is not specified, printout of the buffer begins at buffer address 0 and continues until the last position of the character buffer is printed or until a valid EM character is encountered. Each print line is left-justified. At the end of each printout, a final NL is executed so that the printer is ready to start the next printout. When the print terminating EM order appears in the first print position of the print line, a final NL is not executed, because the printer is already positioned at the left margin for the next printout.

The validity requirements for NL, CR, EM, and FF are as follows:

- NL, CR, and EM are valid only when encountered in a print field during a printout that does not have a line length format specified by the WCC.
- FF is valid in any buffer position.

When the printer supports Vertical Forms Control (VFC), an FF causes the form to index to a predetermined line, and the first print position (the buffer location containing the FF character) is printed as a space character. If VFC is not supported, the FF is invalid.

The rules for NL apply to CR. However, the printer must support CR at least to the extent of accepting and printing it as a space. Invalid NL, CR, EM, and FF are not executed and print as spaces.

**Data integrity** Since printers in an SNA environment do not support a read operation, the data integrity requirement on returned data is not applicable. Where the requirement says "must be returned as sent," for printers it becomes "must be accepted."

Because printers in a non-SNA environment can be read by the application program, these printers are subject to the same data integrity

requirements as displays.

---

## 8.2 Local Copy Function in an SNA Environment

A hard copy of the display screen or a portion of a display screen can be obtained by host application use of the conventional display read function and the printer write function. However, to simplify the host application and reduce line traffic, a specific copy function is provided that accomplishes a transfer of data to a printer without routing through the host application.

Whether initiated by the host or by the operator, the resulting hard copy is a replica of the screen or portion of the screen, with the following exceptions:

- Display lines containing all nulls are suppressed.
- Mismatch of uppercase and lowercase may occur. When the configured printer has the capability for selection of monospace/dualcase without requiring operator action, the printed data must match the case of the displayed data. It is not necessary for the printer to recognize any manual override by the display of uppercase and lowercase.
- Color may not match.

The local copy operation is described on the basis of a nonpartitioned screen; that is, for a display that does not have partitioning capability or for a display with partitioning capability that is in implicit partition state (implicit partition 0). Differences due to partitioning are described under "Partition Mode Considerations" in topic 8.6.

---

## 8.3 Copy Initiation

The host initiates a local copy by sending the display a write type command with the SP bit set to 1 in the WCC byte. If the write type command includes data, the screen is updated before print execution. Once a copy request is accepted, the display keyboard remains locked until the printer completes the print operation. The host is required to send the copy request either as an RQD chain or as an RQE, CD, not-EB chain; this prevents the host from following a copy request immediately with another command.

---

## 8.4 Printer Availability

When more than one printer is authorized for a display's use, a Copy request could result in the printout on any of the authorized printers. From the host view, however, there is only one logical printer, regardless of how many physical printers are authorized for the display's use; the host application has no control over the printer selected. The user must predefine (for example, IML) the printer configuration. A printer must be capable of being configured for system use only (no copy use), for copy use only, or for shared use between the system and copy.

A logical printer is considered available if one or more of the physical printers, configured for the display's use, can immediately execute a display printout. When none of the configured printers is available, the logical printer assumes the status of the most available physical printer. The following are the not-available categories listed in order of best to worst:

1. Short-term busy. A display printout can be executed after *n* queued display printout requests are executed. The maximum delay is 30 minutes. Use by a session is not allowed.
2. Intervention-required condition.
3. Allocation for session use or local use (not copy).
4. Permanent error condition (or an intervention-required situation on an unattended printer).
5. No logical printer configured.

The display rejects a copy request whenever the logical printer is not available. The following are the sense codes used:

082E The logical printer has an intervention-required condition; for example, out of paper, power off, and cover interlock open. This code is used if the printer is attended. In general, *attended* means the condition will likely receive quick attention; for example, the printer is located near a display. If the printer is unattended, the permanently unavailable code should be used.

0807 The logical printer is busy for an indeterminate period of time. This may range from a relatively short time when the printer is being used by another display for a display printout to many hours if all printers are in session. The display should mask short term busy from the host by withholding the `-rsp` (0807).

082F Effectively, the logical printer is permanently not available; for example, a hardware failure.

0801 No printer configured.

After sending an 082E or 0807, an LUSTAT must be sent when the condition clears. However, regardless of how many times the request is repeated, only one LUSTAT is sent when the condition does clear. Sending an LUSTAT 0001 (with `source=printer`) indicates to the host that a copy request will find either an available or short-term-busy logical printer. Exceptions are as follows:

- The LUSTAT 0001 (`source=printer`) reporting the clear of a 082E or 0807 condition is not sent if either of the following is true:
  - After sending 082E/0807 `-rsp`, the Data Traffic Subtree is reset or the session is terminated.

- After sending 082E/0807 -rsp, the logical printer develops a permanent error or becomes not configured.

An LUSTAT 081C/0801 (source=printer) is sent instead of LUSTAT 0001.

- The LUSTAT 0001 (source=printer) does not indicate the host will necessarily find an available or short-term-busy logical printer if one of the following is true:
  - After sending 082E/0807 -rsp, the SLU receives any normal FM data request chain other than a Copy request chain.
  - After sending LUSTAT 0001 (source=printer), the logical printer develops an intervention-required, permanent error, or not-configured condition. The Copy request will be rejected with the appropriate -rsp, that is, 082E, 082F, or 0801.
  - It is not valid for an implementation to allow session contention for a printer prior to honoring outstanding LUSTATs; that is, at least one physical printer must be held after having sent the LUSTAT(s) so that the logical printer is available to execute copy sessions. However, if as a result of an exception condition developing on the held printer the logical printer status of in session results, then 0807 -rsp may also be sent.
  - After sending 082E/0807 -rsp, the display detects a nonprinter exception condition which causes a -rsp to a received request.

The sense codes 082E, 0807, 082F, and 0801 indicate to the host that, if the copy request included screen update data, the screen update was accomplished. If a copy request (with update data) is re-sent, it cannot be guaranteed that the screen will be unchanged. An example where the screen would be changed is if the data used a positioning reference, for example, a Program Tab order that was revised later in the data stream.

---

## 8.5 Display/Printer Compatibility

In general, a print operation is not inhibited because of a mismatch in display/printer capability. However, a copy operation can be inhibited if the printer is unable to accept all the display data; for example, if the print buffer is too small, or if the printer maximum line length is less than the display width. When the copy is inhibited because the printer cannot accept all the display data, the printer is considered to be *not authorized*.

Subtopics:

- 8.5.1 APL Mismatch
- 8.5.2 Character Attribute/Extended Field Attribute (CA/EFA) Mismatch
- 8.5.3 Programmed Symbols (PS) Considerations
- 8.5.4 Extended Color Mismatch
- 8.5.5 Extended Highlighting Mismatch

---

### 8.5.1 APL Mismatch

Where the display has the APL feature, but the printer to which the copy is directed does not have APL, the copy operation occurs using the nonloadable character set. There will be no indication of this mismatch. Such mismatches can be avoided by the proper configuration of copy printers.

---

## 8.5.2 Character Attribute/Extended Field Attribute (CA/EFA) Mismatch

On a copy operation, the CA/EFA must be sent to the printer if the printer supports CA/EFA *and* if the display data to be copied references nonzero values of CA/EFA. When not all the extended functions are referenced in the copy data, an implementation can send only the required information or can send all the CA/EFA information. For example, if only extended color were referenced in the display data, that is, no character set, or extended highlighting selected, an implementation would send only the color information from the CA/EFA or all the CA/EFA information.

---

## 8.5.3 Programmed Symbols (PS) Considerations

A copy operation is not inhibited because of a mismatch of display/printer character set capability or character set load. Whenever the printer cannot print the data against the same loadable character set referenced by the display data, the print is executed against the nonloadable character set of the printer.

The Load PS structured field contains a compare field of 1 bit. If set to compare (bit=0), the LCID can be used in establishing a PS match in a copy operation; for a valid match, the matching LCIDs must both be set to compare. If set to no-compare (bit=1), the LCID cannot be used for establishing a match.

The following summarizes the copy printout operation:

**IF**           The referenced display PS is set to compare, and the printer has a matching LCID also set to compare, then execute printout against the matched LCIDs.

**ELSE**       Execute the printout against the printer default character set.

The printed data could have a mixed appearance, that is, some matching the displayed data and some not matching the displayed data.

---

## 8.5.4 Extended Color Mismatch

A copy operation is not inhibited because of a mismatch in color capability between the display and the printer; that is, no check is made for display/printer compatibility. If the printer does not support color, display data in color is printed in monochrome. If the printer supports color but a different set of colors than the display, the colors not supported are printed in the default color defined by the printer Query Reply color.

---

## 8.5.5 Extended Highlighting Mismatch

Copy is not inhibited because of a mismatch in extended highlighting capability between the display and the printer; that is, no check is made for display/printer extended highlighting capability. If the display specifies a particular form of highlighting not supported by the printer, the data is printed with defaults defined in the printer's Query Reply highlight.

---

## 8.6 Partition Mode Considerations

When the display is in partition mode with or without scrolling, the copy operation changes as follows:

An Outbound 3270DS structured field with a WCC=SP initiates a printout of the window of the designated partition. If a WSF command contains multiple structured fields, the WCC=SP can be set only in the last structured field. If set in any other structured field, the WSF is rejected with sense code X'1001'. The printout of the partition viewport has the same width as the display partition and is left-justified. If the display partition width exceeds the printer line length, the copy operation is rejected with sense code X'0801', no printer configured.

If a W, EW (reset off), or an EWA (reset off) is sent with WCC=SP to a partitioned screen, a printout of partition 0 occurs. If partition 0 does not exist, the copy operation is rejected with sense code X'1005', parameter error. If an EW (reset on) or an EWA (reset on) is sent with WCC=SP to a partitioned screen, the screen is reset to implicit partition 0.

---

## 8.7 Local Copy Command in the BSC Environment

In BSC, the local copy function is accomplished by use of the Copy command. This Copy command is used to transfer buffer data from one terminal to another terminal attached to the same controlling device. The selected terminal is the *to* terminal, the one to which the buffer data is transferred. The *from* terminal, the source of the buffer data to be copied, is identified in the second two bytes that follow the Copy command code; the first byte, called the *copy control character (CCC)*, identifies the type of data to be copied. The CCC can also, at the *to* device, start print operations, specify the printout format for those operations, and, when the terminal is a display station, sound the audible alarm.

The copy data stream is shown in [Figure 8-1 in topic 8.7](#).

[Table 8-1 in topic 8.7](#) describes the function of each CCC bit. A CCC and an address byte must always follow the command code. If they do not, the controlling device aborts the command and generates an error status.

The *from* terminal buffer can be locked (made incapable of being copied) by writing a protected/alphanumeric attribute byte (bit 2=1 and bit 3=0) in address 0.

The Copy command can specify as the *from* device the same device that is selected (the *to* device). This procedure provides a means of programming selective device buffer erase operations as specified by CCC bits 6 and 7.

**Notes:**

1. Copy should not be chained from a W, EW, EWA, or EAU command, since it copies the data as modified by the W or E command.
2. If the CCC start-print bit is set and commands are being chained, Copy should be the last command of the chain. If not, the controller aborts the subsequent command.
3. Copy can be executed from a smaller buffer to a larger buffer, but an attempt to copy from a larger to a smaller buffer causes an operation check.

If the Copy command refers to a source (*from*) terminal, in implicit partition state, that has not received an SFE, SA, MF, or GE, or on which the operator has not entered a character with an extended attribute, or a specific APL/TN character, since the last buffer clear (for example, EW, EWA commands), a copy action takes place. If the source terminal does not satisfy the above criteria, the Copy command is rejected with an operation check (OC) and unit specify (US) status, unless all the following conditions are met:

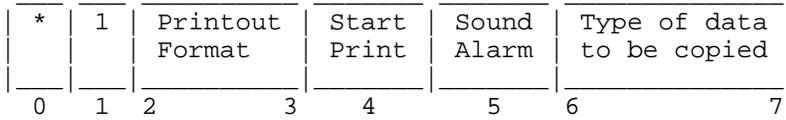
- The source terminal is a display.
- The destination terminal is a printer.
- The source terminal does not have a protected/alphanumeric field attribute in the first buffer position.
- The CCC has bits 4, 6, and 7 (start print and copy entire buffer) set to 1.

If these conditions are met, an attempt is made to produce a local copy. Following print completion, the print buffer is cleared and the appropriate completion status made available at the printer.

**Byte**

0	Start of text
1	Escape
2	Copy command code
3	CCC (see the following table)
4	Source device address
5	End of text

**CCC-Byte Format**



\* Determined by the configuration of bits 2 through 7

Figure 8-1. The Copy Data Stream

Table 8-1. Copy Control Character (CCC)	
Bit	Explanation
0,1	Defined to make the CCC a translatable character. (See <a href="#">Figure C-1 in topic C.0.</a> )
2,3	Define the printout format as follows:  00 The NL, EM, and CR orders in the data stream determine point line length. Provides a 132-print position line when the orders are not present. 01 Specifies a 40-character print line. 10 Specifies a 64-character print line. 11 Specifies an 80-character print line.
4	The start-print bit. When set to 1, initiates a printout operation at the to device after buffer transfers are completed.
5	The sound-alarm bit. When set to 1, sounds the audible alarm at the to device after buffer transfers are completed if that device has an audible alarm.
6,7	Define the type of data to be copied as follows:  00 Only attribute characters are copied. 01 Attribute characters and unprotected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the protected fields. 10 All attribute characters and protected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the unprotected fields. 11 The entire contents of the storage buffer (including nulls) are copied.

## 8.8 Valid LU 1 SCS Control Codes

SCS control codes that are valid for the LU 1 SCS Data Stream are defined by the acceptance of LU 1 BIND byte 18 (Data Stream Flags) and by indications in certain Query Replies, such as the descriptors returned in the Device Characteristics Query Reply. The valid SCS control codes are listed in [Table 8-2](#). All other SCS control codes are invalid for LU 1 usage.

Table 8-2. Valid LU 1 SCS Control Codes	
SCS Control Codes for LU 1	EBCDIC Code
Vertical Channel Select (VCS)	X'04'
Horizontal Tab (HT)	X'05'
Graphic Escape (GE)	X'08'
Vertical Tab (VT)	X'0B'
Form Feed (FF)	X'0C'
Carriage Return (CR)	X'0D'
Enable Presentation (ENP)	X'14'
New Line (NL)	X'15'
Backspace (BS)	X'16'
Interchange Record Separator (IRS)	X'1E'
Inhibit Presentation (INP)	X'24'
Line Feed (LF)	X'25'
Set Attribute (SA)	X'28'
Format (FMT)	X'2B'
Set Horizontal Format (SHF)	X'2BC1(L)(P)'
Set Vertical Format (SVF)	X'2BC2(L)(P)'
Set Line Density (SLD)	X'2BC6(L)(P)'
Set Text Orientation (STO)	X'2BD1(L)83(P)'
Set Print Density (SPD)	X'2BD2(L)29(P)'
Page Presentation Media (PPM)	X'2BD2(L)48(P)'
ASCII Transparent (ATRN)	X'2BFE(L)35(P)'
BELL (BEL)	X'2F'
Transparent (TRN)	X'35(L)(P)'
<b>Note:</b>	
L = length	
P = parameters	

The hierarchy of support allowed by the Data Stream Flags in byte 18 of the LU 1 Bind allows for the selection of the SCS control codes that a device chooses to support. Table 8-3 describes this hierarchy.

Table 8-3. LU 1 BIND Byte 18 (Data Stream Flags)	
Bit	Content Description
0	<b>B'0'</b> Base NL (New Line) FF (Form Feed)
	<b>B'1'</b> Full Base includes Base plus: BS (Backspace) CR (Carriage Return)

	LF (Line Feed) ENP (Enable Presentation) INP (Inhibit Presentation) HT (Horizontal Tab) VT (Vertical Tab)
1	SHF (Set Horizontal Format)  <b>B'0'</b> SHF must not be used  <b>B'1'</b> SHF may be used
2	SVF (Set Vertical Format)  <b>B'0'</b> SVF must not be used  <b>B'1'</b> SVF may be used
3	VCS (Vertical Channel Select)  <b>B'0'</b> VCS must not be used  <b>B'1'</b> VCS may be used
4	SLD (Set Line Density)  <b>B'0'</b> SLD must not be used  <b>B'1'</b> SLD may be used
5	Reserved
6	BEL (Bell)  <b>B'0'</b> BEL must not be used  <b>B'1'</b> BEL may be used
7	TRN (Transparent) and IRS (Interchange Record Separator)  <b>B'0'</b> TRN and IRS must not be used  <b>B'1'</b> TRN and IRS may be used

Support for other SCS control codes is indicated as follows:

- Graphic Escape (GE) support is indicated by setting byte 4, bit 1 to B'1' in the Character Sets Query Reply.
- Set Attribute (SA) support is indicated by returning the value, character mode (X'02') in one of bytes 4 to n in the Reply Mode Query Reply which commits support of character attributes.
- Set Line Density, Set Print Density, Page Presentation Media, Set Text Orientation, and ASCII Transparent support is indicated by descriptors in the Device Characteristics Query Reply.

---

## | 8.9 Early Print Complete (EPC) Operations

When page printers wait until the page is printed before returning a positive response, they will not run at their rated speed. The remedy is to allow them to return a positive response when the data has been received without error in the print buffer, but before the data is printed.

There exists a hazard that an error may occur during the printing of the last page of a print file. If this happens, the error will be reported on the next print job, not the one on which the error occurred, because the printer has responded that the print job has been successfully completed.

The architecture for Early Print Complete (EPC) was created to allow printers to run at their rated speed, while avoiding the erroneous response. EPC mode allows overlap of the loading of the printer buffer and the printing operation. Support of EPC mode is limited to the non-SNA environment (BSC and non-SNA local channel). In the SNA environment, LU1 may be used if the serial load and print required by LU3 gives performance problems with high speed printers.

EPC mode is set on or off by the SREPC flag (located in the Early Print Complete self defining parameter of the Settable Printer Characteristics structured field). Optionally, the printer may provide a printer operator control (eg. switch) in addition to the host application control.

When printer operator control for EPC mode on or off is not provided, the default mode is EPC mode off. When printer operator control is provided, the default is printer operator control enabled; i.e., EPC can be on or off depending on operator selection.

With the EPC mode off, the printer operates in a serial mode. The first time the buffer is loaded and when loading completes, the contents are printed. When the printing completes, the application is signalled print complete. The application then reloads the buffer and starts the print, etc.

The EPC mode permits overlap of the loading and printing which can significantly improve throughput in certain *bulk* print applications. When the EPC mode is set on, the printer is allowed to indicate print complete prior to actual completion provided sufficient buffering is available to accept the next buffer load. When the *early print complete* indication is given is implementation dependent.

The user of the EPC mode must be made aware of the effect on the exception condition reporting and the associated recovery. Several messages may be 'in the pipe' at one time. If an error is detected during the remainder of the printing which occurs after the *early print complete* indication is given, the next or a subsequent transmission will be rejected. Therefore, the sender must be aware that when the EPC mode is in effect, the receipt of a negative response may be the result of either the current or a prior transmission.

Prior to the last transmission (i.e., prior to starting the last print operation) of a job, the application should turn the EPC mode off. The print complete indication for the last buffer printed will then be given after the print operation actually completes.

When the printer operator control (eg. switch) is provided, conflict of control between the printer operator and the host application will be resolved in favor of the host application. The printer operator control is effective in turning the EPC mode on or off until a valid Set Printer

Characteristics structured field, with an EPC self defining parameter, is received which disables printer operator control (SREPC = bin.01' or B'10'). The printer operator control will then have no effect on setting the EPC mode on or off until one of the following occurs:

- A valid Set Printer Characteristics structured field with an EPC self defining parameter which has the SREPC flag = enable printer operator control (B'00') is received.
- A valid EW or EWA with WCC = reset is received.
- A BIND (SNA only).
- A POR.

Support of the early print complete function is indicated by the EPC self defining parameter on the Settable Printer Characteristics query reply. This self defining parameter also indicates whether the printer provides operator control for EPC mode on or off and if supported, whether the EPC mode has been set on or off by the operator.

---

## 8.10 IPDS Data/Non-SNA

This section describes the carrying of the IPDS by the 3270 data stream in a non-SNA environment. For the SNA environment, IPDS data stream is available through LU1.

When the 3270 data stream/IPDS is supported, the printer is in either a 3270 data stream mode or IPDS mode. The 3270 type data and controls and IPDS type data and controls cannot be mixed. In other words, you cannot send 3270-type data and IPDS-type data and then print the composite.

The 3270 IPDS Query Reply indicates the support of IPDS and also defines the maximum transmission size allowed outbound in the IPDS mode. However, the mechanisms provided by the IPDS (for example, STM/ACK REPLY) are used to provide IPDS-related printer characteristics. Also, in the IPDS mode, the exception handling functions provided by the IPDS are used.

Changing modes clears the printer buffers of any data associated with the previous mode. Use of the printer for 3270 local copy is considered to be 3270 mode. For example, if the printer buffer was loaded in the 3270 mode and the printer changed to the IPDS mode, the 3270 data would be cleared. Therefore, a return to the 3270 mode would find a cleared buffer. However, the printer must maintain forms sync across modes as is done between 3270 and SCS. See the 3270 IPDS Query Reply for non-SNA systems on page 6.52 and the Data Streams Query Reply for SNA systems on page 6.16 for more information about IPDS selection.

Subtopics:

- 8.10.1 IPDS Selection
- 8.10.2 Local Copy
- 8.10.3 Pacing
- 8.10.4 Input Transmissions

---

## 8.10.1 IPDS Selection

The default (for example, power on) mode is the 3270 mode. In the 3270 mode, the structured fields defined in the IPDS are rejected. When a device is in the IPDS mode, any non-IPDS structured field (except a valid Data Chaining or Select IPDS structured field) is invalid (refer to the IPDS manual for action taken on invalid structured fields). Also, in the IPDS mode, 3270 orders and control sequences are treated as IPDS data with unpredictable results.

Support of IPDS requires that data chaining (for example, the Data Chain structured field and the Data Chaining Query Reply) be supported. See "Structured Field Grouping" in topic 5.2.

If the Data Chain structured field is not used in the selection sequence (that is, no data chaining), the IPDS mode exists until the end of the transmission. If the Data Chain structured field is used (that is, data chaining), the IPDS mode exists until the end of the data chain.

The IPDS mode is selected by the Select IPDS Mode structured field. The Select IPDS Mode structured field also resets IPDS controls and conditions to the default values. On outbound transmissions (to the device), the Select IPDS Mode structured field must immediately follow a WSF, or immediately follow a WSF Data Chain structured field (GROUP = begin) sequence. On inbound transmissions (from the device), the Select IPDS Mode must immediately follow an AID X'88' or immediately follow an AID X'88'; that is, a Data Chain structured field (GROUP = begin) sequence.

In data chaining, the Select IPDS Mode structured field only appears immediately after the Data Chain structured field of the first transmission of the data chain. That is, the IPDS mode is continued by the subsequent Data Chain structured fields (GROUP = continue) and terminates at the completion of the last Data Chain structured field (GROUP = end). For example, IPDS structured fields and control sequences can span transmissions. If the printer is not in the IPDS mode, a Select IPDS Mode structured field appearing anywhere, except after a WSF command or WSF Data Chain structured field (GROUP = begin) sequence, is rejected. If the printer is in the IPDS mode, a Select IPDS Mode structured field received anywhere except after a Data Chain structured field (GROUP = begin) sequence is treated as an invalid structured field and handled in accordance with the IPDS.

With the exception of the IPDS ACK structured field which reports exception and status conditions, IPDS is an output-only data stream. Sending of the ACK structured field by the device is limited to either of the following:

- Responding to an IPDS outbound structured field that requests a reply
- Reporting exception conditions that were detected while processing an outbound IPDS transmission.

---

## 8.10.2 Local Copy

When in the IPDS mode, the printer cannot be used for operator- initiated local copy. If it is in IPDS mode and the BSC Copy command is received,

the IPDS mode is terminated without error and the copy executed.

---

### **8.10.3 Pacing**

The non-SNA protocols do not provide the chaining and pacing functions provided by SNA. Control of data to the printer in the IPDS mode is accomplished by a combination of data chaining, limiting the transmission size, and link and channel controls.

The 3270 IPDS Query Reply indicates the maximum allowed transmission length during the IPDS mode. The data chaining function allows a message to be divided into transmissions of suitable length without regard to structured field or control boundaries, for example.

The IPDS mode uses the same BSC and channel controls as does the 3270 mode to determine when more data can be sent to the printer. In BSC, a wait before transmit positive acknowledgment (WACK) is returned after acceptance of a transmission if the printer cannot accept more data at this time. When the printer is ready for another transmission, a device end (DE) is returned by the device. In a non-SNA local channel, if the printer cannot accept more data at this time, channel end (CE without DE) is returned after acceptance of a transmission. When the printer is ready for another transmission, a DE is returned to the device.

---

### **8.10.4 Input Transmissions**

Although the IPDS data stream is essentially an output-only data stream, the printer generates some inbound data in the form of an acknowledgment reply. This reply can be generated asynchronously (exception reporting) or synchronously (in reply to a host request).

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 9.0 Chapter 9. Binary Synchronous Communications (BSC) Environment

Subtopics:

- 9.1 Introduction
- 9.2 Transparent Mode
- 9.3 Write Commands
- 9.4 Read Commands
- 9.5 Inbound Transmissions
- 9.6 Inbound Operation (INOP)
- 9.7 Read States
- 9.8 Indicators
- 9.9 Read Acknowledgments
- 9.10 Processing of Read Commands
- 9.11 Processing of Read Partition Query Structured Fields
- 9.12 BSC Copy Command

---

### 9.1 Introduction

This chapter describes how the 3270 data stream operates in a binary synchronous communication (BSC). Except for the differences described in this chapter, the data stream operates the same way as in the SNA environment.

---

### 9.2 Transparent Mode

The SNA environment provides inherent transparency on the data line. This mode of operation permits greater versatility in the range of coded data that can be transmitted, because all data, including the normally restricted data link line control characters, are treated as bit patterns when transmitted in transparent mode. This method of transmission is required when transmitting binary counts and addresses that can appear in the data stream for the functions of extended highlighting, color, and programmed symbols.

BSC data links may be nontransparent or transparent. However, when the SF, SFE, and MF orders and the WSF command are supported, the BSC link must be in the transparent mode. The format of the data stream for BSC is as follows:

For nontransparent mode:

STX	3270 Data Stream	ETX
-----	------------------	-----

For transparent mode:

DLE	STX	3270 Data Stream	DLE	ETX
-----	-----	------------------	-----	-----

---

## 9.3 Write Commands

The EW and EWA commands operate the same in a BSC environment as in the SNA environment. For the W command, the only difference is that the starting buffer location depends upon the following considerations:

- The starting location can be specified by an SBA order that follows the WCC.
- The starting location is the buffer address containing the cursor if the W command is not chained from a Copy or EAU.
- The starting location is the current buffer address if the W command is chained from a read or another write command.

---

## 9.4 Read Commands

The read commands operate in the same way in a BSC environment as in the SNA environment except for the differences that follow.

Subtopics:

- [9.4.1 Read Buffer Command](#)
- [9.4.2 Read Modified Command](#)
- [9.4.3 Test Request Read](#)

---

### 9.4.1 Read Buffer Command

Execution of the Read Buffer command causes all the data in the addressed device's buffer, from the buffer location at which reading starts through the last buffer location, to be transferred to the application program's storage. The transfer of data begins as follows:

- From buffer address 0 if the Read Buffer command is unchained.

- From the current buffer address if the Read Buffer command is chained from either a W, EW, EWA, Read Modified, or another Read Buffer command. Regardless of where the transfer of data begins, data transfer from the buffer terminates when the last character location in the buffer has been transferred, or when the last character of a text block has been transferred.

---

## 9.4.2 Read Modified Command

Read Modified initiates one of three operations as determined by operator actions at the display: (1) read modified, (2) short read, or (3) test request read.

Read Modified functions the same way as in an SNA environment except the buffer location at which the search begins for field attributes that define modified fields is a function of command chaining. This location is one of the following:

- Buffer address 0 if the Read Modified command is unchained or is chained from a Copy.
- The current buffer address if the Read Modified command is chained from a W, EW, Read Modified, Read Modified All, or a Read Buffer command.

The search for modified field attributes ends when the last buffer location is checked.

The transfer of read data is terminated as follows:

- If the last modified field is wrapped from the last buffer location (for example, 479 or 1919) to the first location, the operation is terminated after all the data in the field is transferred (nulls are suppressed). The buffer address at the end of the operation is the address of the next field attribute byte in the buffer. For example, if a modified field extends from address 1900 (the field attribute) to address 79 (wrapped field), the data from address 1901 through 79 is transferred (nulls suppressed). In this case, the read operation is terminated with the buffer address set to 80 (the field attribute byte of the next field).
- If the buffer does not contain a wrapped modified field, the modified data stream is terminated when the last modified field is transferred. At the end of the operation, the buffer address is set to 0.

If the buffer is formatted but none of the fields have been modified, the read data stream consists of the 3-byte read heading (AID plus cursor address).

If the buffer is unformatted, the read data stream consists of the 3-byte read heading followed by all the alphanumeric data in the buffer (nulls suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no field attribute bytes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of data cannot be determined. Data transfer starts at address 0, regardless of command chaining, and continues to the end of the

buffer. At the end of the operation, the buffer address is set to 0.

This read operation can also be terminated by the channel byte count's reaching 0 before all data is read. In this case, the buffer address after termination is undefined.

---

### 9.4.3 Test Request Read

The Read Modified command causes a test request read operation if the TEST REQ or SYS REQ key has been pressed at the selected device. The test request read data stream sent inbound to the application program is as follows:

SOH
%
/
STX
Input Data
ETX

The Test Request Read heading is generated by the display. The remainder of the data stream is the same as described previously for read-modified operations, excluding the 3-byte read heading (AID and cursor address). If the buffer is unformatted, all the alphanumeric data in the buffer is included in the data stream (nulls suppressed), starting at address 0. If the buffer is formatted, each field-attribute byte is examined for a set MDT bit. Each time a set MDT bit is found, the alphanumeric data in the field associated with that bit is sent to main storage (nulls suppressed). If no MDT bits are set, the read data stream consists of the Test Request Read heading only. The buffer location at which the search for the MDT bits begins and the transfer of data ends is the same as described for the read-modified operations.

**Note:** Usage of the test request read function is determined by the application program. Normally, the operator would clear the display, enter test request data in a predefined format, and press the TEST REQ or SYS REQ key.

---

## 9.5 Inbound Transmissions

Inbound transmissions result from an operator enter action, an application program initiated (*unsolicited*) read request, or a host retry of an inbound transmission.

An operator enter action is one that causes an attention identifier to be transmitted inbound. The application program responds with a read request. The application program must acknowledge the inbound transmission before a new inbound operation can be performed.

An unsolicited read operation is an inbound transmission not caused by an operator enter action. No application program acknowledgment is required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device. The application program must acknowledge receipt of an inbound transmission before a new inbound transmission can take place.

A host retry transmission does not cause read state transitions and is not considered a new inbound transmission requiring acknowledgment by the application program. Host retry occurs until an application program acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the response to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation device characteristic (INOP), set by the controller, defines the type.

---

## 9.6 Inbound Operation (INOP)

The INOP determines the operation to be performed when data is transmitted inbound on a retry transmission.

INOP is set by any of the following:

- An operator enter action sets INOP to Read Modified.
- Reception of a Read Partition Query structured field sets INOP to Query.
- Application program acknowledgment of an inbound transmission sets INOP to Read Modified.

---

## 9.7 Read States

While power is on, a device is in one of seven states with respect to read operations. The three primary states are as follows:

- Normal read
- Data pending read
- Read retry.

The data pending and read retry states have the following three substates:

- Enter
- Read
- Stacked enter.

The events that cause transitions between states are shown in Table 9-1 in topic 9.7.3.

Subtopics:

- [9.7.1 Normal Read State](#)
- [9.7.2 Data-Pending States](#)
- [9.7.3 Read-State Transitions](#)
- [9.7.4 Retry States](#)

## 9.7.1 Normal Read State

A device is in normal read state when power is on, prior to initiation of a new read operation, and after use of the Reset key in certain instances. (See "[Read-State Transitions](#)" in topic 9.7.3.)

When in normal read state an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream and to go into a data pending state.

When in normal read state, an application program initiated read operation using an RM, RMA, or RB command causes the data to be transmitted with no state transitions occurring. The device remains in normal read state.

## 9.7.2 Data-Pending States

The three data pending states are as follows:

Enter            The device state after an operator enter action

Read            The device state after reception of a Read Partition Query structured field

Stacked enter    The device state after a Read Partition Query structured field was received while the device was in data pending enter read state or retry enter state (the enter is stacked).

A poll received while the device is in a data pending state causes the data to be transmitted and the device to be placed in the corresponding retry state.

## 9.7.3 Read-State Transitions

The read state transitions for BSC are summarized in [Table 9-1](#).

Table 9-1. Read State Transitions for BSC							
Read States	Data Pending			Retry			
Events	Normal	Enter	Read	Stacke Enter	Enter	Read	Stacke Enter
Enter Action	2	R	R	R	R	R	R

Read Command	1	1	1	1	G	G	G
Read Partition Query	3	4	R	R	4	R	R
Poll	--	5	6	7	--	--	--
Host Acknowledge	--	1	1	2	1	1	2
Reset key	--	1	--	3	1	--	6

**Legend:**

R Reject, no state transition  
G Retry, no state change  
-- No action or state change  
1 Normal read state  
2 Data pending enter state  
3 Data pending read state  
4 Data pending stacked enter state  
5 Retry enter state  
6 Retry read state  
7 Retry stacked enter state

## 9.7.4 Retry States

There are three retry states:

**Enter**            The device state after enter data was transmitted to the application program

**Read**            The device state after Query Reply data was transmitted to the application program

**Stacked enter** The device state after enter data was stacked and the Query Reply data transmitted to the application program.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal read state or, in the case of retry stacked enter, to the data pending stacked enter read state.

## 9.8 Indicators

With reference to [Table 9-1](#) in [topic 9.7.3](#), the indicators displayed in the operator information area of a display are as follows:

<b>State</b>	<b>Indicator</b>
Normal read	No indicator or System Lock

Data pending enter TWAIT  
Data pending read TWAIT  
Data pending stacked enter TWAIT  
Retry enter System Lock  
Retry read TWAIT  
Retry stacked enter TWAIT.

---

## 9.9 Read Acknowledgments

Read acknowledgements depend on the inbound operation. The following sections describe the read acknowledgements when the inbound operation is a Query, Query List, Operator Enter, RM, RMA, or RB Partition command.

Subtopics:

- [9.9.1 Query or Query List](#)
  - [9.9.2 Operator Enter or RM, RMA, or RB Partition Command](#)
- 

### 9.9.1 Query or Query List

A Query or Query List operation is acknowledged by any outbound transmission except one with a read command.

The following commands acknowledge a Query or Query List:

- A WSF command with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields, as long as the WSF is accepted.
  - An EAU command.
  - An EW, EWA, or W with or without a WCC or data. If data is present and an error is detected in the data, the transmission is *not* an acknowledgment.
- 

### 9.9.2 Operator Enter or RM, RMA, or RB Partition Command

An RM, RMA, or RB Partition command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission which restores the keyboard, or destroying the inbound partition.

Any of the following constitutes an acknowledgment when the display is in any of the retry or data pending states:

- If the inbound partition is 0 (implicit or explicit), either of the following:

- An EW, EWA, or Write command with WCC = Keyboard Restore (see note 1).
- An EAU command.
- If the display is in the explicit partition state, an EW or EWA command with the WCC = reset (see note 1).
- A WSF command followed by an outbound 3270DS structured field to the inbound partition with either of the following (see note 2):
  - An EW, EWA, or Write partition command with WCC = Keyboard Restore
  - EAU partition command.
- A WSF command followed by a Destroy Partition structured field to the inbound partition, including explicit and implicit partition 0 (see note 2).
- A WSF command followed by a Create Partition structured field to the inbound partition (see note 2).
- If in implicit partition state, a WSF command followed by a Create Partition structured field (see note 2).
- A WSF command followed by an Erase/Reset structured field. Applies to both implicit and explicit partition state (see note 2).
- A Copy command (BSC only).

In addition, the following constitute a read acknowledgment only when the display is in one of the retry states:

- A WSF command followed by an outbound 3270 DS structured field, with an EW, EWA, W or EAU partition command, directed to any partition. This acknowledgement applies whether or not the EW, EWA, or W partition command is followed by a WCC or data (see note 2).
- An EW, EWA, Write, or EAU command. This acknowledgement applies whether or not a WCC or data follows the EW, EWA or W command (see note 1).

**Notes:**

1. If data follows the WCC and an error is detected in the data, the transmission is not a read acknowledgment.
2. If there is a detected error prior to, or within, the structured field providing the read acknowledgment, the transmission is not a read acknowledgment.

---

## 9.10 Processing of Read Commands

In a BSC environment, a read command (Read Buffer, Read Modified, or Read Modified All) as the first byte of the data stream is processed as follows:

- If the device is in normal read state, then the command performs a read and the display data is transmitted inbound as defined by the following:

1. The Read Modified, Read Modified All, or Read Buffer command
2. The AID (Read Modified command only)
3. The reply mode.

The device remains in normal read state.

- If the device is in a data pending state, then the command performs a read and data is transmitted as defined by:

1. The Read Modified, Read Modified All, or Read Buffer command
2. The AID (Read Modified command only)
3. The inbound reply mode.

The device is placed in normal read state.

- If the device is in a retry state, then the command performs a retry as follows:

1. If the command is Read Modified, and INOP specifies Query, then appropriate Query Replies are transmitted inbound.
2. If the command is Read Modified, and INOP specifies Read Modified, data is transmitted as defined by the following:
  - a. The Read Modified command
  - b. The AID
  - c. The inbound reply mode.

3. If the command is Read Buffer, data is transmitted inbound as defined by the following:
  - a. The Read Buffer command
  - b. The inbound reply.

The device remains in the original retry state.

---

## 9.11 Processing of Read Partition Query Structured Fields

Read Partition Query and the Query Reply structured fields are processed as follows:

If the device is in normal read state, the following occurs:

1. The TWAIT indicator is displayed.
2. INOP is set to Query.
3. The device prepares to generate the required inbound data stream.
4. The device is placed in data pending read state.
5. A later poll causes the data to be transmitted and the device to be placed in retry read state.

If the device is in data pending enter or retry enter state, the following occurs:

1. The outstanding data is stacked.
2. The TWAIT condition remains in effect.
3. INOP is set to Query.
4. The device prepares to generate the required inbound data stream.
5. The device is placed in data pending stacked enter state.
6. A later poll causes the Query Reply data to be transmitted inbound and the device to be placed in retry stacked enter state.

---

## 9.12 BSC Copy Command

BSC uses the Copy command to accomplish the Local Copy function. See Chapter 8, "Printer Operations" for a detailed description of the operation of the Copy command.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 10.0 Chapter 10. Non-SNA Environment (Locally Attached Devices--3272 Version)

Subtopics:

- 10.1 Introduction
- 10.2 Commands
- 10.3 Inbound Transmissions
- 10.4 Inbound Operation (INOP)
- 10.5 Read States
- 10.6 Indicators
- 10.7 Read Acknowledgments - Non-SNA
- 10.8 Processing of Read Commands
- 10.9 Processing of Read Partition Query Structured Fields

---

### 10.1 Introduction

This chapter describes the data stream operations that differ from the SNA version for locally attached devices in a non-SNA environment. Most of the explanations cover only the differences between SNA and non-SNA for locally attached devices.

---

### 10.2 Commands

Except for the Read Modified All command, the commands valid for an SNA environment are also valid for the non-SNA locally attached environment.

The additional commands of Select, No Operation (No-op), and Sense are used in the non-SNA locally attached environment to improve device utilization, to retrieve pending status, and to obtain unit check definition, respectively. These commands are not 3270 data stream commands and are not discussed in this book. See the applicable product manuals for an explanation of these commands.

Subtopics:

- 10.2.1 Write Commands
- 10.2.2 Read Commands

---

## 10.2.1 Write Commands

The EW and EWA commands operate the same as in an SNA environment. However, the starting buffer location of the W command depends on the following considerations:

- The starting location can be specified by an SBA order that follows the WCC.
- The starting location is the buffer address containing the cursor if the Write command is not chained or if it is chained from a Select, EAU, No-Op, or Sense command.
- The starting location is the current buffer address if the write command is chained from a read or another write command.

---

## 10.2.2 Read Commands

The read commands for devices in this non-SNA environment operate in the same way as for an SNA environment except for the differences described in the following sections.

Subtopics:

- 10.2.2.1 Read Buffer Command
- 10.2.2.2 Read Modified Command

---

### 10.2.2.1 Read Buffer Command

Execution of the Read Buffer command causes all data in the addressed device buffer location to be transferred to main storage. The transfer of data begins as follows:

- From buffer address 0 if the Read Buffer command is unchained
- From the current buffer address if the Read Buffer command is chained from either a W, EW, EWA, Read Modified, or another Read Buffer command.

Data transfer from the buffer terminates when the last character location in the buffer has been transferred or before the last character location has been transferred when the channel byte count reaches 0 (in this case, the buffer address after termination is undefined).

---

### 10.2.2.2 Read Modified Command

Read Modified initiates one of three operations as determined by operator

actions at the display: Read Modified, Short Read, or Test Request Read.

*Read-Modified Operation:* The Read Modified command functions in the same way as in an SNA environment except that the buffer location at which the search begins for the field attribute bytes that define modified fields is a function of command chaining. This location is as follows:

- Buffer address 0 if the Read Modified command is unchained or is chained from a Select, Sense, or No-Op command
- The current buffer address if the Read Modified command is chained from a W, EW, Read Modified, or Read Buffer command.

The search for modified fields ends when the channel byte count reaches 0. The transfer of data is terminated as follows:

1. If the last modified field is wrapped from the last buffer location, the operation is terminated after all the data in the field has been transferred (nulls are suppressed). The buffer address at the end of the transfer is the address of the next field attribute byte in the buffer. For example, if a modified field extends from address 1900 (the field attribute byte) to address 79 (wrapped field), the data from addresses 1901 through 79 is transferred (nulls suppressed). In this case, the read operation is terminated with the buffer address set to 80 (the field attribute byte of the next field).
2. If the buffer does not contain a wrapped modified field and if the channel byte count has not reached 0, the modified data stream is terminated when the last modified field is transferred. At the end of the operation, the buffer address is set to 0.
3. If the channel byte count reaches 0 before all modified data is transferred, read operations are terminated and the remaining modified data is not transferred. The buffer address after termination is undefined.

If the buffer is formatted but none of the fields have been modified, the read data stream consists of the 3-byte read heading only.

If the buffer is unformatted, the read data stream consists of the 3-byte read heading followed by all alphanumeric data in the buffer (nulls suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no field attributes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of the data cannot be determined. Data transfer starts at address 0, regardless of command chaining, and continues to the end of the buffer. At the end of the operation, the buffer address is set to 0. This read operation can also be terminated by the channel byte count's reaching 0 before all data is read. In this case, the buffer address after termination is undefined.

*Short-Read Operation:* The Read Modified command causes a short read operation if the Clear, CNCL (Cancel), or a PA key has been pressed at the selected device. During the short read operation, only an AID byte is transferred to the application program. This AID byte identifies the key that was pressed.

*Test Request Read Operation:* The Read Modified command causes a Test Request Read operation if the TEST REQ or SYS REQ key has been pressed at the selected device. The Test Request Read data stream sent inbound to the application program is the same as for the BSC environment, except there is no ETX.

---

## 10.3 Inbound Transmissions

Inbound transmissions result from an operator enter action, a host initiated (unsolicited) read request, or a host retry of an inbound transmission.

An operator enter action is one that causes the attention identifier to be transmitted inbound. The application program responds with a read request. The application program must acknowledge the inbound transmission before a new inbound operation can be performed.

A read operation initiated by the host is an inbound transmission not caused by an operator enter action. Host acknowledgment is not required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device. The application program must acknowledge reception of an inbound transmission before a new inbound transmission can take place. A host retry transmission does not cause read state transitions and is not considered a new inbound transmission requiring host acknowledgment. Host retry occurs until a host acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the reply to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An INOP, set by the controller, defines the type.

---

## 10.4 Inbound Operation (INOP)

The INOP determines the operation to be performed when data is transmitted inbound or when the device is in a data pending state.

INOP is set by any of the following:

- An operator enter action sets INOP to Read Modified.
  - Reception of a Read Partition Query structured field sets INOP to query.
  - Host acknowledgment of an inbound transmission sets INOP to Read Modified.
- 

## 10.5 Read States

While power is on, a device is in one of seven states with respect to read

operations. The three primary states are as follows:

- Normal read
- Data pending read
- Retry read.

The data pending read and retry read states have the following three substates:

- Enter
- Read
- Stacked enter.

The events that cause transitions between the states are shown in Table 9-1 in topic 9.7.3.

Subtopics:

- 10.5.1 Normal Read State
- 10.5.2 Data Pending States
- 10.5.3 Retry State

---

## 10.5.1 Normal Read State

A device is in normal read state when power is on, prior to initiation of a read operation, or after use of the Reset key in certain instances. (See Table 10-1 in topic 10.5.3.)

When in normal read state, an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream, and then to go into a data pending state.

When in normal read state an application program initiated read operation using an RM, RMA, or RB command causes the data to be transmitted with no state transitions occurring. The device remains in normal read state.

---

## 10.5.2 Data Pending States

The three forms of data pending states are as follows:

Enter            Enter data has been transmitted to the application program.

Read            The device state after reception of a read partition query structured field.

Stacked enter    When a read partition structured field has been received while the device is in data pending enter state or retry enter state (the enter data is stacked).

A read command received while the device is in data pending state causes the data to be transmitted inbound and the device to be placed in the corresponding retry state.

An operator enter action is not processed, it is rejected. A host initiated read partition will be rejected.

### 10.5.3 Retry State

The three forms of retry states are as follows:

Enter            When enter data has been transmitted to the host.

Read            When the read data has been transmitted to the host.

Stacked enter   When enter data has been stacked and the Query Reply data has been transmitted to the host.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal read state or, in the case of retry stacked enter, to the data-pending-stacked-enter-read state.

The read state transitions for non-SNA locally attached devices are summarized in Table 10-1.

Table 10-1. Read-State Transitions for Non-SNA Locally Attached Devices--3272 Version							
Read States		Data Pending			Retry		
Events	Normal	Enter	Read	Stacke Enter	Enter	Read	Stacke Enter
Enter Action	2	R	R	R	R	R	R
Read Command	1	5	6	7	G	G	G
Read Partition Query	3	4	R	R	4	R	R
Host Acknowledge	--	1	1	2	1	1	2
Reset key	--	1	--	3	1	--	6

**Legend:**

R Reject, no state transition  
G Retry, no state transition  
-- No action or state change

- |   |                                  |
|---|----------------------------------|
| 1 | Normal Read state                |
| 2 | Data pending enter state         |
| 3 | Data pending read state          |
| 4 | Data pending stacked enter state |
| 5 | Retry enter state                |
| 6 | Retry read state                 |
| 7 | Retry stacked enter state        |

---

## 10.6 Indicators

With reference to Table 9-1 in topic 9.7.3, the indicators displayed in the operator information area of a display are as follows:

State	Indicator
Normal read state	No indicator or System Lock
Data pending enter	System Lock
Data pending read	TWAIT
Data pending stacked enter	TWAIT
Retry enter	TWAIT
Retry read	TWAIT
Retry stacked enter	TWAIT.

---

## | 10.7 Read Acknowledgments - Non-SNA

Read acknowledgements depend on the inbound operation. The following sections describe the read acknowledgements when the inbound operation is a Query, Query List, Operator Enter, RM, RMA, or RB Partition command.

Subtopics:

- 10.7.1 Query or Query List
- 10.7.2 Operator Enter or RM, RMA, or RB Partition Command

---

### 10.7.1 Query or Query List

A Query or Query List operation is acknowledged by any outbound transmission except one with a read command.

The following commands acknowledge a Query or Query List:

- A WSF command with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields, as long as the WSF is accepted.
- An EAU command.
- An EW, EWA, or W command with or without a WCC or data. If data is present and an error is detected in the data, the transmission is not an acknowledgment.

---

## 10.7.2 Operator Enter or RM, RMA, or RB Partition Command

An RM, RMA, or RB Partition command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission which restores the keyboard or destroying the inbound partition.

Any of the following constitute an acknowledgment when the display is in any of the retry or data pending states:

- If the inbound partition is implicit or explicit partition 0, either of the following:
  - An EW, EWA, or W command with WCC = Keyboard Restore (see note 1)
  - An EAU command.
- If in the explicit partition state, an EW or EWA command with the WCC = reset (see note 1).
- A WSF command followed by an outbound 3270DS structured field to the inbound partition with either of the following (see note 2):
  - An EW, EWA, or W partition command with WCC = Keyboard Restore
  - An EAU partition command.
- A WSF command followed by a Destroy Partition structured field to the inbound partition, including explicit and implicit partition 0 (see note 2).
- A WSF command followed by a Create Partition structured field to the inbound partition (see note 2).
- If in implicit partition state, a WSF command followed by a Create Partition structured field (see note 2).
- A WSF command followed by an Erase/Reset structured field. This acknowledgement applies to both implicit and explicit partition state (see note 2).

### Notes:

1. If data follows the WCC and an error is detected in the data, the transmission is not a Read Acknowledgment.
2. If there is a detected error prior to, or within, the structured field providing the Read Acknowledgment, the transmission is not a Read Acknowledgment.

---

## 10.8 Processing of Read Commands

In a non-SNA (locally attached devices--3272 Version) environment, a read command (Read Buffer or Read Modified) as the first byte of the data stream is processed as follows:

- If the device is in normal read state, the command performs a read and the display data is transmitted inbound as defined by:
  - The Read Modified, or Read Buffer command
  - The AID (Read Modified command only)
  - The reply mode.

The device remains in normal read state.

- If the device is in a data pending state, the commands are processed as follows:
  - If the command is Read Modified and INOP specifies Query or Query List, the appropriate Query Replies are transmitted.
  - If the command is Read Modified and INOP specifies a Read Modified, then data is transmitted as defined by:
    - The Read Modified command
    - The AID
    - The inbound reply mode.
  - If the command is Read Buffer, then data is transmitted as defined by the following:
    - The command
    - The reply mode.

The device is placed in the corresponding retry state (Enter, Read, or Stacked Enter).

- If the device is in a retry state, the command performs a retry as follows:
  - If the command is Read Modified, and INOP specifies Query or Query List, then the appropriate Query Replies are transmitted inbound.
  - If the command is Read Modified, and INOP specifies Read Modified, then data is transmitted as defined by the following:
    - The Read Modified command
    - The AID
    - The inbound reply mode.
  - If the command is Read Buffer, then data is transmitted inbound as defined by the following:

- The Read Buffer command
- The inbound reply.

The device remains in the original retry state.

---

## 10.9 Processing of Read Partition Query Structured Fields

Read Partition Query and the Query Reply are processed as follows:

If the device is in normal read state:

1. The TWAIT indicator is displayed.
- | 2. INOP is set to Query or Query List.
3. A channel attention occurs.
4. The device is placed in data pending read state.
5. A later read command causes the data to be transmitted and the device to be placed in retry read state.

If the device is in data pending enter or retry enter state:

1. The outstanding data is stacked.
  2. The TWAIT condition remains in effect.
  - | 3. INOP is set to Query or Query List.
  4. A channel attention occurs.
  5. The device is placed in data pending stacked enter state.
  6. A later Read Modified command causes the data to be transmitted inbound and the device to be placed in retry stacked enter state.
-

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 11.0 Chapter 11. Auxiliary Devices and Workstations

Subtopics:

- 11.1 Introduction
- 11.2 Data Routing
- 11.3 Query Reply
- 11.4 Input Control
- 11.5 Auxiliary Device and Display Interaction
- 11.6 Exception Handling

---

### 11.1 Introduction

This chapter describes how the 3270 data stream supports devices attached to a 3270 display.

The 3270 data stream was defined for use between a host application program and a single display. A *3270 workstation* consists of a 3270 display and one or more auxiliary devices. Auxiliary devices do not accept 3270 data stream commands and orders. However, the 3270 data stream is used to carry data to and from the auxiliary devices. The data must be in the form of structured fields.

The presentation space associated with an auxiliary device is independent of the display presentation space. Data directed to an auxiliary device must not alter the display presentation space and vice versa.

This enhancement of the 3270 data stream is based on a single session between the host application and the 3270 workstation. The workstation has only one network address. (The extensions for supporting a workstation also apply to the non-SNA environment.)

The term device in this section is used in a general sense; that is, a device can be either of the following:

- An actual device (for example, a printer)
- A logical device or process (for example, a DDM or DIA file).

An auxiliary device can support either an IBM data stream (for example, DDM or DIA file) or a non-IBM data stream.

*IBM Data Streams:* A different type auxiliary device Query Reply is defined for each different IBM data stream used by the auxiliary devices, for example, the DDM Auxiliary Device and DIA Auxiliary Device Query Replies.

The Query Reply provides a reference that identifies the IBM data stream supported. For example, the DIA Query Reply indicates the DIA function sets that are supported. The actual description of the data stream is provided by the IBM documentation associated with the data stream.

*Non-IBM Data Streams:* The 3270 data stream workstation can support a non-IBM device, such as an auxiliary device. A non-IBM device is defined here as a device that is manufactured outside IBM and does not use an IBM data stream. The device may carry either an outside manufacturer's logo or an IBM logo.

Only one Query Reply, the OEM Auxiliary Device Query Reply, is defined for all types of non-IBM auxiliary devices.

An OEM Data structured field is provided for carrying the data to and from a non-IBM auxiliary device. A parameter (DSREF) in the OEM Auxiliary Device Query Reply identifies the content of the OEM Data structured field as one of the following:

- A non-IBM data stream that the non-IBM auxiliary device sends or receives. The host application must derive what this data stream is from the DTYPE (Device type) parameter in the OEM Auxiliary Device query reply.
- A value added data stream; that is, a data stream that contains controls in addition to the data recognized or sent by the device. The controls are used by the 3270 data stream workstation. These controls are removed by the workstation prior to sending the non-IBM data stream to the auxiliary device. The DSREF parameter of the OEM Auxiliary Device Query Reply provides reference identification for the "value added" data stream.

---

## 11.2 Data Routing

In single display device implementations there is no ambiguity about the destination or source of the data. The using environment provides the routing, for example, via the network address in an SNA environment.

However, with the 3270 data stream workstation there is more than one destination or source for data. The destination or source must be explicitly identified. The using environment (for example, SNA) provides the same routing function for a 3270 data stream workstation as is currently provided for the single display implementation. Additional routing control for data is provided within the 3270 data stream.

The Destination/Origin structured field must be used to identify the destination or origin of all data to or from auxiliary devices. The same

DOID value is used in the Destination/Origin structured field for sending data to or from a particular auxiliary device. Outbound, from the host application, the DOID indicates the destination of the data. Inbound, to the host application, the DOID indicates the origin of the data.

It is a 3270 data stream workstation implementation responsibility to assign each auxiliary device a unique destination DOID for use in the Destination/Origin structured field. All values except X'0000' and X'FFFF' are available for use by the 3270 data stream workstation implementation.

The display is the default destination or origin if the data destination or origin is not explicitly identified by a Destination/Origin structured field.

Data of a type not supported that is directed to the display or an auxiliary device is rejected. For example, directing DDM data to the display or display data to the DDM auxiliary device causes the data to be rejected.

A Destination/Origin structured field can also be used in routing data to or from the primary display (DOID = X'0000'). It is invalid to use an inbound Destination/Origin structured field from the display (DOID=X'0000') unless the transmission also contains input from one or more auxiliary devices.

At the start of each outbound transmission the destination is the display (and at the start of each inbound transmission the origin is the display). Once a Destination/Origin structured field has established the destination or origin of the data, that destination or origin applies for all structured fields that follow until the end of the transmission unless changed by a subsequent Destination/Origin structured field.

---

## 11.3 Query Reply

The description for each Auxiliary Device Query Reply specifies whether it is sent in reply to either a Query or Query List or only in reply to a Query List.

Return of the Auxiliary Device Query Reply indicates a 3270 data stream workstation implementation; that is, support of the Destination/Origin structured field and one or more auxiliary devices. The Auxiliary Device Query Reply is returned in reply to either a Query List = Auxiliary Device (QCODE List =X'99') Equivalent or All or to a query.

The Query Reply for the individual auxiliary devices provides the DOID value to be used in the Destination/Origin structured field and any other required information concerning the auxiliary device.

A Query or Query List directed to an auxiliary device instead of the display is rejected unless it is specifically allowed by the device. See "Query Reply (IBM Auxiliary Device)" in topic 6.29 .

A separate Query Reply must be returned for each auxiliary device supported. For example, if two identical auxiliary devices were supported, a Query Reply would be returned for each. The DOID reported would be different for each.

When a 3270 data stream workstation supports an auxiliary device, the Query Reply for that device is returned regardless of whether the auxiliary device is available (for example, the necessary support code is not resident, or power is off).

---

## 11.4 Input Control

Some 3270 applications, particularly those not aware of auxiliary devices, cannot cope with unsolicited input from auxiliary devices. Therefore, the host application is given control over the time at which an auxiliary device is permitted to send in data. The control is achieved by the INCTRL (Input Control) flag in the Destination/Origin structured field. The INCTRL flag has meaning only outbound (to the auxiliary device) and is ignored on inbound. When the Destination/Origin structured field is directed to an auxiliary device, the INCTRL flag applies to that device. When the Destination/Origin structured field is directed to the display (DOID = X'0000'), the INCTRL flag applies to all (a global application) auxiliary devices supported. In other words, it provides global control.

The default (for example, POR) is input disabled. Once input is enabled for an auxiliary device, it remains enabled until disabled by any of the following:

- The auxiliary device receives a Destination/Origin structured field with INCTRL flag = B'10' (input disable).
- The display receives a Destination/Origin structured field with INCTRL = B'10' (global input disable).
- The workstation receives an EW or EWA with WCC = Reset.
- A Clear local function (for example, the Clear key is pressed).
- A power on reset.
- The workstation receives a Bind (SNA only).

Receiving a Destination/Origin structured field from the host application with INCTRL = B'01' causes no change in the input enabled/disabled state at the auxiliary device. Also, if the INCTRL flag value is the same as the existing input enable/disable states, the state is unchanged. For example, if the auxiliary device input enable/disable state is input enabled, receiving a Destination/Origin structured field with INCTRL = B'00' (input enable) is accepted and the input enable/disable state remains enabled.

**Note:** There is one exception during which an auxiliary device can send input without being enabled. An Exception Condition structured field, reporting unavailability of the auxiliary device, can be sent in reply to a Destination/Origin structured field sequence attempting to use the auxiliary device.

---

## 11.5 Auxiliary Device and Display Interaction

The auxiliary devices conform to the Read operations described in Chapter 3, "3270 Data Stream Commands" in topic 3.0 except where otherwise noted in this chapter.

When data is read in from an auxiliary device, the rules or states for read retry and read acknowledgment apply. For example, once a transmission is sent from an auxiliary device, data from that device cannot be sent inbound until a read acknowledgment is received. If the data from an auxiliary device is transmitted in multiple transmissions, each transmission requires an acknowledgment. An inbound transmission can contain data from the display or data from one or more auxiliary devices. When display data is sent in the same transmission as auxiliary device data, the Outbound 3270DS structured field must be used for the display data. An inbound transmission containing data from auxiliary devices must start with an AID of X'88' which indicates that structured fields follow. The same conditions that acknowledge a Query Reply also acknowledge inbound transmissions from an auxiliary device.

An outbound transmission to an auxiliary device constitutes a read acknowledgment according to the description for outbound display transmissions. The fact that the transmission is to an auxiliary device adds no additional acknowledgment function. For example, a transmission to an auxiliary device would acknowledge an outstanding Query Reply transmission because the transmission contained a WSF. Furthermore, in an SNA environment a transmission to an auxiliary device would constitute an acknowledgment to an outstanding enter transmission only if the transmission put the workstation in a send or contention state.

Only one display type read can occur in an outbound transmission. When in structured field form, the display type read must be the last structured field in the transmission. A display type read is defined as any of the following:

- A Query or Query List structured field
- A Read Partition structured field
- A Read Buffer, Read Modified, or Read Modified All command.

In an outbound transmission, data to an auxiliary device (for example, a DDM file) can initiate inbound data from the auxiliary device. Inbound data can be initiated from multiple auxiliary devices by a single outbound transmission containing multiple Destination/Origin structured fields. Inbound data from one or more auxiliary devices can be initiated in an outbound transmission which also contains a display type read. When this occurs, the display type read is performed first.

A display type read always takes priority over pending inbound data from an auxiliary device. A display operator enter action is considered a display type read. If inbound data is pending from one or more auxiliary devices, an operator enter action takes priority and uses the next available inbound transmission.

When the data from an auxiliary device must be sent in multiple transmissions (for example, a transmission size limit imposed for certain data), each inbound transmission is treated like an enter, to the extent that sending of the data is initiated by the device. A host read acknowledgment is required prior to sending the next part of the data. Therefore, data from an auxiliary device which is sent in multiple transmissions could have some interspersed display transmissions. Also, the display operator must not be locked out as a result of an auxiliary

device condition; for example, if the power is off, or if a diskette is removed.

---

## 11.6 Exception Handling

An exception condition on an auxiliary device must not cause termination of the host to 3270 data stream workstation session. That is, an auxiliary device exception condition must not cause a negative response. The exception conditions must be reported at the application level.

In general, the exception handling is defined by the data stream used by the auxiliary device. The DIA data stream documentation defines the exception heading for a DIA auxiliary device. The DDM data stream documentation defines that for a DDM auxiliary device.

There are some instances where the exception condition is handled within the 3270 data stream. For example, if the auxiliary device is not available (for example, if the power is off, or if the processing code is not resident), the unavailability is reported by returning a Destination/Origin structured field followed by an Exception/Status structured field with the code field = X'0801' (Resource Not Available). Another example might be if the host exceeds the transmission size specified in an Auxiliary Device Query Reply. In this case the code is X'084C' (permanent insufficient resource).

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 12.0 Chapter 12. Double-Byte Coded Character Set (DBCS)-Asia

Subtopics:

- 12.1 Introduction
- 12.2 Codepoints
- 12.3 DBCS Fields
- 12.4 DBCS Character Attribute (SA)
- 12.5 Shift Out (SO)/Shift In (SI)
- 12.6 Set Attribute (SA) Order and SO/SI Interaction
- 12.7 Orders

---

### 12.1 Introduction

This chapter describes the double-byte coded character set operation that is defined for use by the South East Asia Region (SEAR) countries, Japan, Korea, and Taiwan. The primary use is for Kanji.

This operation provides the capability for one- and 2-byte character sets.

Support of DBCS-Asia is indicated by the DBCS-Asia Query Reply and the Character Set Query Reply.

The operation of the DBCS-Asia character sets differ in many ways from the remainder of the 3270 data stream. However, when the DBCS-Asia implementation is operated exclusively in the 1-byte mode, the differences are limited to the following:

- Delete key operation
- The insert mode operation on an unformatted screen.

For the DBCS-Asia mode, the Delete key operates as follows:

1. If the cursor is located in a character location in an unprotected field, the Delete key deletes the character from the character location identified by the cursor and sets the MDT bit to 1. The cursor does not move.

2. All remaining characters, to the right of the cursor in the same field shift one column or two columns to the left, depending on whether a 1-byte character or a 2-byte character is deleted.
3. Vacated locations at the end of the field are filled with Nulls.

In an insert operation on an unformatted screen, text does not wrap at the end of the screen. Normal operation allows wrap at the end of the screen.

---

## 12.2 Codepoints

The following sections define the codepoints used by the DBCS-Asia operation.

Subtopics:

- [12.2.1 Graphic Codes](#)
- [12.2.2 Two-Byte Coded Field Selection Designators](#)
- [12.2.3 Two-Byte Coded Printer and Control Characters](#)

---

### 12.2.1 Graphic Codes

When the current character set attribute specifies a DBCS character set, the data stream values are interpreted as follows:

- X'00' through X'3F' and X'FF' are control codes.
- X'4040' is a space (blank).
- Combinations of X'41' thru X'FE' are language-dependent graphics.

A maximum of 1+190x190 graphics (X'4040' and combinations of X'41' thru X'FE') exist in DBCS-Asia, but there are fewer than 1+190x190 unique graphics in each language.

The integrity of these codepoints is maintained; that is, if read back, the same codepoints are transmitted as were received.

---

### 12.2.2 Two-Byte Coded Field Selection Designators

The following five 2-byte codepoints are defined as Field Selection designators for the 2-byte character set. The function is the same as for that of 1-byte character sets.

#### Codepoint Meaning

X'0000' Two-byte coded NULL  
X'4040' Two-byte coded Blank  
X'4250' Two-byte coded &  
X'426E' Two-byte coded >  
X'426F' Two-byte coded ?.

---

## 12.2.3 Two-Byte Coded Printer and Control Characters

When the current character set attribute specifies a DBCS character set, each character is represented by a 2-byte codepoint. The special characters for printer controls, namely NL, EM, FF, CR, DUP, and FM, are also represented by 2-byte codes as follows:

```
NL  X'0015'  
EM  X'0019'  
FF  X'000C'  
CR  X'000D'  
DUP X'001C'  
FM  X'001E'.
```

The first byte in each of the above codes is not interpreted as a null character but is included in the 2-byte coded character representation definition.

When the current character set attribute specifies a DBCS character set, a null character is defined as X'0000'.

This 2-byte coded null has the same function as two 1-byte coded nulls in Insert Mode, and it is suppressed for the read modified operation.

---

## 12.3 DBCS Fields

This section describes the support of a 2-byte field that consists of only DBCS characters.

Subtopics:

- [12.3.1 Character Set Attribute Type and Values](#)
- [12.3.2 Data Stream Processing](#)
- [12.3.3 Operator Interface](#)

---

### 12.3.1 Character Set Attribute Type and Values

The character set attribute is used with the SFE and the MF orders in the 3270 data stream to specify a character set for the field, as follows:

Type	Value
X'43'	Character Set

The following are valid settings for the character set attribute value byte:

Codepoint	Meaning
X'00'	Default character set
X'40' to X'EF'	Local ID for loadable character set
X'F0' to X'F7'	Local ID for nonloadable 1-byte character set
X'F8' to X'FE'	Local ID for nonloadable 2-byte character set.

Character set X'F8' is assigned to the DBCS character set. The sequence of SFE-X'43'-X'F8' defines a DBCS field.

---

## 12.3.2 Data Stream Processing

The graphic codepoints in the data stream reference the character set that is currently in effect.

The decision on whether a graphic codepoint is treated as a 1-byte or 2-byte character is determined by the "look left" rule. This means that in the buffer the application looks back towards the beginning of the buffer for the extended field attribute (formatted buffer). If the first buffer position (when looking left) is reached without encountering an SI or SO, the buffer is unformatted and the graphic codepoints are treated as if they refer to the default character set.

To achieve the desired results when using 2-byte fields, DBCS-Asia requires that the appropriate extended field attribute (character sets) be established before loading data into the field. If the application does not follow this procedure, transmissions may be rejected.

Subtopics:

- [12.3.2.1 Exception Conditions](#)
- [12.3.2.2 Display Buffer Manipulation](#)

---

### 12.3.2.1 Exception Conditions

If both bytes of a 2-byte graphic code are not in the range of X'40' through X'FE' the transmission is rejected with sense code 1003 or sense code 1001. Transmissions with invalid 2-byte representations of control codes are rejected with sense codes 1003 or 1001.

Two-byte codes cannot be separated into two 1-byte codes with an order inserted between the two halves. If this is done, the order is considered as the second half of a 2-byte code and the transmission is rejected with sense codes 1003 or 1001.

In a DBCS field, sending the sequence of a Null-Start Field or Null-Start Field Extended does not cause an exception condition. This sequence is used for filling the dead position. The *dead position* is a single-byte position that can occur at the end of a DBCS unprotected field and in which the operator cannot enter data.

If the end of processing of a W, EW, or EWA command or of an outbound 3270DS structured field (with W, EW, or EWA) occurs after receiving the first half of a 2-byte code, the transmission is rejected with either sense code 1005 or 1001.

A transmission that attempts to start writing a 2-byte character into a buffer position that is the second half of a 2-byte character is rejected with sense codes 1005 or 1001. One exception is that the second half of a 2-byte character can be replaced with a field attribute by an SF or SFE without error.

If a character attribute, type=character set, other than X'00' is received in the 2-byte field, the transmission is rejected.

If the 2-byte field contains a character attribute, type=character set, other than X'00', the data stream is rejected during post processing. *Post processing* is described on page [12.5.2](#). The character attribute, type=character set, is changed to X'00'.

---

### 12.3.2.2 Display Buffer Manipulation

As stated previously, DBCS-Asia assumes that an application properly establishes a field as 1-byte or 2-byte before attempting to enter data into the field. DBCS-Asia also assumes, for example, that an application does not change a field from 1-byte to 2-byte (or vice versa) without also updating the contents of the field in the same transmission. This section defines what happens when the application performs such buffer manipulations.

A transmission that only changes an extended field attribute from a 1-byte character set to a 2-byte character set changes how the codepoints are treated. That is, if the field character set is changed from 1-byte to 2-byte while leaving the field contents unchanged, the codepoints are treated as forming 2-byte characters and the character attribute, type=character set, is changed to default. Similarly, if the field character set is changed from 2-byte to 1-byte, the codepoints are treated as 1-byte characters.

Also, the way the codepoints in a 2-byte field are treated is changed by moving the field attribute (or adding a new attribute = 2-byte character set) within the field so that the attribute is an odd number of buffer positions from the attribute location. The codepoints are still treated as forming 2-byte characters but the pairing is shifted from the original pairing. For example, the new 2-byte characters are formed by pairing the second half of an old 2-byte character with the first half of the following old 2-byte character.

When a field contains control codes (codepoints below X'40'), manipulation of the field attribute as described above can result in leaving the buffer with invalid DBCS characters. When this occurs, the invalid characters are handled as follows:

- No error response occurs.
- Information displayed for the invalid DBCS is device dependent.
- On a read operation, an X'4040' is sent inbound for the invalid character.

Also, the manipulation of a field attribute can result in a single, non-null codepoint being left over at the end of a DBCS field; for example, a non-null in the dead position. A non-null in the dead position is handled as follows:

- No error response occurs.
- Information displayed for the non-null is device dependent.
- On a read buffer, the non-null dead position is sent inbound as an X'00' (null).
- On a read modified operation, the non-null dead position is treated as if it were a null and suppressed.

---

### 12.3.3 Operator Interface

The following sections describe the differences that the user is likely to notice when using DBCS-Asia.

Subtopics:

- [12.3.3.1 Fields that Span Rows](#)
- [12.3.3.2 Long and Short Cursors](#)
- [12.3.3.3 Using the Delete Key](#)
- [12.3.3.4 Entering Data with MSR](#)

---

#### 12.3.3.1 Fields that Span Rows

When a field spans rows, a DBCS character can span rows in the character buffer. That is, the first half of the character is in the rightmost column and the second half is in the leftmost column. One way of handling this situation is to have the left half of the DBCS character displayed in the rightmost column on the screen and the right half displayed in the leftmost column of the next row. However, an implementation can provide a different approach.

For example, if the device has an additional column on the screen (for example, eighty-first column for eighty-column displays), a DBCS character is displayed in the rightmost column and the additional (eighty-first) column on the screen. The leftmost column in the next row on the screen is reserved.

---

#### 12.3.3.2 Long and Short Cursors

There are two types of cursors used in DBCS-Asia: a cursor that is one character cell long (the *short cursor*) and a cursor that is two character cells long (the *long cursor*). When an operator positions a cursor in a DBCS field, the cursor is automatically two character cells long.

An operator cannot position a long cursor starting on the second half of a DBCS character. Any attempt by the operator to position a long cursor starting on the second half of a DBCS character causes the cursor to jump to the next position where the operator can enter data.

---

#### 12.3.3.3 Using the Delete Key

If the cursor is located in a character location in an unprotected field, the Delete key deletes the character from the character location identified by the cursor and sets the MDT bit to 1. The cursor does not move. All remaining characters to the right of the cursor in the same field shift one or two columns to the left depending on whether a 1-byte character or a DBCS character is deleted. Vacated locations at the end of the field are filled with Nulls.

---

### 12.3.3.4 Entering Data with MSR

Data entered into the DBCS field by the MSR is inhibited.

---

## 12.4 DBCS Character Attribute (SA)

This section describes the support of a DBCS character set character attribute. The interaction with the other methods of selecting a DBCS character set (DBCS field or Shift Out [SO]/Shift In [SI]) is described later in this chapter.

The SA order, with the character set attribute type X'43' and its attribute value X'F8', sets the current character set character attribute to a DBCS character set.

---

## 12.5 Shift Out (SO)/Shift In (SI)

This section defines the operation of the Shift Out (SO) and Shift In (SI) controls. SO = X'0E' and SI = X'0F'. The SO/SI controls do not have any effect in the data stream. However, the SO/SI controls are stored in the display buffer and do affect how the data in the buffer is interpreted, displayed, and read. The buffer locations from SO to SI are termed a DBCS subfield.

Support of SO/SI is reported in the DBCS-Asia Query Reply. The character set associated with the SO is the DBCS character set, which has a SET ID = X'80' and an LCID = X'F8'. The Set ID of this character set is reported in the DBCS-Asia Query Reply. The LCID and CGCSGID associated with the SET ID is reported in the Character Set Query Reply.

Subtopics:

- [12.5.1 Referencing the Character Set](#)
- [12.5.2 Exception Conditions For SO/SI](#)

---

### 12.5.1 Referencing the Character Set

The character set is referenced as follows:

- Formatted Screen:

- If there is no SO or SI in the buffer between the extended field attribute (EFA) and a subsequent buffer position N, the codepoint in buffer position N references the character set designated by the EFA.
  - In moving from a buffer position N left (back) to the EFA, if the first SO/SI encountered is an SI, the codepoint in buffer position N references the character set designated by the EFA.
  - In moving from a buffer position N left to the EFA, if the first SO/SI encountered is an SO, the codepoint in the buffer position N references the character set associated with SO, which is the DBCS character set.
- o Unformatted Screen:
    - If there is no SO or SI between the first buffer position and a subsequent buffer position N, the codepoint in buffer position N references the default character set.
    - In moving from buffer position N left to the first buffer position, if the first SO/SI encountered is an SI, the codepoint in buffer position N references the default character set.
    - In moving from buffer position N left to the first buffer position, if the first SO/SI encountered is an SO, the codepoint in buffer position N references the DBCS character set.

## 12.5.2 Exception Conditions For SO/SI

The following rules apply to the use of SO/SI:

- o SO/SI must appear as SO/SI pairs in the buffer (unformatted screen) or in a field (formatted screen). To be paired, the SO must precede the SI. For example, the following buffer sequence is invalid:
 

```
FA - - - - SI - - - -SO - - - FA
```
- o SO/SI controls do not have to be paired in the data stream. Unpaired SO/SI controls can occur in a buffer update.
- o SO/SI must not be sent to a DBCS field.
- o In a DBCS subfield, the buffer locations with a character attribute type of character set (= X'00') must be in contiguous pairs to accommodate the 2-byte characters.
- o In a DBCS subfield, the buffer contents with a character attribute type of character set (= other than X'00') are rejected.

There are some additional SO/SI exception considerations associated with the PT, RA, and EUA 3270 orders; see "Orders" in topic 12.7.

The validity of SO/SI use is verified as the data stream is processed, and again after processing the command or the Outbound 3270DS structured field. This second validity checked is termed post processing.

Subtopics:

- 12.5.2.1 Data Stream Processing
- 12.5.2.2 Post Processing

---

### 12.5.2.1 Data Stream Processing

A transmission is rejected and processing stopped immediately for any of the following SO/SI error conditions:

- An SO is received, and when looking left in the buffer an SO is encountered prior to encountering one of the following:
  - An SI
  - The first buffer position (unformatted buffer)
  - An FA or EFA (formatted buffer).
- An SI is received, and when looking left in the buffer an SO is not encountered before encountering one of the following:
  - An SI
  - The first buffer position (unformatted buffer)
  - An FA or EFA (formatted buffer).
- An SO or SI is received and the EFA for the field designates the DBCS character set.

---

### 12.5.2.2 Post Processing

The validity of SO/SI is checked again after processing a command or an Outbound 3270DS structured field.

A transmission is rejected during post processing if any of the following conditions exist in the display buffer:

- Looking left from an SO, another SO is encountered before encountering either an SI, the first buffer position (unformatted buffer) or an FA or EFA (formatted buffer).
- Looking left from an SI, either an SI, the first buffer position (unformatted buffer) or an FA or EFA is encountered prior to encountering an SO.
- An SO or SI exists in a DBCS field.
- Within a DBCS subfield, the positions with the character

attribute=field inherit are not contiguously paired.

- Within a DBCS subfield, the positions with the character attribute=other than X'00' are found.

Data integrity is not maintained for those errors detected during post processing since the buffer has already been altered.

---

## 12.6 Set Attribute (SA) Order and SO/SI Interaction

The SO/SI controls do not apply to buffer positions where an SA has set the character set character attribute of the codepoint to something other than default.

The character set character attribute set by an SA does not apply to SO and SI themselves, because they are control codes. However, the other character attributes (for example, color or highlighting) do apply because SO and SI take buffer positions (for example, NL or CR).

Subtopics:

- [12.6.1 SO/SI Creation by Operator](#)
- [12.6.2 Types of Fields](#)
- [12.6.3 Graphic Character Input](#)

---

### 12.6.1 SO/SI Creation by Operator

The application program has control over whether or not SO/SI can be created as a result of operator entry of data. The control is provided using the input control attribute (type = X'FE'). This attribute can only be set on a field basis by either the SFE or MF orders. The input control attribute format is as follows:

Attribute Type	Attribute Value
X'FE'	Input Control

The input control attribute value has a 1-byte value:

#### Content Meaning

X'00' SO/SI creation disabled  
X'01' SO/SI creation enabled.

The default (for example, power on) is SO/SI creation disabled. The input control attribute only controls the creation of SO/SI by the operator. This attribute has no effect on SO/SI use in the outbound or inbound data stream. For example, if the SO/SI creation is disabled for a field, the application can send SO/SI to the field and the SO/SI appears in a read of the field.

If the operator is entering 2-byte data, and the use of both SO/SI and a character attribute is valid, the SO/SI is used to identify the 2-byte data. For example, the SO/SI is used to identify the 2-byte data if the following are true:

- The operator selects the 2-byte character set associated with SO.
- The operator enters 2-byte data in an unprotected field. A field is unprotected if the extended field attribute is a character set other than that associated with SO.
- SO/SI creation is enabled. The SO/SI is used to identify the 2-byte data, even if the reply mode = character.

The character attribute is used to identify the 2-byte data if the SO/SI creation is disabled, and the reply mode = character. If the SO/SI creation is disabled and the reply mode = field or extended field, input is inhibited unless the operator is entering into a DBCS subfield that the application created.

## 12.6.2 Types of Fields

The following table summarizes how operator-entered DBCS characters are identified in the display buffer for different conditions.

It is assumed that an operator has selected the DBCS character set. The data is being entered into an unprotected field that contains no DBCS subfields.

Conditions			Two-Byte Data Identified Via			Input Inhibit
Field Character Set	Reply Mode	SO/SI Creation	SO/SI	CA	EFA	
Not the same as SO	Field or extended field	Disabled				X(1)
Not the same as SO	Character	Enabled	X			
Not the same as SO	Character	Disabled		X		
Not the same as SO	Character	Enabled	X			
Same as SO	Field or extended field	Enabled or disabled			X	
Same as SO	Character	Enabled or			X(2)	

disabled

(1) If the application creates DBCS subfields within the field, an operator can then enter DBCS data into the subfields. The entered DBCS data would be identified in the display buffer by the SO/SI delimiters.

(2) If the operator selects the DBCS character set, then X = the character attribute. If the operator selects field inherit, then X = the extended field attribute.

---

## 12.6.3 Graphic Character Input

This section describes the operation in a field with a 1-byte character set and with the SO/SI creation set on.

An operator cannot position a short cursor on the second half of a DBCS character nor a long cursor starting on the second half of a DBCS character. Any attempt by the operator to position a short cursor on the second half of a DBCS character or a long cursor starting on the second half of a DBCS character causes the cursor to jump to the next position where the operator can enter data.

Subtopics:

- [12.6.3.1 Outside DBCS Subfield](#)
- [12.6.3.2 Inside DBCS Subfield](#)
- [12.6.3.3 Delete Key](#)
- [12.6.3.4 Erase EOF Key](#)
- [12.6.3.5 Insert Mode](#)
- [12.6.3.6 Automatic Delete Operation](#)
- [12.6.3.7 MSR](#)

---

### 12.6.3.1 Outside DBCS Subfield

If the operator selects the DBCS character set and places a character into a position outside a DBCS subfield, the sequence created is as follows:

SO, the character, SI (if required to maintain the SO/SI pairing)

The sequence starts with the current cursor position, and after the operation, the cursor is positioned following the entered character.

For example, the buffer contains:

FA	C1/C2	C3/C4	C5	C6	FA
----	-------	-------	----	----	----

Where:

FA = Field Attribute

Cx/Cy = DBCS characters  
Cx = 1-byte characters.

The operator selects the DBCS character set, positions the cursor under C2 and C3, and enters C7/C8. The buffer becomes:

FA	C1	SO	C7/C8	SI	C6	FA
----	----	----	-------	----	----	----

The cursor appears under SI and C6.

---

### 12.6.3.2 Inside DBCS Subfield

When the operator selects the DBCS character set, the operation is as described previously. If the SI is overwritten, an SI is added following the character.

When the operator selects a character set other than the DBCS character set, the operation can be summarized as follows:

- The DBCS subfield is terminated with an SI.
- The character is entered after the SI. If the character is a 1-byte character, a space is automatically added in the next position and an SO is automatically added after the space, except when the character is entered into the last 2-byte character position of the DBCS subfield.
- If the character is entered into the last 2-byte position of the DBCS subfield, the SO would not be added.

For example, the buffer contains:

FA	SO	Ca/Cb	Cc/Cd	Ce/Cf	Cg/Ch	Ci/Cj	SI
----	----	-------	-------	-------	-------	-------	----

If the operator positions the short cursor under Cc (the first half of the DBCS character Cc/Cd) and enters a 1-byte character, C1, the buffer becomes:

FA	SO	Ca/Cb	SI	C1	sp	SO	Cg/Ch	Ci/Cj	SI
----	----	-------	----	----	----	----	-------	-------	----

A space is automatically added, and the cursor appears under the space.

---

### 12.6.3.3 Delete Key

The operation of the Delete key on an SO or an SI itself is inhibited since the deletion of it changes the characteristics of the subsequent

characters. The operation of the Delete key on the sequence of SOSI or SISO is allowed.

---

### 12.6.3.4 Erase EOF Key

When the cursor is at a DBCS character in a DBCS subfield, the operation of the Erase EOF key erases the character and all subsequent characters within the field but recreates SI in the cursor position.

---

### 12.6.3.5 Insert Mode

*Nulls in the Subfield:* When there are no DBCS subfields in an unprotected field, a character can be inserted if there are sufficient nulls in the field from (and including) the cursor position to the end of the field -- one null for a 1-byte character and four nulls for a DBCS character. If there are sufficient nulls and a character is inserted, the character formerly occupying the cursor location and all remaining characters within the field (except for nulls and characters to the right of nulls) are shifted one or two locations to the right.

When there is a DBCS subfield in an unprotected field, the insert mode operation is as described above except the nulls in the subfield cannot be used for the insert operation. That is, if there are not sufficient nulls from (and including) cursor location to the end of the field, other than nulls in the subfield, an insert operation is inhibited. When there are sufficient nulls outside the DBCS subfield and an insert operation occurs, the nulls in the subfield are shifted right in the same way as the characters in the subfield.

Therefore, to avoid possible operator problems the host application program should not create nulls in a DBCS subfield.

Such nulls are never created by operator action.

*Unformatted Screen:* The insert mode operation in the unformatted buffer does not wrap at the right-lower corner of the screen. This definition is unique to DBCS-Asia and it applies to a device which supports DBCS-Asia even when the host is operating the device in a 1-byte mode.

The Delete key and the Erase EOF key also work to the right lower corner of the screen.

---

### 12.6.3.6 Automatic Delete Operation

The following case shows the automatic delete operation.

The buffer contains:

FA	C1	C2	SO	C3/C4	C5/C6	SI	FA
----	----	----	----	-------	-------	----	----

If the operator puts the long cursor under C2 and S0 and enters Cx/Cy, the buffer becomes:

FA	C1	S0	Cx/Cy	C5/C6	SI	nu	FA
----	----	----	-------	-------	----	----	----

C4 (the second half of C3/C4) is deleted and the remainder of the DBCS subfield is shifted left one position. A null is added at the end of the field.

### 12.6.3.7 MSR

Data entered into the DBCS subfield by the MSR is inhibited.

## 12.7 Orders

The following table describes how 3270 orders are processed for DBCS:

Order (Abbreviation)	Operation
Start Field (SF)	This order cannot be used for defining DBCS fields. It operates the same for DBCS-Asia.
Set Buffer Address (SBA)	This order sets the current buffer address. The buffer address can be set for the second half of a DBCS character. However, if an attempt is made to write graphic data to this address, the data stream is rejected with a sense code of 1005 or 1001.
Insert Cursor (IC)	This order sets the cursor to the current buffer address. When the cursor address is set to the second half of a DBCS character, it is repositioned to the first half after processing the command or the Outbound 3270DS structured field, without informing the host.
Program Tab (PT)	This order operates the same for DBCS-Asia except that the padding function is done when the PT order follows SI. If the PT order follows SO the data stream is rejected with a sense code of 1003 or 1001.
Erase Unprotected to Address (EUA)	<p>An EUA order is not executed and is rejected for any of the following conditions:</p> <ul style="list-style-type: none"> <li>◦ The execution of the EUA would have resulted in an odd number of nulls in a DBCS field (except Dead Position) or DBCS subfield.</li> <li>◦ The start address is the second half of a DBCS character.</li> <li>◦ The start address is in a DBCS subfield and the stop address is in a following field.</li> </ul> <p>A single (unpaired) null must not be created by EUA in the DBCS field, except for the dead position.</p>

Start Field Extended (SFE)	This order is used for defining DBCS fields. Also, it is used to specify a new attribute type, input control. There is nothing unique for the SFE operation in DBCS-Asia.
Modified Field (MF)	This order is used to specify a new attribute type, Input Control. There is nothing unique for the MF operation in DBCS-Asia.
Repeat to Address (RA)	<p>This order repeats a character in the buffer from the current buffer address to the specified stop address (not inclusive).</p> <p>The character to be repeated can be a 1-byte or a DBCS character. The RA order sequence does not explicitly indicate whether the repeated character is 1-byte or DBCS. Therefore, the decision is made implicitly on the basis of how the first byte following the stop address would be treated if received in place of RA:</p> <ul style="list-style-type: none"> <li>◦ If the byte would have been treated as a 1-byte code, then the byte is repeated according to the RA order.</li> <li>◦ If the byte would have been treated as the first byte of a DBCS character, then the first 2 bytes following the stop address are repeated according to the RA order.</li> </ul> <p>If the current buffer address is set to the position for the second half of an existing DBCS character and the order is encountered in the data stream, the data stream is rejected with a sense code of 1005.</p> <p>When the order repeats a DBCS character, the number of positions to be repeated must be even. That is, the stop address minus current buffer address must be even. If not, the data stream is rejected with a sense code of 1001 or 1005.</p> <p>An RA order sequence with either SI or SO as the repeated character is invalid and is rejected with a sense code of 1003, 1001, or 1005.</p>

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## 13.0 Chapter 13. Local Format Storage

Subtopics:

- [13.1 Introduction](#)
- [13.2 Architecture Requirements](#)
- [13.3 Format Distribution](#)
- [13.4 Sample Sequences](#)
- [13.5 Format Selection](#)
- [13.6 User Selected Formats](#)

---

### 13.1 Introduction

Local Format Storage (LFS) provides for host-controlled loading of formats into the format storage of a device and for subsequent run-time presentation of the stored formats. The intent is to provide enhanced performance through decreased line transmission time and, in some cases, decreased host application turnaround time. Additionally, overall system performance should be improved by reducing contention for the host connection.

The use of this function can result in significant reduction of traffic on the host to control unit transmission facilities and in improved response time, because the display of a stored format is initiated by calling its name rather than transmitting the format data from the host.

This is accomplished by storing a 3270 data stream representation of the formatted presentation space. Format creation is the responsibility of the host application program. Each format sent to a device must be specifically tailored to that device and must have a unique name.

These formats can be presented on any device which supports the 3270 Data Stream, including protocol converters.

---

### 13.2 Architecture Requirements

The function of loading formats into a format storage utilizes the 3270 data stream Auxiliary Device architecture as defined in [Chapter 11, "Auxiliary Devices and Workstations" in topic 11.0](#). An Auxiliary Device does not necessarily have to be an actual device. For example, it can be

a file function or, as in this case, a format storage function.

In addition to the requirements shown below, the device must be a display supporting EBASE level of function and the auxiliary device architecture (for example, the Destination/Origin structured field). See Appendix E, "Functions Required for Systems Application Architecture (SAA) Support" for a definition of EBASE functions.

Adding, replacing, or deleting formats requires support of the:

- Format Storage Auxiliary Device Query Reply
- Load Format Storage structured field.

The accessing of formats by the host application requires support of the:

- Format Presentation Query Reply
- Select Format Group structured field
- Present Absolute Format structured field
- Present Relative Format structured field.

---

## 13.3 Format Distribution

The format storage function is associated with the display returning the Format Storage Auxiliary Device Query Reply. The formats to be loaded are addressed only to that display.

The Destination/Origin structured field that precedes the Load Format Storage structured fields causes the formats to be routed to the format storage function rather than to the display itself. Formats can be added, replaced, or deleted; format groups can be deleted; and format storage can be reset at any time under control of the designated host.

If the F/T flag of the Format Storage Auxiliary Device Query Reply indicates only one format per transmission (or per Data Chain; see non-SNA considerations below), then the host application is limited to sending only one Load Format Storage structured field per transmission (or Data Chain). After sending the format using the Load Format Storage structured field, the host application cannot send another format until the device sends in an Exception/Status structured field (preceded by a Destination/Origin structured field with a DOID = Format Storage Auxiliary Device). The Exception/Status structured field can either contain a sense code of X'0000' (Acknowledge) or one of the exception sense codes (Reject).

If the F/T flag indicates that more than one format can be sent per transmission (or Data Chain), then the host application can send more than one Load Format Storage structured field per transmission (or Data Chain). After sending the transmission (or Data Chain), the host application cannot send more formats until the device sends an Exception/Status structured field indicating Acknowledgement or Rejection. If the transmission (or Data Chain) is rejected, the Exception/Status structured field must include the Failing Format self defining parameter to indicate the name of the first format encountered with an exception condition.

In the non-SNA environment, if the outbound transmission size is limited (for example, by the LIMOUT value imposed or by BSC link limitations to a size smaller than a format), support of the Data Chaining function is

required. This function allows a structured field to span transmission. Use of the Data Chaining function is limited to the non-SNA environment. The chaining function provided by SNA makes support of Data Chaining unnecessary in the SNA environment.

---

## 13.4 Sample Sequences

The following are some sample sequences.

For a case in which only one format per transmission (or Data Chain) is allowed:

- Host sends - WSF, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Load Format Storage structured field. In the SNA environment, the host must send the transmission with CD, EB.
- Device returns - AID 88, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Exception/Status structured field (may be either 'acknowledge' or 'reject').

Repeat the above until all the formats have been loaded.

For the same case as above but in which non-SNA Data Chaining is required because the format does not fit in one transmission:

- Host sends - WSF, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Data Chain SF (= Begin), and Load Format Storage structured field which is the structured field header and part of the format.
- Host sends - WSF, Data Chain SF (= continue) which is the next part of the format. Note that the SF header is not repeated.
- Host sends - WSF, Data Chain SF (= end) which is the remainder of the format.
- Device returns - AID 88, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Exception/Status structured field (may be either 'acknowledge' or 'reject').

Repeat the above until all the formats have been loaded.

In a case in which more than one format is allowed per transmission (or Data Chain):

- Host sends - WSF, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Load Format Storage SF, or Load Format Storage SF.
  - Device returns - AID 88, Destination/Origin SF (DOID = Format Storage Auxiliary Device), Exception/Status SF (may be either 'acknowledge' or 'reject').
-

## 13.5 Format Selection

Although the formats are loaded to the particular display address which indicates support in the Query Reply, the formats stored are available to all the displays on the same controller. Provision is made for the host to determine at any time the number of groups assigned, the number of formats loaded, the number of local names used, and the amount of storage space available.

Formats are stored in available storage using a directory structure in which each group identifies a directory of all the formats loaded with the specified group name. The Select Format Group structured field assigns a directory to be searched when subsequent Present Absolute Format or Present Relative Format structured fields are received from the host. The most recently selected group directory is the one searched for the format to be presented.

Another directory can be selected by issuing the Select Format Group structured field specifying a different name. If a request is made for a non-existent group name, an exception indication is returned, and the selected group is set to indicate 'no group selected'. There is no default group name, so an existing group name must be correctly specified using a Select Format Group structured field before the host can request that any stored formats be presented by Local Format Storage.

Only the most recently selected directory is searched for the specified format, so it is the sender's responsibility to assure that the proper group name has been specified and that the desired format was loaded with the specified group name.

The Present Absolute Format and Present Relative Format structured fields are then sent by the host to select the name of the format to be presented. The difference between them is that the Present Relative Format structured fields contain an offset value that is added to each address-dependent order in the format data stream.

The 3270 data stream can contain multiple Present Format structured fields, and each one is processed in its turn, with the resulting format being passed to outbound data stream processing for each iteration of the Present Format structured field in the data stream. By issuing Present Relative Format requests with different offset values, users can cause recurring portions of a format to be presented in different locations on the screen.

Additional 3270 data can be sent by the host to a terminal following the Present Absolute Format and Present Relative Format structured fields. This is supported by the host concatenating a 3270 data stream structured field to the Present Absolute Format or Present Relative Format structured field. When this occurs, the control unit sends the data, unaltered, to normal outboard data stream processing after presenting the specified format.

Both the Present Absolute Format and Present Relative Format structured fields are sent with a presentation command specifying either a 3270 data stream Write, Erase/Write Alternate command and they contain a 3270 data stream WCC byte. These are the command and WCC to be used to display the format on the screen.

Using the format name contained in this structured field, the currently selected group directory is searched for a stored format with the same name. When the specified stored format is found, the format presentation command (Write, Erase/Write, or Erase/Write Alternate) and WCC are extracted from the Present Format structured field and passed to outbound data stream processing as if the command had been received from the host.

**Note:** Since formats are processed as 3270 data stream outbound structured fields, screen size is not changed for the Erase/Write or Erase/Write Alternate commands contained within the formats. The host application can establish the screen size by sending an Erase/Write or Erase/Write Alternate command as standard 3270 data stream transmissions prior to the Present Format structured field. An Erase/Reset structured field can also be used, but this resets all Set Partition Characteristics self-defining parameters that are currently in effect.

Presentation of any stored format can be initiated by host command, and presentation of selected 'local' formats can be initiated by a terminal user without host intervention, provided that local formats have been defined with the Load Formats structured field, and local format selection has been enabled.

---

## 13.6 User Selected Formats

The stored format name is a combination of the group name and format name. The user therefore does not necessarily know the stored format name. To accommodate user selection of formats, the Load Format Storage structured field provides a means of optionally specifying a "local name" for formats. When specified as a user-selected format, depending on flags in the Set Partition Characteristics structured field, the local name is treated as either:

- A pseudonym for a group name and format name (Default)
- A local format name qualified by the currently selected group name.

A search is made for a format with the local name, or for a format with the local name that is a member of the current group, depending on parameters passed in the Set Partition Characteristics structured field. If a match is found, the format is displayed as though a Present Absolute Format structured field had been received from the host.

This function can be enabled in two different ways:

- As a customization option
- Through the data stream.

User-selected formats can be enabled from each host application with the Set Partition Characteristics structured field. If this method is selected, user-selected formats must be enabled for each logical session as required. Once enabled, user selection of formats remains enabled until one of the following events occurs:

- A Set Partition Characteristics structured field is received that resets user-selected formats.

- An Erase Write or Erase Write Alternate command is received with WCC = reset.
- An SNA Bind command is received.
- An Erase/Reset structured field is received.

**Note:** User selection of formats mode is not reset by the CLEAR key, SYS REQ key, TEST key, or device power on.

If this method is selected, the default user-selected formats function is enabled for all sessions on this host. This option specifies the default mode, and does not preclude user-selected formats from being disabled or enabled through the data stream.

The local name search is initiated when the user enters data on a clear, unformatted screen. The data is considered to be a user request for local format presentation. Up to eight characters entered by the user, starting at the upper left corner of the presentation space, are used as the local name.

No validation is performed on the local name entered by the user.

The format presentation command used by the control unit is either Erase/Write or Erase/Write Alternate. The control unit determines which command to use based on the format screen size flag set by the user in the Load Format Storage structured field that distributes the format to the control unit. The default WCC (X'02' - Keyboard Restore) is used. After displaying the format, the SNA and keyboard states are the same as they were before the user entered the request for local format presentation.

If the Bracket state is Between Brackets (BETB), the SNA state remains in 'send' state during the local call-up fetch and display operation, thereby causing normal outbound requests to be rejected with sense code X'081B' (Receiver in Transmit Mode). When the user presses the ENTER key, the SLU goes to in-bracket state. The SLU begins a bracket and starts sending data. If the host sends a Bid or Begin Bracket (BB) before the first chain element is received, the SLU rejects the Bid or BB with an X'081B' (Bracket Bid Reject).

If no match to a stored format local name is found, the data is sent to the host in the normal fashion.

Identification and detection of a locally displayed format, when it is read by the host, is a host application responsibility. New panels should be developed in accordance with SAA requirements. Conventions such as requiring that the panel title always appear on line 1 are part of this architecture. This would aid in determining the types of panels that are being sent to the host and allow them to be processed in a standard fashion.

The host application is also responsible to ensure that the keyboard is restored following a Clear AID, thus allowing the user to enter local format requests.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

## A.0 Appendix A. SNA Sense Codes

Table A-1 lists the SNA sense codes for 3270 data stream command, order, and structured field errors.

For non-SNA environments, all 3270 data stream errors are rejected with an Op-Check (OC). Invalid commands are rejected with Command Reject (CR).

Table A-1. SNA Sense Codes	
Command, Order, and Structured Field Errors	Sense Code (hex)
COMMANDS (valid for all commands)	
Invalid command code	1003/100
Data following read command or EAU	1003/100
ORDERS (valid for all orders)	
Invalid data stream order	1003/100
Incomplete parameter list or parameter missing	1005/100
Invalid codepoint to be repeated by RA	1003/100
Invalid address	1005/100
RA with GE and no data following	1005/100
Modify Field Order:	
Current buffer location does not contain field attribute character	1005/100
Invalid attribute type	1003/100
Invalid color or highlighting attribute value	1003/100
Unknown character set attribute value in range X'01' - X'FE'	0863
Character set attribute value = X'FF'	1003/100
Field actions attribute value is reserved	1003/100
Set Attribute Order:	
Invalid attribute type	1003/100
Invalid color or highlighting attribute value	1003/100
Unknown character set attribute value in range X'01' - X'FE'	0863
Character set attribute value = X'FF'	1003/100

Set Buffer Address Order: Address not in form specified for partition (12-, 14-bit)	1005/100
Invalid address flag in 12-, 14-bit mode (high order bits = B'10')	1005/100
Start Field Extended Order: Invalid attribute type	1003/100
Invalid color or highlighting attribute value	1003/100
Unknown character set attribute value in range X'01' - X'FE'	0863
Character set attribute value = X'FF'	1003/100
Field actions attribute value is reserved	1003/100
STRUCTURED FIELDS (valid for all structured fields) Invalid structured field type	1003/100
Missing structured field parameter	1005/100
Reserved field is not zero	1003/100
RCV state and all windows have zero extent	0883
PID is unknown partition	1005/100
Incorrect length	1005/100
Activate Partition Zero extent window	1005/100
Begin/End of File Received an End of File without a Begin of File	1003/100
Received a second Begin of File without a previous End of F	11003/100
Create Partition PID > maximum allowed	1005/100
Invalid UOM value	1003/100
Invalid A-MODE	1003/100
Specification error in viewport/window	1005/100
Base > supported number	1005/100
Insufficient resource	1005/100
CS=X'0002'-X'FFFE' and horizontal windowing not supported	1001
RS=X'0002'-X'FFFE' and vertical windowing not supported	1001
Data Chain Received a Data Chain structured field with GROUP parameter = Continue or End when a data chain does not exist.	1003/100
Received a Data Chain structured field with GROUP parameter = Begin when a data chain already exists.	1003/100
Received a transmission starting with any 3270 command except WSF or EW/EWA (with WCC = Reset) while a data chain exists.	1005/100
Destination/Origin	See Note 1

Destroy Partition PID > maximum allowed	1005/100
Erase/Reset	See Note 1
FMH-1 Invalid field value Incomplete header RH indicates FMH-1 present but no FMH appears	1008 1008 1008
Load Color Table Table exhausted prior to LCTDEF Attempted to modify preserved entry Table modified but not LCTID	1003/100 1005/100 1005/100
Load Line Type Unsupported UNITS value Table modified but not LTTID Invalid parameter length Unsupported parameters Insufficient storage	1003/100 1005/100 1005/100 1003/100 084C
Load Programmed Symbol Set Invalid data type Invalid LCID (values X'40' through X'EF' are not considered) CHAR not available RWS not available RWS is valid but not installed EXTN not supported P-length incorrect Flag bits 5-6 not zero Invalid LH or LW Invalid SUBSN Color resource not available Invalid ST.SUBSN Excess bits in data Compressed data terminator incorrectly specified PS resource not available ECHAR not provided ECHAR reached prior to exhaustion of data Insufficient storage SET not supported NW or NH value too large	1003/100 1003/100 1005/100 084C 084C 1005/100 1005/100 1003/100 1005/100 1003/100 1005/100 1003/100 084C 1005/100 1005/100 1005/100 084C 1005/100 1005/100 084C 1003/100 1005/100
Modify Partition PID > maximum allowed Invalid reserved bits Specification error in window/viewport Invalid windowing parameters	1005/100 1003/100 1005/100 1005/100
Object Control Spanning error Unsupported mode Invalid command code Maximum entity length exceeded Invalid parameter length	1005/100 1003/100 1005/100 1003/100 1005/100

Unsupported parameters	1003/100
Insufficient storage	084C
Insufficient data	1005/100
<hr/>	
Object Data	
Spanning error	1005/100
Unsupported mode	1003/100
Invalid command code	1005/100
Maximum entity length exceeded	1003/100
REP/APPEND = B'01'	1005/100
Entity (REP) length exceeded	1005/100
Specified entity does not exist	1005/100
Invalid parameter length	1005/100
Unsupported parameters	1003/100
Insufficient storage	084C
Insufficient data	1005/100
<hr/>	
Object Picture	
Spanning error	1005/100
Unsupported mode	1003/100
Invalid command code	1005/100
REP/APPEND = B'10'	1005/100
Segment storing not supported	1003/100
Maximum entity length exceeded	1003/100
REP/APPEND = B'01'	1003/100
Entity (REP) length exceeded	1005/100
Specified entity does not exist	1005/100
Invalid parameter length	1005/100
Unsupported parameters	1003/100
Insufficient storage	084C
Insufficient data	1005/100
<hr/>	
OEM Data	See Note 2
<hr/>	
Outbound Text Header	
PID > maximum allowed	1005/100
Invalid command	1003/100
Violation of PS integrity	082B
Invalid WCC bit	1003/100
Unsupported control in header	1003/100
Parameter error in header	1005/100
Inconsistent controls in header	1005/100
Graphic in header	1005/100
Cursor column > partition width	1005/100
Incorrect A-MODE	1005/100
<hr/>	
Outbound Type1 Text	
PID > maximum allowed	1005/100
Unsupported control	1003/100
Parameter error in control	1005/100
Incorrect A-MODE	1005/100
Insufficient resource	084C
<hr/>	
Outbound 3270 Data Stream	
Incorrect A-MODE	1005/100
Invalid CMD byte	1003/100
Violation of presentation space integrity	082B
Data following EAU partition command	1003/100

WCC print bit = 1, but this is not the last structured field	1001
Present Absolute/Relative Format	
Invalid format presentation command	1003/100
Group not selected	1009
No formats loaded	0868
Format not found	0869
Format offset out of range	1005/100
Format processing error	087A
Read Partition	
Invalid read type code	1003/100
Invalid Q-code	1003/100
PID not equal to X'FF' for Query	1005/100
Read Partition not last structured field in chain	1005/100
Device in retry state	0871
RU Chain containing Read Partition does not specify CD	0829
RU Chain containing Read Partition does specify EB	0829
Request Recovery Data	
Request Recovery Data not last structured field in chain	1005/100
Device in retry state	0871
RU Chain containing Request Recovery Data does not specify CD	0829
RU Chain containing Request Recovery Data does specify EB	0829
Reset Partition	See Note 1
Restart	See Note 1
Save/Restore Formats	
Received chain containing Save/Restore structured field = primary/save which does not specify CD	0829
Received chain containing Save Restore Format structured field = primary/save which specifies EB	0829
Save/Restore Format structured field not last structured field in chain	1005/100
SCS Data	See Note 1
Set Checkpoint Interval	See Note 1
Select Color Table	
Table exhausted prior to LCTDEF	1003/100
Attempted to modify preserved entry	1005/100
Table modified but not LCTID	1005/100
Select Format Group	See Note 1

Select IPDS Mode When in 3270 mode, receive a Select IPDS Mode structured field which is not immediately after a WSF or a WSF, Data Chain structured field (=begin)	ld 1003/100
Set MSR Control Invalid magnetic data or length specification Invalid encode character set or length specification Encode aborted by operator Not last structured field in a chain or chain sent with neither RQD or CD	1001 1005/100 0824 0843
Set Partition Characteristics For Viewport Outlining self-defining parameter: This parameter is not supported Incorrect length Value is greater than maximum allowed	1005/100 1005/100 1005/100
Set Printer Characteristics	See Note 1
Set Reply Mode Invalid reply mode Invalid attribute type in A-list Incorrect reply mode for partition type	1003/100 1003/100 1005/100
Set Window Origin Window spec. outside presentation space CW > 0, but horizontal windowing not supported	1005/100 1005/100

**Notes:**

- Where an entry in the table is shown without any sense codes (for example, Erase/Reset), the only applicable sense code conditions are those shown in the Structured Fields (All) section on page [A.0](#).

The following are included for clarification:

- The validity of the field attribute is not checked.
  - W, EW, EWA, and WSF commands without data are accepted without error but are treated as no-op.
- When the OEM Data structured field is used in conjunction with the Destination/Origin structured field (that is, for data to or from a direct accessed OEM auxiliary device), this table does *not* apply. In this situation, all exception conditions associated with the OEM Data structured field must be reported either through the OEM data stream or with the Exception/Status structured field. Use of the subsystem -rsp (or OP-Chk) is not allowed.



# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
 DATETIME = 06/26/92 11:20:04  
 BLDVERS = 1.2  
 TITLE = 3270 Data Stream Programmer's Reference  
 AUTHOR =  
 COPYR = © Copyright IBM Corp. 1981, 1992

## B.0 Appendix B. Reset Actions

The following tables (Reset Actions - Part 1 and Reset Actions - Part 2) summarize the reset actions performed as a result of the actions listed. Under each heading is a reference linked to a legend or note that describes the reset action. To view these references, move the cursor to the desired reference in the table and press Enter. Also, you can scroll to the end of Part 2 of the tables to view the appropriate reference.

Table B-1. Reset Actions - Part 1			
Action	Partitions	Inbound Reply Mode	INPID/INOP
Jump key	<u>6</u>	<u>1</u> <u>3</u>	<u>NC</u>
Clear key SSCP	<u>9</u>	<u>9</u>	<u>NC</u>
Clear key Unowned	<u>R</u>	<u>ID</u>	<u>NC</u>
Clear key LULU	<u>R</u>	<u>R</u>	<u>9</u>
System Request key SSCP (Unowned)	<u>9</u>	<u>13</u>	<u>NR</u>
System Request key SSCP (LULU)	<u>9</u>	<u>9</u>	<u>R</u>
System Request key Unowned	<u>R</u>	<u>R</u>	<u>NR</u>
Receipt of RU	<u>R</u>	<u>R</u>	<u>NR</u>
System Request key LULU	<u>R</u>	<u>R</u>	<u>NR</u>
Test key Enter	<u>R</u>	<u>R</u>	<u>NR</u>
Test key Exit	<u>9</u>	<u>9</u>	<u>NR</u>
WCC in EW/EWA	<u>R</u>	<u>R</u>	<u>R</u>
WCC Reset in 3270DS EW/EWA Only	<u>NC</u>	<u>12</u>	<u>16</u>
Reset Partition	<u>NC</u>	<u>12</u>	<u>16</u>

Power on	<u>R</u>	<u>13</u>	<u>NR</u>
Clear LULU	<u>13</u>	<u>13</u>	<u>13</u>
DACTLU LULU	<u>ID</u>	<u>ID</u>	<u>NR</u>
DACTLU SSCP	<u>ID</u>	<u>ID</u>	<u>NR</u>
ACTLU SSCP	<u>9</u>	<u>9</u>	<u>NR</u>
DACTLU Unowned	<u>ID</u>	<u>ID</u>	<u>NR</u>
ACTLU Unowned	<u>ID</u>	<u>ID</u>	<u>NR</u>
Unbind LULU <u>8</u>	<u>16</u>	<u>13</u>	<u>NR</u>
Unbind SSCP	<u>9</u>	<u>9</u>	<u>NR</u>
Bind SSCP	<u>9</u>	<u>9</u>	<u>R</u>
Bind Unowned	<u>R</u>	<u>R</u>	<u>R</u>
Set Inbound Reply Mode	<u>NC</u>	<u>2</u>	<u>NC</u>
Create Partition	<u>4</u>	<u>12</u>	<u>16</u>
Destroy Active Partition	<u>6</u>	<u>1</u>	<u>16</u>
Destroy Not Active Partition	<u>5</u>	<u>5</u>	<u>16</u>
Clear Partition key LULU	<u>NC</u>	<u>NC</u>	<u>9</u>
Host Acknowledgment	<u>NC</u>	<u>NC</u>	<u>R</u>
Erase/Reset	<u>R</u>	<u>R</u>	<u>R</u>

A sense code of 082B also causes resetting of all the functions listed in the table, except for PS LCID and PS CONTENT.

Action	Highlight Color Character Set		PS Content PS LCID
	Selection	Indicators	
Jump key	<u>NC</u>	<u>7</u>	<u>NC</u>
Clear key SSCP	<u>9</u>	<u>9</u>	<u>NC</u>
Clear key Unowned	<u>R</u>	<u>7</u>	<u>NC</u>
Clear key LULU	<u>ID</u>	<u>7</u>	<u>NC</u>
System Request key SSCP (Unowned)	<u>9</u>	<u>9</u>	<u>NC</u>

System Request key SSCP (LULU)	<u>9</u>	<u>9</u>	<u>NC</u>
System Request key Unowned	<u>R</u>	<u>7</u>	<u>NC</u>
Receipt of RU	<u>R</u>	<u>7</u>	<u>NC</u>
System Request key LULU	<u>ID</u>	<u>7</u>	<u>NC</u>
Test key Enter	<u>R</u>	<u>7</u>	<u>NR</u>
Test key Exit	<u>R</u>	<u>7</u>	<u>15</u>
WCC in EW/EWA	<u>ID</u>	<u>7</u>	<u>NC</u>
WCC Reset in 3270DS EW/EWA Only	<u>NC</u>	<u>11</u>	<u>NC</u>
Reset Partition	<u>NC</u>	<u>11</u>	<u>NC</u>
Power on	<u>NR</u>	<u>R</u>	<u>R</u>
Clear LULU <u>8</u>	<u>NC</u>	<u>10</u>	<u>10</u>
DACTLU LULU	<u>ID</u>	<u>ID</u>	<u>NC</u>
DACTLU SSCP	<u>ID</u>	<u>ID</u>	<u>NC</u>
ACTLU SSCP	<u>9</u>	<u>9</u>	<u>NC</u>
DACTLU Unowned	<u>ID</u>	<u>ID</u>	<u>NC</u>
ACTLU Unowned	<u>ID</u>	<u>ID</u>	<u>NC</u>
Unbind LULU (8)	<u>NC</u>	<u>7</u>	<u>NC</u>
Unbind SSCP	<u>9</u>	<u>9</u>	<u>NC</u>
Bind SSCP	<u>9</u>	<u>9</u>	<u>NC</u>
Bind Unowned	<u>R</u>	<u>R</u>	<u>NC</u>
Set Inbound Reply Mode	<u>NC</u>	<u>7</u>	<u>NC</u>
Create Partition	<u>NC</u>	<u>7</u>	<u>NC</u>
Destroy Active Partition	<u>ID</u>	<u>7</u>	<u>NC</u>
Destroy Not Active Partition	<u>NC</u>	<u>NC</u>	<u>NC</u>
Clear Partition key LULU	<u>NC</u>	<u>NC</u>	<u>NC</u>
Host Acknowledgment	<u>NC</u>	<u>NC</u>	<u>NC</u>
Erase/Reset	<u>ID</u>	<u>7</u>	<u>NC</u>

A sense code of 082B also causes resetting of all the functions listed in the table, except for PS LCID and PS CONTENT.

**Legend:**

**ID** = Implementation defined (effect visible on external interface).  
**NA** = Not Applicable.  
**NC** = No change.  
**NR** = Not relevant (will be changed by subsequent action).  
**R** = Reset (effect visible on external interface).

**Notes:**

1. The inbound reply mode is set to the mode of the newly activated partition.
2. The inbound reply mode is changed to the mode described in the structured field.
3. The alternate character set is changed to the mode of the newly activated partition.
4. Add the partition name to the list along with its attributes.
5. Delete the partition name from the list along with its attributes. Reset if it is the last partition.
6. Change the active partition to the next in the list. Reset if it is the last partition.
7. Display exactly those attribute selection indicators that are honored as a result of the inbound reply mode in the current partition. If the inbound reply mode is reset, no operator selection is displayed.
8. In the unowned state, the last application write is still displayed (that is, the operator may be able to add data and perform a local copy).
9. The state will already be set.
10. No change. SNA does not permit Clear in the FM layer.
11. Reset if the active partition; otherwise, make no change.
12. Reset the referenced partition only.
13. Change the inbound reply mode to allow all selections.
14. Reset when the test mode alters the programmed symbols.
15. Reset if directed to INPID.

16. Screen remains--operator interaction restricted to the active partition.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## C.0 Appendix C. 12-, 14-, and 16-Bit Addressing

This appendix describes the 12-, 14-, and 16-bit addressing that the SNA 3270 data stream allows. With 12-bit addressing, an address is created from 2 bytes of binary information. The 6 low-order bits of each byte are joined to provide a 12-bit address. The address specifies the buffer position, not the line and column position on the display surface. For example, on a 480-character display, the buffer addresses are 0 to 479. To specify a 12-bit buffer address of 160 (binary 000010100000), bits 2-7 of the first byte are set to 000010; bits 2-7 of the second byte are set to 100000:

xx000010	xx100000
----------	----------

The 12-bit binary value is a combination of the two 6-bit values of the first and second bytes.

With 14-bit addressing, an address is created from 2 bytes of binary information, but the 14 bits of the address are contiguous. For example, an address of 800 decimal (X'320') would be represented as follows:

xx000011	00100000
----------	----------

With 16-bit addressing, all bits in both bytes are used. For example, an address of 3100 decimal (X'C1C') would be represented as follows:

00001100	00011100
----------	----------

SNA products (displays) that support only 12-bit addressing ignore bits 0 and 1 of each address byte in outbound data streams. For inbound data streams, they should set these bits in accordance with [Figure C-1 in topic C.0](#), although the bits provide no function other than retaining compatibility with previously written applications.

When a product supports both 12- and 14-bit addressing, bits 0 and 1 of the first address byte are flag bits and have the following significance:

**xx Setting Meaning**

B'00' 14-bit binary address follows  
B'01' 12-bit coded address follows  
B'10' Reserved  
B'11' 12-bit coded address follows.

For inbound data streams that contain 12-bit addresses, the display generates the B'01' or B'11' setting, using Figure C-1 in topic C.0.

A partition may be defined (using the Create Partition structured field) to operate with either 16-bit addressing or 12- and 14-bit addressing. When 16-bit address mode is specified in Create Partition, bits 0 and 1 are part of the address, outbound buffer addresses are interpreted as 16-bit binary, and inbound addresses are generated as 16-bit binary. If no partitions are defined, 12- and 14-bit addressing is assumed.

The SNA 3270 data stream can be EBCDIC or ASCII. With ASCII, each of the allowable characters is a graphic symbol; that is, there are no unprintable characters in the data stream. In addition, only 12-bit addressing is used with ASCII. Figure C-1 shows the 64 binary values permitted, using bits 2 through 7, and defines how they are transformed into ASCII values. Field attributes and write control characters also require special processing for conversion from binary values to graphic symbols when ASCII or 12-bit addressing is used.

Bits 2-7	EBCDIC	ASCII
00 0000	40	20
00 0001	C1	41
00 0010	C2	42
00 0011	C3	43
00 0100	C4	44
00 0101	C5	45
00 0110	C6	46
00 0111	C7	47
00 1000	C8	48
00 1001	C9	49
00 1010	4A	5B
00 1011	4B	2E
00 1100	4C	3C
00 1101	4D	28
00 1110	4E	2B
00 1111	4F	21
01 0000	50	26

01 0001	D1	4A
01 0010	D2	4B
01 0011	D3	4C
01 0100	D4	4D
01 0101	D5	4E
01 0110	D6	4F
01 0111	D7	50
01 1000	D8	51
01 1001	D9	52
01 1010	5A	5D
01 1011	5B	24
01 1100	5C	2A
01 1101	5D	29
01 1110	5E	3B
01 1111	5F	5E

<b>Bits 2-7</b>	<b>EBCDIC</b>	<b>ASCII</b>
10 0000	60	2D
10 0001	61	2F
10 0010	E2	53
10 0011	E3	54
10 0100	E4	55
10 0101	E5	56
10 0110	E6	57
10 0111	E7	58
10 1000	E8	59
10 1001	E9	5A
10 1010	6A	7C
10 1011	6B	2C
10 1100	6C	25
10 1101	6D	5F
10 1110	6E	3E

10 1111	6F	3F
11 0000	F0	30
11 0001	F1	31
11 0010	F2	32
11 0011	F3	33
11 0100	F4	34
11 0101	F5	35
11 0110	F6	36
11 0111	F7	37
11 1000	F8	38
11 1001	F9	39
11 1010	7A	3A
11 1011	7B	23
11 1100	7C	40
11 1101	7D	27
11 1110	7E	3D
11 1111	7F	22

Figure C-1. Conversion of Binary Values to Hexadecimal Values that Obtain Graphic Symbols

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## D.0 Appendix D. Special Applications

Subtopics:

- D.1 Introduction
- D.2 Query Reply (Anomaly Implementation)

---

### D.1 Introduction

This appendix contains information on how some implementations that do not conform to the 3270 Data Stream report those deviations to the host.

---

### D.2 Query Reply (Anomaly Implementation)

This section describes the Query Reply (Anomaly Implementation) structured field.

Subtopics:

- D.2.1 Function
- D.2.2 Format
- D.2.3 Additional Content Description
- D.2.4 3270 PC Application-to-Application

---

#### D.2.1 Function

The Query Reply (Anomaly Implementation) structured field indicates that an implementation of a function does not conform to the 3270 data stream operation for that function.

When the Anomaly Implementation Query Reply is supported, it is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List [=X'9D'], or All).

Although a host application must accept the Anomaly Implementation query

reply, the associated anomaly implementation may not be supported.

---

## D.2.2 Format

The following table shows the format of the Query Reply (Anomaly Implementation) structured field:

Byte	Content	Content Description
0-1	<b>L</b>	Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Query reply
3	<b>QCODE</b>	<b>X'9D'</b> Anomaly Implementation
4	<b>RES</b>	Reserved - must be zeros
5	<b>ANREF</b>	Anomaly reference number
6-7	<b>LIMIN</b>	Maximum bytes/inbound transmission
8-9	<b>LIMOUT</b>	Maximum bytes/outbound transmission
10	<b>LPARM</b>	Length in bytes of device dependent parameters (includes LPARM byte)
11-n	<b>DPARMS</b>	Device dependent parameters

---

## D.2.3 Additional Content Description

The content of the Query Reply (Anomaly Implementation) structured field is further described as follows:

**ANREF** Provides the reference number assigned to the anomaly implementation. Refer to the associated product documentation for the description of the anomaly implementation.

**LIMIN** Indicates the maximum number of bytes of Anomaly Auxiliary Device information the host application can expect in an inbound transmission.

The sum of bytes contained in all the structured fields associated with the Anomaly Auxiliary Device following an AID X'88' will be equal to or less than the value specified (in hexadecimal) in LIMIN.

**LIMOUT** Indicates the maximum number of bytes of Anomaly Auxiliary Device information the host application is allowed to send in an outbound transmission.

The sum of bytes contained in all the structured fields associated with an Anomaly Auxiliary Device, following a WSF command, must be equal to or less than the LIMOUT value (in hexadecimal). If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been

processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound data to the Anomaly Auxiliary Device.

**LPARM** Gives the number of bytes of device-dependent parameters + the LPARM byte.

**DPARMS** Provides the device-dependent information associated with the anomaly implementation.

---

## D.2.4 3270 PC Application-to-Application

This version of the Anomaly Implementation Query Reply is approved for use by Release 2.0 of the 3270 PC.

The following table shows the format of this version of the Anomaly Implementation Query Reply:

Byte	Content	Content Description
0-1	<b>L</b>	<b>X'0019'</b> Length of this structured field
2	<b>SFID</b>	<b>X'81'</b> Query Reply
3	<b>QCODE</b>	<b>X'9D'</b> Anomaly Implementation
4	<b>RES</b>	Reserved - must be zeros
5	<b>ANREF</b>	<b>X'01'</b> Anomaly reference number for a 3270 PC Application-to-Application
6-7	<b>LIMIN</b>	Maximum bytes of 3270 PC application data that will be sent in an inbound transmission
8-9	<b>LIMOUT</b>	Maximum bytes of 3270 PC application data allowed in an outbound transmission
10	<b>LPARM</b>	<b>X'0F'</b> Length in bytes of device-dependent parameters (includes LPARM byte)
11-12	<b>DOID</b>	Destination/Origin ID
13-24	<b>APLNME</b>	Application name

The content of this version of the Anomaly Implementation Query Reply is further described as follows:

**ANREF** 3270 PC Application/Application. Refer to the *IBM 3270 PC Control Program Programming Guide* for a description of the anomaly implementation.

**LIMIN** The sum of bytes contained in all the structured fields associated with the Anomaly Auxiliary Device, following an AID X'88', is equal to or less than the value specified (in hexadecimal) in LIMIN.

**LIMOUT** The sum of bytes contained in all the structured fields associated with an Anomaly Auxiliary Device following a WSF command must be equal to or less than the LIMOUT value (in hexadecimal). If this limit is exceeded, the transmission is rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound data to the Anomaly Auxiliary Device.

**DOID** The DOID is the 2-byte ID provided by the 3270 PC for use in the Destination/Origin structured field used in routing data to or from the 3270 PC application.

**APLNME** The APLNME is the 12-byte EBCDIC name assigned to the application by the 3270 PC.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## E.0 Appendix E. Functions Required for Systems Application Architecture (SAA) Support

Subtopics:

- [E.1 Introduction](#)
- [E.2 Query Replies](#)
- [E.3 Structured Fields](#)
- [E.4 Basic 3270 Commands](#)
- [E.5 Basic 3270 Orders](#)
- [E.6 3270 Format Control Orders](#)

---

### E.1 Introduction

This appendix lists the functions that the 3270 data stream must support for SAA. These functions are required for Extended Function Base Support (EBASE) in the 3270 data stream. Each function is listed in this appendix with a pointer to the specific sections of this book that explain that function.

To read about SAA, refer to *Systems Application Architecture: An Overview*. Other SAA publication titles are provided in the list of related publications in the preface of this book.

---

### E.2 Query Replies

The following Query Replies are necessary for SAA support. See [Chapter 6, "Inbound Structured Fields,"](#) for information about these Query Replies:

Query Reply (Character Sets)  
Query Reply (Device Characteristics) (LU 1 printers only)  
Query Reply (Implicit Partition)  
Query Reply (Null)  
Query Reply (SCS Data) (LU 1 printers only)  
Query Reply (Summary)  
Query Reply (Usable Area).

---

## E.3 Structured Fields

The following structured fields are necessary for SAA support. See Chapter 5, "Outbound/Inbound and Outbound Structured Fields," for information about these structured fields:

Read Partition  
Erase/Reset  
Outbound 3270DS.

---

## E.4 Basic 3270 Commands

The following 3270 commands necessary for SAA support. See Chapter 3, "3270 Data Stream Commands," for information about these commands:

Erase All Unprotected (EAU)  
Erase/Write (EW)  
Erase/Write Alternate (EWA)  
Read Buffer  
Read Modified  
Read Modified All  
Write (W)  
Write Structured Field (WSF).

**Note:** Read Buffer and Read Modified All are not supported by printers. The Read Modified command is supported only for retry of a query.

---

## E.5 Basic 3270 Orders

The following orders are those required for SAA support. See Chapter 4, "Character Sets, Orders, and Attributes," for an explanation of these orders:

Start Field  
Set Buffer Address  
Program Tab  
Insert Cursor  
Repeat to Address  
Erase Unprotected to Address.

---

## E.6 3270 Format Control Orders

The following table shows the format control orders that must be present for SAA support:

Orde	Meaning	EBCDIC	ASCII	Displayed as
NUL	NULL	X'00'		A space, suppressed on Read Modified

SUB	Substitute Character	X'3F'	-----	A solid circle
DUP	Duplicate	X'1C'	X'1C'	An overscore asterisk
FM	Field Mark	X'1E'	X'1E'	An overscore semicolon
FF	Form Feed	X'0C'	X'0C'	A space
CR	Carriage Return	X'0D'	X'0D'	A space
NL	New Line	X'15'	X'0A'	A space
EM	End of Medium	X'19'	X'19'	A space
EO	Eight Ones	X'FF'	-----	A space

NUL is read back as a null (X'00') on a Read Buffer operation but is not read back on Read Modified operations.

NL, EM, FF, and CR are printer control codes with no display function. However, the code must be supported to the extent of being accepted and, on reading back, must appear as NL, EM, FF, and CR, respectively. All are displayed as a space. EO is also displayed as a space.

FM and DUP are displayed as above. When read back, they appear as the FM and DUP codes.

FM and DUP can be entered from the keyboard. They are stored in the display buffer as controls; the current character set selection has no effect on them. They are transmitted to the application program as control codes.

The SUB local function, Error Override, entered from the keyboard, is required only as a part of Field Validation.

Some of these control codes provide a print format function when they are received by a printer. The control codes and their functions are as follows:

**NL** (New Line) moves the print position horizontally to the left margin and vertically down to the next line.

**CR** (Carriage Return) moves the print position horizontally to the left margin.

**EM** (End of Medium) terminates the print operation.

**FF** (Form Feed) moves the print position to the top and left margin of the next page.

---

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## F.0 Appendix F. Hexadecimal Index

Subtopics:

- [F.1 Introduction](#)
- [F.2 Commands](#)
- [F.3 Orders](#)
- [F.4 Format Control Orders](#)
- [F.5 Structured Fields \(One-Byte ID\)](#)
- [F.6 Structured Fields \(Two-Byte ID\)](#)
- [F.7 SCS Control Codes](#)

---

### F.1 Introduction

This appendix indexes all commands, orders, format control orders, structured fields, and SCS control codes by hexadecimal value.

---

### F.2 Commands

Hex Value	Name	Page
6E	Read Modified All	<a href="#">3.6.2.5</a>
6F	Erase All Unprotected	<a href="#">3.5.5</a>
7E	Erase/Write Alternate	<a href="#">3.5.3</a>
F1	Write	<a href="#">3.5.1</a>
F2	Read Buffer	<a href="#">3.6.1.1</a>
F3	Write Structured Field	<a href="#">3.5.4</a>
F5	Erase/Write	<a href="#">3.5.2</a>
F6	Read Modified	<a href="#">3.6.2</a>

---

### F.3 Orders

Hex Value	Name	Page
05	Program Tab	<a href="#">4.3.7</a>
08	Graphic Escape	<a href="#">4.3.10</a>
11	Set Buffer Address	<a href="#">4.3.3</a>
12	Erase Unprotected To Address	<a href="#">4.3.9</a>
13	Insert Cursor	<a href="#">4.3.6</a>
1D	Start Field	<a href="#">4.3.1</a>
28	Set Attribute	<a href="#">4.3.4</a>
29	Start Field Extended	<a href="#">4.3.2</a>
2C	Modify Field	<a href="#">4.3.5</a>
3C	Repeat To Address	<a href="#">4.3.8</a>

---

## F.4 Format Control Orders

Hex Value	Name	Page
00	Null	<a href="#">4.3.11</a> (all)
0A	New Line	
0C	Form Feed	
0D	Carriage Return	
19	End Of Medium	
1C	Duplicate	
1E	Field Mark	
3F	Substitute	
FF	Eight Ones	

---

## F.5 Structured Fields (One-Byte ID)

Hex Value	Name	Page
00	Reset Partition	<a href="#">5.21</a>
01	Read Partition	<a href="#">5.19</a>

03	Erase/Reset	<u>5.9</u>
06	Load Programmed Symbols	<u>5.13</u>
09	Set Reply Mode	<u>5.30</u>
0B	Set Window Origin	<u>5.31</u>
0C	Create Partition	<u>5.7</u>
0D	Destroy Partition	<u>5.8</u>
0E	Activate Partition	<u>5.5</u>
40	Outbound 3270DS	<u>5.16</u>
41	SCS Data	<u>5.23</u>
4A	Select Format Group	<u>5.25</u>
4B	Present Absolute Format	<u>5.17</u>
4C	Present Relative Format	<u>5.18</u>
80	Inbound 3270 DS	<u>6.4</u>

## F.6 Structured Fields (Two-Byte ID)

<b>Hex Value</b>	<b>Name</b>	<b>Page</b>
0F01	Set MSR Control	<u>5.27</u>
0F02	Destination/Origin	<u>5.35</u>
0F04	Select Color Table	<u>5.24</u>
0F05	Load Color Table	<u>5.10</u>
0F07	Load Line Type	<u>5.12</u>
0F08	Set Partition Characteristics	<u>5.28</u>
0F0A	Modify Partition	<u>5.14</u>
0F0F	Object Data	<u>5.37</u>
0F10	Object Picture	<u>5.38</u>
0F11	Object Control	<u>5.36</u>
0F1F	OEM Data	<u>5.39</u>
0F21	Data Chain	<u>5.34</u>
0F22	Exception/Status	<u>6.2</u>
0F24	Load Format Storage	<u>5.11</u>
0F71	Outbound Text Header	<u>5.15</u>
0F83	Select IPDS Mode	<u>5.41</u>

0F84	Set Printer Characteristics	<u>5.29</u>
0F85	Begin/End Of File	<u>5.6</u>
0FB1	Inbound Text Header	<u>6.3</u>
0FC1	Type 1 Text (Outbound)	<u>5.32</u>
0FC1	Type 1 Text (Inbound)	<u>6.6</u>
81nn	Query Reply	<u>6.8</u>
80	Summary	<u>6.48</u>
81	Usable Area	<u>6.51</u>
82	Image	<u>6.30</u>
83	Text Partitions	<u>6.49</u>
84	Alphanumeric Partitions	<u>6.9</u>
85	Character Sets	<u>6.12</u>
86	Color	<u>6.13</u>
87	Highlighting	<u>6.28</u>
88	Reply Modes	<u>6.42</u>
8A	Field Validation	<u>6.23</u>
8B	MSR Control	<u>6.34</u>
8C	Field Outlining	<u>6.22</u>
8E	Partition Characteristics	<u>6.38</u>
8F	OEM Auxiliary Device	<u>6.36</u>
90	Format Presentation	<u>6.24</u>
91	DBCS-Asia	<u>6.17</u>
92	Save/Restore Format	<u>6.44</u>
94	Format Storage Auxiliary Device	<u>6.25</u>
95	Distributed Data Management	<u>6.19</u>
96	Storage Pools	<u>6.47</u>
97	Document Interchange Arch.	<u>6.20</u>
98	Data Chaining	<u>6.15</u>
99	Auxiliary Device	<u>6.10</u>
9A	3270 IPDS	<u>6.52</u>
9C	Product Defined Data Stream	<u>6.41</u>
9D	Anomaly Implementation	<u>D.2</u>
9E	IBM Auxiliary Device	<u>6.29</u>
9F	Begin/End Of File	<u>6.11</u>

A0	Device Characteristics	<u>6.18</u>
A1	RPQ Names	<u>6.43</u>
A2	Data Streams	<u>6.16</u>
A6	Implicit Partition	<u>6.31</u>
A7	Paper Feed Techniques	<u>6.37</u>
A8	Transparency	<u>6.50</u>
A9	Settable Printer Chars.	<u>6.46</u>
AA	IOCA Auxiliary Device	<u>6.32</u>
AB	Cooperative Proc. Requestor	<u>6.14</u>
B0	Segment	<u>6.45</u>
B1	Procedure	<u>6.40</u>
B2	Line Type	<u>6.33</u>
B3	Port	<u>6.39</u>
B4	Graphic Color	<u>6.26</u>
B5	Extended Drawing Routine	<u>6.21</u>
B6	Graphic Symbol Sets	<u>6.27</u>
FF	Null	<u>6.35</u>
1030	Request Recovery Data	<u>5.20</u>
1031	Recovery Data	<u>6.5</u>
1032	Set Checkpoint Interval	<u>5.26</u>
1033	Restart	<u>5.22</u>
1034	Save/Restore Format	<u>5.40</u>

---

## F.7 SCS Control Codes

Hex Value	Name	Page
04	Vertical Channel Select (VCS)	<u>8.8</u> (all)
05	Horizontal Tab (HT)	
08	Graphic Escape (GE)	
0B	Vertical Tab (VT)	
0C	Form Feed (FF)	
0D	Carriage Return (CR)	
14	Enable Presentation (ENP)	

15	New Line (NL)	
16	Backspace (BS)	
1E	Interchange Record Separator (IRS)	
24	Inhibit Presentation (INP)	
25	Line Feed (LF)	
28	Set Attribute (SA)	
2B	Format (FMT)	
2BC1(L)(	)Set Horizontal Format (SHF)	
2BC2(L)(	)Set Vertical Format (SVF)	
2BC6(L)(	)Set Line Density (SLD)	
2BD1(L)8	(Set Text Orientation (STO)	
2BD2(L)2	(Set Print Density (SPD)	
2BD2(L)4	(Page Presentation Media (PPM)	
2BFE(L)3	(ASCII Transparent (ATRN)	
2F	Bell (BEL)	
35(L)(P)	Transparent (TRN)	

**Note:**

L = length  
P = parameters

# IBM BookManager Print Preview

DOCNUM = GA23-0059-07  
DATETIME = 06/26/92 11:20:04  
BLDVERS = 1.2  
TITLE = 3270 Data Stream Programmer's Reference  
AUTHOR =  
COPYR = © Copyright IBM Corp. 1981, 1992

---

## ABBREVIATIONS List of Abbreviations

---

### A

---

**ACK.** Acknowledge.

**ACTLU.** Activate Logical Unit.

**ACTPU.** Activate Physical Unit.

**AID.** Attention identifier.

**Alt.** Alternate.

**A/N.** Alphanumeric.

**APA.** All points addressable.

**API.** Application program interface.

**APL.** A Programming Language.

**ASCII.** American National Standard Code for Information Interchange.

---

## B

---

**BB.** Begin bracket.

**BETB.** Between-bracket state.

**BSC.** Binary synchronous communication.

---

## C

---

**CA.** Character attributes.

**CC.** (1) Control check. (2) Chain Command (flag).

**CCC.** Copy control character.

**CD.** Change direction.

**CE.** Channel-end.

**CGCSGID.** Coded graphic character set global identifier.

**char.** Character.

**CI.** Calling Indicator.

**cmd.** Command.

**cncl.** Cancel.

**cont.** Continuous.

**cpi.** Characters per inch.

**CUT.** Control unit terminal.

---

## D

---

**D.** Display.

**DACTLU.** Deactivate logical unit.

**DACTPU.** Deactivate physical unit.

**DBCS.** Double-byte character set.

**DE.** Device-end.

**DLE.** Data link escape.

**DOID.** Destination/origin ID.

---

## E

---

**EAU.** Erase All Unprotected.

**EB.** End bracket.

**EBCDIC.** Extended Binary-Coded Decimal Interchange Code.

**EM.** End of message.

**ENP.** Enable Presentation.

**EOF.** End of field.

**EOR.** End of Record.

**EPC.** Early Print Complete.

**ETX.** End of Text.

**EUA.** Erase Unprotected to Address.

---

## **F**

---

**FA.** Field attribute.

**FF.** Form Feed.

**FM.** (1) Field mark. (2) Function management.

**FMH.** Function management header.

---

## **G**

---

**GDDM.** Graphical Data Display Manager.

**GE.** Graphic escape.

---

## **H**

---

**hex.** Hexadecimal.

**HG.** Hardware group.

**HT.** Horizontal Tab.

---

## **I**

---

**I.** Information (format).

**IC.** Insert Cursor.

**ID.** (1) Identification. (2) Identifier.

**IML.** Initial microcode load.

**ind.** Indicator.

**INOP.** Inbound operation.

**INPID.** Inbound partition identity.

**INS.** Insert.

**I/O.** Input/output.

**IPDS.** Intelligent Printer Data Stream.

**IRS.** Interrecord-separator character.

---

## **K**

---

**k.** 1000.

**K.** 1024.

---

## **L**

---

**LCID.** Logical channel identifier.

**LF.** Line feed.

**LFS.** Local Format Storage.

**LH.** Link header.

**LRC.** Longitudinal redundancy check.

**LT.** Logical terminal.

**LU.** Logical unit.

**LUSTAT.** Logical unit status.

---

## **M**

---

**max.** Maximum.

**MDT.** Modified data tag.

**MF.** Modify field.

**MHS.** Magnetic hand scanner.

**MPP.** Maximum print position.

**MSR.** Magnetic stripe reader.

---

## **N**

---

**NL.** New Line.

**NOP.** No operation.

**NTT.** Nippon Telephone and Telegraph.

**NUL.** Null.

**NUM.** Numeric.

---

# O

---

**OC.** Operation Check.

**OEM.** Original equipment manufacturer.

---

# P

---

**PA.** (1) Program access. (2) Program attention.

**PC.** Personal computer.

**PEL.** Picture element.

**PF.** Program function.

**PID.** Product-set ID.

**PLU.** Primary logical unit.

**POST.** Power-On Self Test.

**PS.** Programmed symbols.

**PSS.** (1) Programmable Store System.  
(2) Programmed symbol set.

**PT.** Program Tab.

---

# R

---

**RA.** Repeat to Address.

**RB.** Read Buffer.

**RCV.** Receive.

**Req.** Request.

**resp.** Response.

**RH.** Request/response header.

**RM.** Read Modified.

**RO.** Ring Out.

**RPQ.** Request for price quotation.

**R/R.** Request/response.

**RSP.** Response.

**RSS.** Reverse start sentinel.

**RU.** Request/response unit.

**RWS.** Read/write storage.

---

# S

---

**SA.** Selection addressing.

**SAA.** Systems Application Architecture.

**SBA.** Set Buffer Address.

**SCS.** SNA character string.

**SDLC.** Synchronous Data Link Control.

**sec.** Second.

**SF.** Start field.

**SFE.** Start Field Extended.

**SHF.** Set Horizontal Format.

**SI.** Suppress Index.

**SLU.** Secondary logical unit.

**SNA.** Systems Network Architecture.

**SOH.** Start-of-heading character.

**SOR.** Start of record.

**SP.** Specific Poll.

**SPC.** Set Printer Characteristics.

**SPD.** Selector pen detect.

**SS.** Surge suppressor.

**SSCP.** System services control point.

**STX.** Start of text.

**SUB.** Substitute.

**SVF.** Set Vertical Format.

---

## **T**

---

**TH.** Transmission header.

---

## **U**

---

**U.** Unprotected.

**UOM.** Unit of Measure.

**US.** (1) Unit specify (2) Unit separator.

---

## **V**

---

V. Volt.

VFC. Vertical forms control.

---

## W

---

WACK. Wait before transmit positive acknowledgement.

WCC. Write control character.

WSF. Write Structured Field.

---

# Glossary

---

## A

---

**acknowledgment.** The transmission, by a receiver, of acknowledge characters as an affirmative response to a sender.

**active.** (1) Able to communicate on the network. A token-ring network adapter is active if it is able to transmit and receive on the network. (2) Operational. (3) Pertaining to a node or device that is connected or is available for connection to another node or device. (4) Currently transmitting or receiving.

**active logical terminal (LT).** In MLT, the currently displayed logical terminal. Synonymous with *foreground logical terminal*. Contrast with *background logical terminal*.

**adapter.** (1) A general term for a device that provides some transitional function between two or more devices. (2) In a local area network, within a communicating device, a circuit card with its associated software that enables the device to communicate over the network.

**address.** (1) A value that identifies a register, a particular part of storage, a data source, or a data sink. The value is represented by one or more characters. (2) To refer to a device or an item of data by its address. (3) In word processing, the location, identified by an address code, of a specific section of the recording medium or storage. (4) The location in the storage of a computer where data is stored. (5) In data communication, the unique code assigned to each device or workstation connected to a network.

**alphanumeric field.** A field that may contain any alphabetic, numeric, or special characters.

**alternate character set.** A character set, located in the terminal, from which characters are obtained for display and printing by using the graphic escape character in the data stream.

**American National Standard Code for Information Interchange (ASCII).** The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphics characters.

**application.** (1) The use to which an information processing system is put, for example, a payroll application, an airline reservation application, or a network application. (2) A collection of software components used to perform specific types of work on a computer.

**application program.** (1) A program written for or by a user that applies to the user's work. Some application programs receive support and services from a special kind of application program called a network application program. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

**asynchronous.** (1) Pertaining to two or more processes that do not depend upon the occurrence of a specific event such as a common timing signal. (2) In Fiber Distributed Data Interface (FDDI) rings, a type of data traffic that does not need bounded access delay to the medium and guaranteed throughput.

**attaching device.** Any device that is physically connected to a network and can communicate over the network.

**attention (ATTN).** An occurrence external to an operation that could cause an interruption of the operation.

**attention field.** In the 3270 Information Display System, a detectable field in which the designator character is a null, a space, or an

ampersand.

**attention identifier (AID).** (1) A code in the inbound 3270 data stream that identifies the source or type of data that follows. (2) A character in a data stream indicating that the user has pressed a key, such as ENTER, that requests an action by the system.

**attention key.** A function key on terminals that, when pressed, causes an I/O interruption in the processing unit.

**attribute.** (1) A characteristic. (2) A terminal display language or transformation definition language (TDL) keyword that specifies a particular quality for the TDL object with which it is associated.

**attribute type.** In the 3270 data stream, a code that identifies the properties from which an associated set of attribute values can be selected. See also *extended color*.

**attribute value.** In the 3270 data stream, a code immediately following the attribute type that specifies a particular property from the set defined by the attribute type.

**audible alarm.** (1) An alarm that is sounded when designated events occur that require operator attention or intervention before system operation can continue. (2) A special feature that sounds a short, audible tone automatically when a character is entered from the keyboard into the next-to-last character position on the screen. The tone can also be sounded under program control.

**automatic skip (auto-skip).** After entry of a character into the last character position of an unprotected display field, automatic repositioning of the cursor from a protected and numeric field to the first character position of the next unprotected display field.

---

## B

---

**background logical terminal (LT).** In MLT, any logical terminal that is not currently displayed. Contrast with *active logical terminal (LT)*.

**base color.** The capability of displaying or printing all characters in a field, in one of four colors, on a color terminal by use of combinations of the field protection and the field intensify bits of the field attribute.

**binary synchronous communication (BSC).** A form of telecommunication line control that uses a standard set of transmission control characters and control character sequences, for binary synchronous transmission of binary-coded data between stations. Contrast with *synchronous data link control (SDLC)*.

**bind command.** A command used to start a session and to define the characteristics of that session. Contrast with *unbind command*.

**bracket.** In SNA, one or more chains of request units (RUs) and their responses, which are exchanged between two LU-LU half-sessions and represent a transaction between them. A bracket must be completed before another bracket can be started. Examples of brackets are data base inquiries/replies, update transactions, and remote job entry output sequences to workstations.

**buffer.** (1) A portion of storage used to hold input or output data temporarily. (2) A routine or storage used to compensate for a difference in data rate or time of occurrence of events, when transferring data from one device to another.

**buffer address.** The address of a location in the buffer.

**byte.** (1) A string that consists of a number of bits, treated as a unit, and representing a character. (2) A binary character operated upon as a unit and usually shorter than a computer word. (3) A string that consists of a particular number of bits, usually 8, that is treated as a unit, and that represents a character. (4) A group of 8 adjacent binary digits that represent one extended binary-coded decimal interchange code (EBCDIC) character.

---

## C

---

**card.** In the 3174 Establishment Controller, a unit of electronic circuitry contained in a plastic casing (or cassette) and providing the controller with a specialized function, for example, a Terminal Adapter or an Encrypt/Decrypt Adapter.

**channel.** (1) A functional unit, controlled by a host computer, that handles the transfer of data between processor storage and local peripheral equipment. (2) A path along which signals can be sent. (3) The portion of a storage medium that is accessible to a given reading or writing station. (4) In broadband transmission, a designation of a frequency band 6 MHz wide.

**channel-attached.** Pertaining to attachment of devices directly by data channels (I/O channels) to a computer. Synonym for *local*. Contrast with *telecommunication-attached*.

**character attribute.** The properties of a character with respect to its color, highlighting, and character set. See also *extended field attribute*.

**character buffer.** The read/write storage used by a partition for storing character or graphic data for display or printing on a terminal.

**character mode.** A mode in which input is treated as alphanumeric data, rather than graphic data.

**character position.** (1) A location on the screen at which one character can be displayed. (2) An addressed location in the buffer at which 1 character can be stored.

**character set.** (1) A defined collection of characters. (2) A group of characters used for a specific reason, for example, the set of characters a printer can print. (3) The collection of graphic characters required to support a specific language.

**code page.** An assignment of graphic characters and control function meanings to all code points.

**code point.** A 1-byte code representing one of 256 potential characters.

**command.** (1) A request for performance of an operation or execution of a program.  
(2) A character string from a source external to a system that represents a request for system action.

**command retry.** A channel and control unit procedure that causes a command to be retried without requiring an I/O interruption.

**component.** (1) Hardware or software that is part of a functional unit. (2) A functional part of an operating system, for example, the scheduler or supervisor. (3) In systems with VSAM, a named, cataloged collection of stored records, such as the data component or index component of a key-sequenced file or alternate index. (4) In System/38 graphics, the representation of a data group on a chart. (5) See *terminal component* and *solid state component*.

**configuration.** The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units. More specifically, the term *configuration* may refer to a hardware configuration or a software configuration. See also *system configuration*.

**control block.** (1) A storage area used by a computer program to hold control information. (2) In the IBM Token-Ring Network, a specifically formatted block of information provided from the application program to the Adapter Support Interface to request an operation.

**control character.** (1) A character whose occurrence in a particular context specifies a control function. (2) A character used to specify that a control unit is to perform a particular operation.

**control codes.** (1) Code points and their assigned control function meanings. (2) The hexadecimal values hex 00 through hex 3F, and hex FF in the 3270 data stream. ASCII control codes are the hexadecimal values hex 00 through hex 1F and 7F.

**control function.** Synonym for *control operation*.

**controller.** A unit that controls input/output operations for one or more devices.

**control operation.** An action that affects the recording, processing, transmission, or interpretation of data; for example, starting or stopping a process, carriage return, font change, rewind, and end of transmission. Synonymous with *control function*.

**control unit.** A general term for any device that provides common functions for other devices or mechanisms. Synonym for controller.

**control unit terminal (CUT).** A terminal that relies on the 3174 to interpret the data stream. Examples are the 3178, 3179, 3278 Model 2, and 3279 Model S2A.

**control unit terminal (CUT) mode.** A host-interactive mode that enables an IBM 3270 Personal Computer customized in this mode to run only one session emulating a 3178, 3179, 3278 Model 2, or 3279 Model S2A.

**conversion.** (1) In programming languages, the transformation between values that represent the same data item but belong to different data types. Information may be lost as a result of conversion because accuracy of data representation varies among different data types. (2) The process of changing from one method of data processing to another or from one data processing system to another. (3) The process of changing from one form of representation to another, for example, to change from decimal representation to binary representation.

**copy control character (CCC).** A character used in conjunction with the Copy command to specify the type of data to be copied.

**copy operation.** An operation that copies the contents of the buffer from one terminal to another terminal attached to the same control unit.

**create.** In 3174 central site customizing, to create a library member for a network controller, and store the customizing data for that library member on a Library diskette.

**cursor.** (1) A movable, visible mark used to indicate the position at which the next operation will occur on a display surface. (2) A unique symbol that identifies a character position in a screen display, usually the character position at which the next character to be entered from the keyboard will be displayed.

**customization.** Procedures that tailor the control unit microcode to fit the various types of display stations and printers and the method of host attachment that a particular control unit will handle.

---

## D

---

**data.** (1) A representation of facts, concepts, or instructions in a formalized manner suitable for communication, interpretation, or processing by human or automatic means. (2) Any representations such as characters or analog quantities to which meaning is or might be assigned.

**data chaining.** In synchronous data link control (SDLC) data transmission, the chaining together of scattered segments of storage data to assemble a complete SDLC frame.

**Data Entry keyboard.** A keyboard layout designed for data entry applications.

**data link.** (1) Any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communication control unit, or connects one communication control unit with another. (2) The assembly of parts of two data terminal equipment (DTE) devices that are controlled by a link protocol, and the interconnecting data circuit, that enable data to be transferred from a data source to a data sink. (3) In SNA, see also *link*.

**Note:** A telecommunication line is only the physical medium of transmission. A data link includes the physical medium of transmission, the protocol, and associated devices and programs; it is both physical and logical.

**data stream.** (1) All data transmitted through a data channel in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format. See also *data stream format*.

**data stream format.** In SNA, the format of the data elements (end-user data) in the request unit (RU). See also *3270 data stream* and *SNA character string (SCS)*.

**data transfer.** (1) The result of the transmission of data signals from any data source to a data receiver. (2) The movement, or copying, of data from one location and the storage of the data at another location.

**decode.** (1) To convert data by reversing the effect of some previous encoding. (2) To interpret a code. Contrast with *encode*.

**default.** Pertaining to an attribute, value, or option that is assumed when none is explicitly specified.

**default destination.** A destination for display stations and printers that is defined in AEA customization.

**designator character.** A character or space that immediately follows the field attribute character in a detectable field to denote either a selection field or an attention field. The designator character controls whether a detect on the field will or will not cause an attention. For a nonattention-producing field, the designator character also determines whether the modified data tag for the field is to be set or reset as a result of a selector-pen detect.

**destination.** Any point or location, such as a node, station, or a particular terminal, to which information is to be sent.

**detectable.** An attribute of a display field.

**device.** (1) A mechanical, electrical, or electronic contrivance with a specific purpose. (2) An input/output unit such as a terminal, display, or printer.

**diskette.** A flexible magnetic disk enclosed in a protective container.

**display field.** (1) An area in the display buffer that contains a set of characters that can be manipulated or operated upon as a unit. (2) A group of consecutive characters (in the buffer) that starts with an attribute character (defining the characteristics of the field) and contains one or more alphanumeric characters. The field continues to, but does not include, the next attribute character.

**display station.** An input/output device containing a display screen and an attached keyboard that allows a user to send information to or receive information from the system.

**dot.** One point in a printer or display block matrix.

**downstream.** (1) In the direction of data flow or toward the destination of transmission. (2) From the processor toward an attached unit or end user. (3) Contrast with *upstream*.

**downstream load (DSL).** The capability of a distributed function terminal to receive its control program from the control unit to which it is attached. A disk containing the terminal's control program is loaded into the control unit.

**duplex.** Pertaining to communication in which data can be sent and received at the same time.

---

## E

---

**EBCDIC.** Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

**emulation.** (1) The imitation of all or part of one system by another, primarily by hardware, so that the imitating system accepts the same data, executes the same programs, and achieves the same results as the imitated computer system. (2) The use of programming techniques and special machine features to permit a computing system to execute programs written for another system. (3) Imitation; for example, imitation of a computer or device. (4) See *terminal emulation*. (5) Contrast with *simulation*.

**enabled.** (1) On a LAN, pertaining to an adapter or device that is active, operational, and able to receive frames from the network. (2) Pertaining to a state of a processing unit that allows the occurrence of certain types of interruptions. (3) Pertaining to the state in which a transmission control unit or an audio response unit can accept incoming calls on a line.

**encode.** To convert data by the use of a code or a coded character set in such a manner that reversion to the original form is possible. *Encode* is sometimes loosely used when complete reversion is not possible. Contrast with *decode*.

**envelope.** (1) Information added to a frame or other message unit to allow it to be transmitted using a protocol other than the protocol in which the message unit originated. (2) To surround or enclose a message unit in information to allow the message unit to be transmitted using a protocol other than the protocol in which the message originated.

**Erase All Unprotected (EAU) command.** A 3270 data stream command that erases all unprotected fields and inserts nulls.

**Erase Unprotected to Address (EUA) order.** A data stream order that erases all unprotected character positions (inserts nulls) from the current buffer address up to, but not including, the specified stop address.

**event.** (1) An occurrence or happening. (2) An occurrence of significance to a task; for example the completion of an asynchronous operation, such as an input/output operation.

**extended binary-coded decimal interchange code (EBCDIC).** A coded character set consisting of 8-bit coded characters.

**extended color.** (1) A capability that allows color terminals to display or print fields or characters in colors using extended field and character attributes. (2) An attribute type in the extended field attribute and character attribute.

**extended field attribute.** Additional field definition to the field attribute that controls defining additional properties; for example, color, highlighting, character set, and field validation. The extended field attribute is altered by information passed in the Start Field Extended and Modify Field orders.

**extended highlighting.** (1) A function that provides blink, reverse video, and underscore for emphasizing fields or characters on devices supporting extended field attributes and character attributes. (2) An attribute type in the extended field attribute and character attribute. (3) An attribute passed between session partners in the Start Field Extended, Modify Field, and Set Attribute orders.

**feature.** A part of an IBM product that may be ordered separately by the customer.

**field.** See *display field*.

**field attribute.** A control character stored in the character buffer in the first character position of a field. For those devices supporting the 3270 data stream, a field attribute defines protected/unprotected, alphanumeric/numeric, detectable/nondetectable, display/nondisplay, intensity, and modified data tag (MDT).

**field inherit.** A bit setting in the character attribute that defaults the character properties to the extended field attributes or device default if the buffer is unformatted.

**file.** A named set of records stored or processed as a unit.

**flag.** (1) An indicator or parameter that shows the setting of a switch. (2) Any of various types of indicators used for identification, for example, a wordmark. (3) A character that signals the occurrence of some condition, such as the end of a word. (4) Deprecated term for *mark*.

**foreground logical terminal (LT).** Synonym for *active logical terminal (LT)*.

**formatted display.** A display screen in which the attributes of one or more display fields have been defined by the user. Contrast with *unformatted display*.

**frame.** (1) The unit of transmission in some LANs, including the IBM Token-Ring Network and the IBM PC Network. It includes delimiters, control characters, information, and checking characters. On a token-ring network, a frame is created from a token when the token has data appended to it. On a token bus network (IBM PC Network), all frames including the token frame contain a preamble, start delimiter, control address, optional data and checking characters, end delimiter, and are followed by a minimum silence period. (2) A housing for machine elements. (3) In synchronous data link control (SDLC), the vehicle for every command, every response, and all information that is transmitted using SDLC procedures. Each frame begins and ends with a flag.

**full duplex.** Synonym for *duplex*.

**function.** In NetView DM, a function is the specification of a transmission activity on a resource or group of resources. Functions are

grouped into phases. In CSCM, resources are known as data objects.

---

## G

---

**generate.** In 3174 central site customizing, to write a Control diskette containing the customizing data for a particular controller. Also, to print a mailing address label and a diskette label for a particular controller.

**graphic escape.** In the 3270 data stream, a control code used to introduce a graphic character (hex 40 through hex FE) from an alternate character set.

**group.** (1) A set of related records that have the same value for a particular field in all records. (2) A collection of users who can share access authorities for protected resources. (3) A list of names that are known together by a single name.

---

## H

---

**hexadecimal.** (1) Pertaining to a selection, choice, or condition that has 16 possible values or states. (2) Pertaining to a fixed-radix numeration system, with radix of 16. (3) Pertaining to a numbering system with base of 16; valid numbers use the digits 0 through 9 and characters A through F, where A represents 10 and F represents 15.

**hexadecimal number.** The 1-byte hexadecimal equivalent of an EBCDIC character.

**host application program.** An application program processed in the host computer.

**host attachment.** A mode of SNA communication in which the processor acts as a secondary SNA device.

**host interface.** Interface between a network and the host computer.

**host logical unit (LU).** An SNA logical unit (LU) located in a host processor, for example, an ACF/VTAM application program.

**host system.** (1) A data processing system used to prepare programs and operating environments for use on another computer or controller. (2) The data processing system to which a network is connected and with which the system can communicate. (3) The controlling or highest-level system in a data communication configuration; for example, a System/38 is the host system for the workstations connected to it.

---

## I

---

**IBM Cabling System.** A permanently installed wiring system that eliminates the need to rewire when terminals are moved from one location to another within an office complex. It allows transmission of data at very high speeds and is the foundation for installing a local area network.

**initial microcode load (IML).** The action of loading the operational microcode.

**input device.** A device in a data processing system by which data may be entered into the system.

**input/output (I/O).** (1) Pertaining to a device whose parts can perform an input process and an output process at the same time. (2) Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process. (3) Pertaining to input, output, or both.

**Insert Cursor (IC) order.** An order that displays the cursor at the current buffer address.

**intensified display.** An attribute of a display field; causes data in that field to be displayed at a brighter level than other data displayed on the screen.

**interface.** (1) A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, and other characteristics as appropriate. (2) A shared boundary. An interface may be a hardware component to link two devices or a portion of storage or registers accessed by two or more computer programs. (3) Hardware, software, or both, that links systems, programs, or devices.

---

## K

---

**keyboard definition.** A customizing procedure for defining a maximum of four modified keyboard layouts for modifiable keyboards only. Most characters, symbols, and functions can be relocated, duplicated, or deleted from almost any keyboard position. Synonym for *modify keyboard*.

---

## L

---

**layer.** (1) One of the seven levels of the Open Systems Interconnection reference model. (2) In open systems architecture, a collection of related functions that comprise one level of hierarchy of functions. Each layer specifies its own functions and assumes that lower level functions are provided.

(3) In SNA, a grouping of related functions that are logically separate from the functions of other layers. Implementation of the functions in one layer can be changed without affecting functions in other layers.

**light pen.** A light-sensitive pick device that is used by pointing it at the display surface.

**line control characters.** Characters that regulate the transmission of data over a line; for example, delimiting messages, checking for transmission errors, and indicating whether a station has data to send or is ready to receive data.

**link.** (1) The logical connection between nodes including the end-to-end link control procedures. (2) The combination of physical media, protocols, and programming that connects devices on a network. (3) In computer programming, the part of a program, in some cases a single instruction or an address, that passes control and parameters between separate portions of the computer program. (4) To interconnect items of data or portions of one or more computer programs. (5) In SNA, the combination of the link connection and link stations joining network nodes.

**local.** Pertaining to a device accessed directly without use of a telecommunication line. Synonym for *channel-attached*. Contrast with *remote*.

**local format storage.** In 3174, this function allows the controller to store pre-defined formatted screens and subsequently be viewed by a terminal user. The formatted screens are downloaded from the host into the 3174.

**location.** With reference to a 3174, a place within the 3174 chassis where a particular card or adapter is inserted.

**logical terminal (LT).** In MLT, one of five sessions available to share one display station.

**logical unit (LU).** In SNA, a port through which an end user accesses the SNA network in order to communicate with another end user and through which the end user accesses the functions provided by system services control points (SSCPs). An LU can support at least two sessions, one with an SSCP and one with another LU, and may be capable of supporting many sessions with other logical units.

---

## M

---

**main storage.** Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent processing.

**mark.** A symbol or symbols that indicate the beginning or the end of a field, a word, an item of data or a set of data such as a file, record, or block.

**medium.** A physical carrier of electrical or optical energy.

**memory.** Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing. Synonymous with *main storage*.

**modified data tag (MDT).** A bit in the attribute character of a display field that, when set, causes that field to be transferred to the channel during a read-modified operation. The modified data tag may be set by a keyboard input to the field, a selector-pen detection in the field, a card read-in operation, or program control. The modified data tag may be reset by a selector-pen detection in the field, program control, or ERASE INPUT key.

**Modify Field (MF) order.** An order that allows specified field and

extended attributes to be modified, without having to respecify all of the attributes of the field.

**modify keyboard..** Synonym for *keyboard definition*.

**multidrop (network).** A network configuration in which there are one or more intermediate nodes on the path between a central node and an endpoint node.

---

## N

---

**name.** An alphanumeric term that identifies a data set, statement, program, or cataloged procedure.

**network.** (1) A configuration of data processing devices and software connected for information interchange. (2) An arrangement of nodes and connecting branches. Connections are made between data stations.

**node.** (1) Any device, attached to a network, that transmits and/or receives data. (2) An endpoint of a link, or a junction common to two or more links in a network. (3) In a network, a point where one or more functional units interconnect transmission lines.

---

## O

---

**open.** (1) To make an adapter ready for use. (2) A break in an electrical circuit. (3) To make a file ready for use.

**operator information area (OIA).** The area below the line near the bottom of the display area where graphics and alphanumeric characters are displayed to define the status of the terminal or the system to the operator.

**option.** (1) A specification in a statement, a selection from a menu, or a setting of a switch, that may be used to influence the execution of a program. (2) A hardware or software function that may be selected or enabled as part of a configuration process. (3) A piece of hardware (such

as a network adapter) that can be installed in a device to modify or enhance device function.

**order code.** A code that may be included in the write data stream transmitted for a display station or printer; provides additional formatting or definition of the write data.

**order sequence.** A sequence in the data stream that starts with an order code and includes a character address and/or data characters related to the order code.

**original equipment manufacturer (OEM).** A manufacturer of equipment that may be marketed by another manufacturer.

**output device.** A device in a data processing system by which data may be received from the system.

---

## P

---

**pacings.** (1) A technique by which a receiving station controls the rate of transmission of a sending station to prevent overrun. (2) In SNA, a technique by which a receiving component controls the rate of transmission of a sending component to prevent overrun or congestion.

**parallel.** (1) Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus. (2) Pertaining to concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device. (3) Pertaining to concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. (4) Pertaining to the simultaneity of two or more processes. (5) Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. (6) Contrast with *serial*.

**parameter.** (1) A variable that is given a constant value for a specified application and that may denote the application. (2) An item in a menu for which the user specifies a value or for which the system provides a value when the menu is interpreted. (3) Data passed between programs or procedures.

**port.** (1) An access point for data entry or exit. (2) A connector on a device to which cables for other devices such as display stations and

printers are attached.

**Power-On Self-Test (POST).** A series of diagnostic tests that are run each time the computer's power is turned on.

**primary logical unit (PLU).** In SNA, the logical unit (LU) that contains the primary half-session for a particular LU-LU session. Contrast with *secondary logical unit*.

**program access (PA) key.** On a display device keyboard, a key that produces a call to a program that performs display operations. See also *program function (PF) key*.

**program attention key.** On a display device keyboard, a key that produces an interruption to solicit program action. See also *program access (PA) key* and *program function (PF) key*.

**program function (PF) key.** On a display device keyboard, a key that passes a signal to a program to call for a particular display operation. See also *program access (PA) key*.

**programmable symbols (PS).** Customer-defined symbols. There are a maximum of 190 symbols in a programmed symbol set.

**programmed symbol set (PSS).** A set of fonts that can be system-defined or defined by the user and to which a code can be assigned.

**programmed symbols (PS).** In the 3270 Information Display System, an optional feature that stores up to six user-definable, program-loadable character sets of 190 characters each in terminal read/write storage for display or printing by the terminal.

**Program Tab (PT) order.** An order that advances the current buffer address to the address of the first character location of the next unprotected field and resets the character attributes of all characters in the field that are replaced by nulls.

**protected field.** (1) In word processing, preset data or an area that cannot be changed or overridden by an operator without altering the program. (2) On a display device, a display field in which a user cannot enter, modify, or erase data. Contrast with *unprotected field*.

**protocol.** (1) A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication. (2) In SNA, the meanings of and the sequencing rules for requests and responses used for managing the network, transferring data, and synchronizing the states of network components. (3) A specification for the format and relative

timing of information exchanged between communicating parties.

**put.** In 3174 central site customizing, to store data from the working copy into a library member.

---

## R

---

**remote.** Pertaining to a system, program, or device that is accessed through a telecommunication line. Contrast with *local*.

**remove.** (1) To take an attaching device off a network. (2) To stop an adapter from participating in passing data on a network.

**Repeat to Address (RA) order.** An order that stores a specified alphanumeric or null character in up to 480 buffer locations, starting at the current buffer address and ending at, but not including, the specified stop address.

**request for price quotation (RPQ).** An alteration or addition to the functional capabilities that the controller provides.

**ring out (RO).** In an IBM Token-Ring Network, the transmit or output receptacle on an access unit or repeater.

**routing.** (1) The assignment of the path by which a message will reach its destination. (2) In SNA, the forwarding of a message unit along a particular path through a network, as determined by parameters carried in the message unit, such as the destination network address in a transmission header.

---

## S

---

**scroll.** To move all or part of the display image vertically or horizontally to display data that cannot be observed within a single display image.

**secondary logical unit (SLU).** In SNA, the logical unit (LU) that contains the secondary half-session for a particular LU-LU session. Contrast with *primary logical unit*.

**segment.** A section of cable between components or devices on the network. A segment may consist of a single patch cable, multiple patch cables connected, or a combination of building cable and patch cables connected.

**selector pen.** A pen-like instrument that can be attached to a display station. When a program using full-screen processing is assigned to the display station, the pen can be used to select items on the screen or to generate an attention. Synonym for *light pen*.

**serial.** (1) Pertaining to a process in which all events occur one after the other; for example, serial transmission of the bits of a character according to V24 CCITT protocol. (2) Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. (3) Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. (4) Contrast with *parallel*.

**server.** (1) A device, program, or code module on a network dedicated to providing a specific service to a network. (2) On a LAN, a data station that provides facilities to other data stations. Examples are a file server, print server, and mail server.

**session.** (1) In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations. (2) In MLT, synonymous with logical terminal (LT). (3) In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested.

**Set Attribute (SA) order.** (1) An order that specifies an attribute-type-value pair defining the property to be applied to subsequent characters in the data stream. An SA order is required for each property assigned. (2) An order that associates attributes in the EAB with individual characters.

**Set Buffer Address (SBA) order.** An order that sets the buffer address to a specified location.

**Set Printer Characteristics.** This 3270 data stream structured field allows an application program to control the setting and resetting of certain functions on CUT printers.

**simulation.** (1) The representation of selected characteristics of the behavior of one physical or abstract system by another system. In a digital computer system, simulation is done by software; for example, (a)

the representation of physical phenomena by means of operations performed by a computer system, and (b) the representation of operations of a computer system by those of another computer system. (2) Contrast with *emulation*.

**SNA character string (SCS).** A character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

**solid-state component.** A component whose operation depends on control of electric or magnetic phenomena in solids, for example, a transistor, crystal diode, or ferrite core.

**Start Field (SF) order.** (1) A data stream order that establishes the start of a data field for displaying or printing. (2) An order that indicates a specified location that contains an attribute byte and not a text character.

**Start Field Extended (SFE) order.** (1) A data stream order that defines the start of a field that includes extended field attribute type-value pairs. (2) An order that generates an extended field attribute in the EAB and at the current buffer location.

**station.** (1) An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line. (2) A location in a device at which an operation is performed, for example, a read station. (3) In SNA, a link station.

**storage.** A unit into which recorded text can be entered, in which it can be retained and processed, and from which it can be retrieved. See also *memory*.

**structured field.** A data stream format that permits variable-length data and controls to be parsed into its components without having to scan every byte.

**subsystem.** A secondary or subordinate system, or programming support, usually capable of operating independently of or asynchronously with a controlling system. The 3174 and its attached terminals are an example of a subsystem.

**Suppress Index (SI) order.** An order that generates the suppress index character, valid only for the 3288 Model 2 printer. This character inhibits a line index to allow overprinting.

**synchronous.** (1) Pertaining to two or more processes that depend on the occurrences of a specific event, such as common timing signal.

(2) Occurring with a regular or predictable time relationship.

**Synchronous Data Link Control (SDLC).** A discipline conforming to subsets of the Advance Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop. See also *binary synchronous communication (BSC)*.

**system configuration.** A process that specifies the devices and programs that form a particular data processing system.

**Systems Application Architecture (SAA).** An architecture developed by IBM that consists of a set of selected software interfaces, conventions, and protocols, and that serves as a common framework for application development, portability, and use across different IBM hardware systems.

**system services control point (SSCP).** In SNA, the focal point within an SNA network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units (PUs) and logical units (LUs) within its domain.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

---

## T

---

**telecommunication-attached.** Pertaining to the attachment of devices by teleprocessing lines to a host processor. Synonym for *remote*. Contrast with *channel-attached*.

**terminal.** In data communication, a device, usually equipped with a keyboard and display device, capable of sending and receiving information.

**terminal component.** A separately addressable part of a terminal that performs an input or output function, such as the display component of a keyboard-display device or a printer component of a keyboard-printer

device.

**terminal emulation.** The capability of a microcomputer, personal computer, 3270 CUT mode display station, 3270 printer, ASCII display station, or ASCII printer to operate as if it were a particular type of terminal linked to a processing unit and to access data.

**transaction.** In an SNA network, an exchange between two programs that usually involves a specific set of initial input data that causes the execution of a specific task or job. Examples of transactions include the entry of a customer's deposit that results in the updating of the customer's balance, and the transfer of a message to one or more destination points.

**transparency.** See *transparent*

**transparent.** (1) Pertaining to operations or data that are of no significance to the user. (2) In data transmission, pertaining to information not recognized by the receiving program or device as transmission control characters.

**type.** In the 3174 Establishment Controller, the identifying number of a card. For example, 9150 is a type number of the terminal adapter in the 3174.

---

## U

---

**unbind.** In SNA, to deactivate a session between logical units.

**unbind command.** A command used to reset the protocols for a session. Contrast with *bind command*.

**unformatted display.** A display screen on which the user has not defined a display field. Contrast with *formatted display*. See also *protected field*.

**unprotected field.** A displayed field in which a user can enter, modify, or delete data. Contrast with *protected field*.

**update.** In 3174 central site customizing, to tailor a library member's customizing data, in working copy, and put it back to the library disk.

**upstream.** (1) In the direction opposite to data flow or toward the source of transmission. (2) Toward the processor from an attached unit or end user. (3) Contrast with *downstream*.

---

## V

---

**variable.** (1) In computer programming, a character or group of characters that refers to a value and, in the execution of a computer program, corresponds to an address. (2) A quantity that can assume any of a given set of values.

**vector.** One or more related fields of data, in a specified format. A quantity usually characterized by an ordered set of numbers.

**version.** A separate IBM-licensed program, based on an existing IBM-licensed program, that usually has significant new code or new function.

**viewport.** In the 3270 Information Display System, an area on the usable area of the display surface through which an operator views all or a portion of the data outlined by the window on the presentation plane.

---

## W

---

**workstation.** (1) An I/O device that allows either transmission of data or the reception of data (or both) from a host system, as needed to perform a job: for example, a display station or printer. (2) A configuration of I/O equipment at which an operator works. (3) A terminal or microcomputer, usually one connected to a mainframe or network, at which a user can perform tasks.

**write.** To make a permanent or transient recording of data in a storage device or on a data medium.

**write control character (WCC).** A character used in conjunction with a Write command to specify that a particular operation, or combination of

operations, is to be performed at a display station or printer.

**Write Structured Field (WSF) command.** A command used to transmit data in structured field format.

---

## 3

---

**3270 data stream.** (1) The commands, control codes, orders, attributes, and data or structured fields for 3270 devices, that are transmitted inbound to an application program or outbound to a terminal. (2) Data being transferred from or to an allocated primary or tertiary device, or to the host system, as a continuous stream of data and 3270 Information Display System control elements in character form.

---

## Index

### A

---

Activate Partition structured field, [2.5](#), [5.1](#), [5.5](#)  
format of, [5.5](#)  
addressing  
ASCII, [C.0](#)  
EBCDIC, [C.0](#)  
modes, in Create Partition structured field, [4.3.3](#)  
12-bit, [C.0](#)  
14-bit, [C.0](#)  
16-bit, [C.0](#)  
affirmative replies to trigger queries, [4.4.9](#)  
AID  
See attention identifier (AID)  
algorithm, compression, examples of use, [5.13.12](#)  
alphanumeric fields, [1.4.1](#)  
alphanumeric operation of the magnetic stripe reader, [7.2.2](#)  
Alphanumeric Partitions structured field (Query Reply), [6.9.2](#)  
format of, [6.9.2](#)  
altering field attributes, [4.4.1](#)  
alternate character set, [4.3.10](#), [6.12.3.3](#), [6.12.4.3](#)  
American National Standard Code for Information Interchange  
nonsupport for WSF command, [3.3](#)  
Anomaly Implementation structured field (Query Reply), [D.2](#)  
format of, [D.2.2](#)  
APL mismatch, [8.5.1](#)  
application initiated reads, [3.8.2](#)  
application program codes, MSR, [7.2.2.1](#)  
Application-to-Application, 3270 PC., [D.2.4](#)  
ASCII  
See American National Standard Code for Information Interchange

- attention identifier (AID), [1.2](#)
  - bytes sent from display to application (table), [3.5.6](#)
  - in the inbound data stream, [3.5.6](#)
- attribute pair, [1.4.3](#)
  - definition of, [1.4.3](#)
- attribute types, [1.4](#)
  - and associated attribute values, [1.4.3](#)
  - and selection rules, [4.4.5](#)
  - character, [1.4](#)
  - character set, [1.4.3](#)
  - codes for (table), [4.4.5](#)
  - color, [1.4.3](#)
  - extended field, [1.4](#)
  - extended highlighting, [1.4.3](#)
  - field, [1.4](#)
  - field outlining, [1.4.3](#)
  - field validation, [1.4.3](#)
  - transparency, [1.4.3](#)
- attribute values, [1.4.3](#)
  - all character attributes, [4.4.6.1](#)
  - and selection rules, [4.4.6](#)
  - extended highlighting, [4.4.6.3](#)
  - 3270 field attribute, [4.4.6.2](#)
- attributes
  - background transparency, [4.4.6.7](#)
  - character, [4.4](#), [4.4.3](#)
  - Color, [4.4.6.4](#)
  - conflict resolution between, [4.4.4](#)
  - data stream, types of, [1.4](#)
  - default values for, [4.6](#)
  - Extended Field, [4.4.2](#)
  - extended field, characteristics defined by, [1.4.2](#), [1.4.3](#)
  - field, [4.4](#), [4.4.1](#)
  - field outlining, [4.4.6.6](#)
  - field validation, [4.4.6.8](#)
  - introduction to, [4.4](#)
  - mandatory entry, [4.4.8](#)
  - mandatory fill, [4.4.7](#)
  - trigger, [4.4.9](#)
- automatic scroll, [7.3.7.4](#)
- autoskip field, [1.4.1](#)
- auxiliary device exception condition parameter, [6.2.4](#)
- auxiliary device status, [6.2.5](#)
- Auxiliary Device structured field (Query Reply), [6.10](#)
  - format of, [6.10.2](#)
- auxiliary devices and work stations
  - data routing, [11.2](#)
  - exception handling, [11.6](#)
  - input control, [11.4](#)
  - interaction with displays, [11.5](#)
  - introduction to, [11.1](#)
  - Query Reply, [11.3](#)
- availability of printers, [8.4](#)

## B

---

- background transparency, [4.4.6.7](#)
  - value byte, valid bit settings for, [4.4.6.7](#)
- base color, [1.4.1](#)
- Begin/End of File structured field (Query Reply), [6.11](#)
  - format of, [6.11.2](#)
- Binary Synchronous Communications environment
  - copy command, [9.12](#)
  - host acknowledgments, [9.8](#)
  - introduction, [9.1](#)

- processing of read commands, [9.10](#)
- read commands in, [9.4](#)
- read partition query structured fields, processing of, [9.11](#)
- read state transitions, [9.7.3](#)
- read states, [9.7](#)
- retry states, [9.7.4](#)
- transparent mode, [9.1](#)
- write commands in, [9.3](#)
- bit definitions for field attributes (table), [4.4.1](#)
- bit string, compressed, creating, [5.13.10](#)
- bits, terminator, [5.13.11](#)
- BSC
  - See Binary Synchronous Communications environment
  - BSC environment, local copy command in, [8.7](#)
  - buffer allocation self-defining parameter, [6.9.4](#)
  - byte, command, [1.1](#)

## C

---

- cabling system
  - See IBM Cabling System
- character
  - contained in magnetic stripe, [7.2.1](#)
  - designator, [7.4.2](#)
- character and extended field attributes (figure), [1.4](#)
- character attribute
  - and SA orders, examples (table), [4.3.4](#)
  - characteristics defined by, [1.4.3](#)
  - conflict resolution, [4.4.4](#)
  - DBCS, [12.4](#)
  - Extended Field Attribute (EFA) mismatch, [8.5.2](#)
  - overlap in, [4.5](#)
  - processing of, [4.5](#)
  - values of, [4.4.6.1](#)
  - what it does, [4.4](#), [4.4.3](#)
- character buffer, [2.2](#)
  - and the 3270 data stream, [1.3](#)
  - definition of, [2.2](#)
  - general information, [2.2](#)
  - locations, address range, [2.2](#)
  - mapping the display to, figure, [1.3](#)
- character cell
  - division of, [5.13.7](#)
  - slicing of (figure), [5.13.8](#)
- character set
  - alphanumeric, magnetic stripe, [7.2.2.1](#)
  - and SA orders, examples (table), [4.3.4](#)
  - attribute type and values, DBCS, [12.3.1](#)
  - introduction to, [4.2](#)
  - loadable, [4.2.2](#)
  - nonloadable, [4.2.1](#)
  - numeric, magnetic stripe, [7.2.2.1](#)
  - numeric, used for 3275/3277 compatibility mode, [7.2.1](#)
  - referencing, DBCS, [12.5.1](#)
  - value byte, valid settings for, [4.4.6.5](#)
- character set attribute type, [1.4.3](#)
- Clear function, [7.3.3](#)
- Clear key, effects of, [2.6](#), [4.4.1](#)
- Clear Partition function, [7.3.2](#)
- codepoints used in DBCS-Asia, [12.2](#)
- codes for attribute types (table), [4.4.5](#)
- codes for orders, [4.3](#)
- codes, display-generated, [7.2.2.1](#)
- codes, stripe, [7.2.2.1](#)
- color

- and SA orders, examples (table), [4.3.4](#)
- attribute type, [4.4.6.4](#)
- character attribute overlap (example), [4.5](#)
- value byte, valid settings for, [4.4.6.4](#)
- color attribute type, [1.4.3](#)
- Color structured field (Query Reply), [6.13](#)
  - example, [6.13.4](#)
  - format of, [6.13.2](#)
- color, base, [1.4.1](#)
- command byte, [1.1](#)
- command codes
  - in RU chains, [3.1](#)
  - list of, [3.3](#)
- commands
  - data stream, [3.1](#)
  - Erase All Unprotected, [1.5](#), [3.5.5](#)
  - Erase/Write, [1.5](#), [3.5.2](#)
  - Erase/Write Alternate, [1.5](#), [3.5.3](#)
  - hexadecimal index of, [F.2](#)
  - in a non-SNA environment, [10.2](#)
  - local copy, in the BSC environment, [8.7](#)
  - orders sent with write commands, [1.6](#)
  - Read Buffer, [1.5](#), [3.6.1.1](#)
  - Read Modified, [1.5](#), [3.6.2](#)
  - Read Modified All, [1.5](#), [3.6.2.5](#)
  - sending information to display, [1.5](#)
  - used by applications, [1.5](#)
  - within structured fields, [3.2](#)
  - Write, [1.5](#), [3.5.1](#)
  - write commands, general format of, [3.1](#)
  - Write Structured Field, [1.5](#), [3.5.4](#)
  - 3270 data stream, [1.5](#)
- comparison
  - rules and header bits, [5.13.9](#)
- compatibility of printers with displays, [8.5](#)
- compressed bit string, creating, [5.13.10](#)
- compression, [5.13.6](#)
  - algorithm, examples of use, [5.13.12](#)
  - creating the compressed bit string, [5.13.10](#)
  - overview, [5.13.6](#)
  - process of, [5.13.8](#)
- concepts, [1.1](#)
- conditions
  - enter-inhibit, [3.11.3](#)
  - input-inhibit, [3.11](#)
  - partition wait (PWAIT), [2.7](#)
  - system lock, [2.8](#), [3.11.2](#)
  - terminal wait, [2.9](#)
- conflict resolution between attributes, [4.4.4](#)
- considerations for partition mode, [8.6](#)
- control codes, LU 1 SCS, [8.8](#)
- control orders, format, [4.3.11](#)
- copy command, in BSC, [9.12](#)
- copy control character, [8.7](#)
- copy initiation, printers, [8.3](#)
- Create Partition structured field, [2.3.2](#), [5.1](#), [5.7](#)
  - format of, [5.7](#)
- cursor, [1.3](#)
  - in partitions, [2.2.3](#)
  - overview, [7.3.1.1](#)

## D

---

- Data Chain structured field, [5.1](#), [5.34](#)
  - format of, [5.34.2](#)

- data chaining (non-SNA), [5.2.2](#)
- data pending states, non-SNA environment, [10.5.2](#)
- data routing, [11.2](#)
- data stream
  - attention identifiers in, [1.2](#)
  - attributes, types of, [1.4](#)
  - commands, [1.5](#), [3.1](#)
  - format of, [1.2](#)
  - inbound, contents of, [1.2](#)
  - inbound, format of, [3.6](#)
  - inbound, orders in, [1.6](#)
  - inbound, orders permitted in, [4.3](#)
  - IPDS, [8.10](#)
  - non-SNA, [8.10](#)
  - orders, [1.6](#)
  - orders and attributes, [4.1](#)
  - outbound, contents of, [1.2](#)
  - overview, [1.2](#)
  - processing, DBCS, [12.3.2](#)
  - sense codes (SNA), [A.0](#)
  - SNA format, [1.1](#)
  - structured field errors, sense codes for, SNA, [A.0](#)
  - structured fields in, [1.7](#)
  - Write Structured Field, format of, [3.5.4](#)
- data, definition of, [1.2](#)
- DBCS
  - See double byte-coded character set (DBCS-Asia)
- DBCS-Asia structured field (Query Reply), [6.17](#)
  - format of, [6.17.2](#)
- DDM (Distributed Data Management) structured field (Query Reply), [6.19](#)
  - format of, [6.19.2](#)
- defaults
  - for attributes, [4.6](#)
- Delete key, [7.3.1.7](#)
- designator characters, [7.4.2](#)
- Destination/Origin structured field, [5.1](#), [5.35](#)
  - format of, [5.35.2](#)
- Destroy Partition structured field, [2.5](#), [5.1](#), [5.8.1](#)
  - format of, [5.8.1](#)
- detectable field, [1.4.1](#)
- Device Characteristics structured field (Query Reply), [6.18](#)
  - format of, [6.18.2](#)
- display interaction with auxiliary devices, [11.5](#)
- display states, [2.6](#)
- display-generated codes, [7.2.2.1](#)
- display/printer compatibility, [8.5](#)
- distributed data management
  - See Query Reply structured fields, DDM (Distributed Data Management)
- Document Interchange Architecture structured field (Query Reply), [6.20](#)
  - format of, [6.20.2](#)
- double byte-coded character set (DBCS-Asia)
  - character attribute, [12.4](#)
  - character set attribute type and values, [12.3.1](#)
  - codepoints used, [12.2](#)
  - data stream processing, [12.3.2](#)
  - Delete key in, [12.1](#)
  - exception conditions for SO/SI, [12.5.2](#)
  - fields, [12.3](#)
  - graphic character input, [12.6.3](#)
  - graphic codes, [12.2.1](#)
  - introduction, [12.1](#)
  - operator interface, [12.3.3](#)
  - orders, [12.7](#)
  - referencing the character set, [12.5.1](#)
  - set attribute (SA) order, [12.6](#)
  - shift out (SO)/shift in (SI), [12.5](#)
  - SO/SI interaction, [12.6](#)
  - types of fields, [12.6.2](#)
- Duplicate (DUP) key, [7.3.1.8](#)

## E

---

early print complete, [8.9](#)  
EBCDIC  
  See Extended Binary-Coded Decimal Interchange Code (EBCDIC)  
enter actions  
  overview, [3.11](#)  
  processing of, [3.11.4](#)  
enter actions, operator, [3.8.1](#)  
enter-inhibit condition, [3.8.1](#), [3.11.3](#)  
EPC  
  See early print complete  
Erase All Unprotected command, [1.5](#)  
  code for, [3.3](#)  
  what it does, [3.5.5](#)  
Erase EOF key, [7.3.1.4](#)  
Erase Input key, [7.3.1.5](#)  
Erase Unprotected to Address order  
  code for, [4.3](#)  
  what it does, [4.3.9](#)  
erase write command, [1.5](#)  
Erase/Reset structured field, [5.1](#), [5.9.2](#)  
  format of, [5.9.2](#)  
Erase/Write Alternate command, [1.5](#)  
  code for, [3.3](#)  
  in BSC environment, [9.3](#)  
  what it does, [3.5.3](#)  
Erase/Write command  
  code for, [3.3](#)  
  in BSC environment, [9.3](#)  
  what it does, [3.5.2](#)  
Error Override key, [7.3.4](#)  
EUA  
  See Erase Unprotected to Address order  
exception handling, auxiliary devices and workstations, [11.6](#)  
Exception/Status structured field, [6.2.2](#)  
  format of, [6.2.2](#)  
explicit partition, [2.3.2](#)  
extended and character field attributes (figure), [1.4](#)  
Extended Binary-Coded Decimal Interchange Code (EBCDIC)  
  command codes, [3.3](#)  
  in AID bytes, [3.5.6](#)  
  WCC byte, ASCII-translatable, [3.4](#)  
extended field attribute  
  and formatted fields, [1.3](#)  
  characteristics defined by, [1.4.2](#)  
  general discussion, [4.4](#)  
  what they do, [4.4.2](#)  
Extended File Attribute/Character Attribute mismatch, [8.5.2](#)  
extended highlighting attribute type, [1.4.3](#)

## F

---

field attribute, [1.3](#), [1.4.1](#), [4.3.2](#)  
  altering, [4.4.1](#)  
  and formatted fields, [1.3](#)  
  bit definitions for (table), [4.4.1](#)  
  values of, [4.4.6.2](#)  
  what it does, [4.4](#), [4.4.1](#)  
field format, selector pen, [7.4.1](#)

Field Mark key, [7.3.1.9](#)  
field outlining, [4.4.6.6](#)  
field outlining attribute, [1.4.3](#)  
Field Outlining structured field (Query Reply), [6.22](#)  
  format of, [6.22.2](#)  
field validation attribute, [1.4.3](#), [4.4.6.8](#)  
  value byte, valid bit settings for, [4.4.6.8](#)  
Field Validation structured field (Query Reply), [6.23](#)  
  format of, [6.23.2](#)  
fields  
  defined characteristics of, [1.4.1](#)  
format  
  data stream, [1.2](#)  
format control orders, [4.3.11](#)  
  control codes, format, [4.3.11](#)  
  format control codes, [4.3.11](#)  
Format Presentation structured field (Query Reply), [6.24](#)  
  format of, [6.24.2](#)  
formatted fields, example of (figure), [1.3](#)

## G

---

GE  
  See Graphic Escape order  
general concepts, [1.1](#)  
graphic character input, DBCS, [12.6.3](#)  
graphic codes, DBCS-Asia, [12.2.1](#)  
  field, DBCS, [12.3](#)  
Graphic Color structured field (Query Reply), [6.26](#)  
  format of, [6.26.2](#)  
Graphic Escape order  
  code for, [4.3](#)  
  what it does, [4.3.10](#)  
Graphic Symbols Sets structured field (Query Reply), [6.27](#)  
  format of, [6.27.2](#)  
group parameter, definition of, [5.2](#)  
grouping of structured fields, [5.2](#)

## H

---

header bits, comparison rules and, [5.13.9](#)  
hexadecimal index, [F.1](#)  
highlight, [4.3.4](#), [6.28](#)  
  format of, [6.28.2](#)  
highlighting  
  extended, values of, [4.4.6.3](#)  
  value byte, valid settings for, [4.4.6.3](#)  
horizontal dimensional parameters descriptor, [6.18.5](#)  
  format of, [6.18.5.2](#)  
host acknowledgments, BSC, [9.8](#)  
host acknowledgments, non-SNA environment, [10.6](#)  
host retry, [3.8.4](#)

## I

---

IC

See Insert Cursor order

ID codes for outbound structured fields, list of, [5.1](#)

implicit partition, [2.3.1](#)

inbound data stream, contents of, [1.2](#)

inbound data stream, format of, [3.6](#)

inbound operation in BSC environment, [9.6](#)

inbound operations (INOP)

definition of, [3.7](#)

in BSC environment, [9.6](#)

in non-SNA environment, [10.4](#)

overview, [3.9](#)

inbound partition identifier (INPID)

overview, [3.10](#)

inbound structured fields

AID bytes sent from display to application (table), [3.5.6](#)

attention identifiers (AID) in, [3.5.6](#)

Exception/Status, [6.2](#)

inbound reply modes, [3.6.1](#)

Inbound Text Header, [6.3](#)

Inbound 3270DS, [6.4](#)

overview, [6.1](#)

Query Replies

Alphanumeric Partitions, [6.9](#)

Anomaly Implementation, [D.2](#)

Auxiliary Device, [6.10](#)

Begin/End of File, [6.11](#)

Color, [6.13](#)

DBCS-Asia, [6.17](#)

DDM, [6.19](#)

Device Characteristics, [6.18](#)

Document Interchange Architecture, [6.20](#)

Field Outlining, [6.22](#)

Field Validation, [6.23](#)

Format Presentation, [6.24](#)

Graphic Color, [6.26](#)

Graphic Symbol Sets, [6.27](#)

Highlight, [6.28](#)

introduction, [6.7](#)

Line Type, [6.33](#)

MSR Control, [6.34](#)

Null, [6.35](#)

OEM Auxiliary Device, [6.36](#)

overview, [6.7](#)

Paper Feed Techniques, [6.37](#)

Port, [6.39](#)

Procedure, [6.40](#)

Product Defined Data Stream, [6.41](#)

Query Reply, [6.8](#)

Reply Modes, [6.42](#)

RPQ Names, [6.43](#)

Save/Restore Format, [6.44](#)

Segment, [6.45](#)

Storage Pools, [6.47](#)

Summary, [6.48](#)

Text Partitions, [6.49](#)

Usable Area, [6.51](#)

3270 IPDS, [6.52](#)

Recovery Data, [6.5](#)

Type 1 Text Inbound, [6.6](#)

Inbound Text Header structured field, [6.3.2](#)

format of, [6.3.2](#)

inbound transmissions, [9.5](#)

inbound transmissions in a non-SNA environment, [10.3](#)

Inbound 3270DS structured field, [6.4.2](#)

format of, [6.4.2](#)

index, hexadecimal, [F.1](#)

indicators, [9.8](#)

indicators, non-SNA environment, [10.6](#)

INOP  
See inbound operations (INOP)  
INPID  
See inbound partition identifier (INPID)  
input control, auxiliary devices and workstations, [11.4](#)  
input transmissions, [8.10.4](#)  
input-inhibit, [3.11](#)  
See also enter-inhibit condition  
Insert Cursor order  
code for, [4.3](#)  
what it does, [4.3.6](#)  
Insert Mode key, [7.3.1.6](#)  
intensified display field, [1.4.1](#)  
introduction, [1.1](#)  
IPDS data stream, [8.10](#)  
IPDS selection, [8.10.1](#)

## K

---

keyboard  
actions and scrolling, [7.3.7.2](#)  
actions in partitions, [7.3.6](#)  
actions with attribute selection keys, [7.3.5](#)  
functions, [7.3](#)  
restore, [3.12.1](#)  
restore function, [3.4](#)  
keys  
Clear, [2.6](#)  
Delete, [7.3.1.7](#)  
Delete, in DBCS-Asia, [12.1](#)  
DUP (duplicate), [7.3.1.8](#)  
Erase EOF (erase to end of field), [7.3.1.4](#)  
Erase Input, [7.3.1.5](#)  
Field Mark, [7.3.1.9](#)  
Insert Mode, [7.3.1.6](#)  
keyboard functions, [7.3](#)  
PA (Program Attention) keys, [7.3.1.10](#)  
that affect the data stream, [7.3.1](#)

## L

---

light pen  
See selector pen  
Line Type structured field (Query Reply), [6.33](#)  
format of, [6.33.1](#)  
Load Line Type structured field, [5.1](#), [5.12.2](#)  
format of, [5.12.2](#)  
Load Programmed Symbols structured field, [5.1](#)  
loadable character sets, [4.2.2](#)  
local copy function of printers in an SNA environment, [8.2](#)  
longitudinal redundancy check, [7.2.1](#)  
LRC  
See longitudinal redundancy check  
LU 1 SCS control codes, [8.8](#)  
LUSTAT, [8.4](#)

## M

---

- magnetic slot reader, [7.2](#)
  - application program codes, [7.2.2.1](#)
  - cards, secure and nonsecure, [7.2.2.2](#)
  - numeric/alphanumeric operation, [7.2.2](#)
  - operation of, [7.2](#)
  - stripe codes, [7.2.2.1](#)
- magnetic-stripe cards, [7.2.2.2](#)
- management of presentation spaces (figure), [2.6](#)
- mandatory entry, [4.4.8](#)
- mandatory fill, [4.4.7](#)
  - programming notes for, [4.4.7](#)
- mapping the display to the character buffer (figure), [1.3](#)
- MF
  - See Modify Field order
- mismatch, APL, [8.5.1](#)
- Modify Field order
  - code for, [4.3](#)
  - order sequence, [4.3.5](#)
  - what it does, [4.3.5](#)
- Modify Partition structured field, [5.1](#), [5.14.2](#)
  - format of, [5.14.2](#)
- MSR
  - See magnetic slot reader
- MSR Control structured field (Query Reply), [6.34](#)
  - format of, [6.34.1](#)
- multiple partitions, [2.4](#)
- multiple usable area self-defining parameter, [6.51.4](#)

## N

---

- negative reply to trigger query, [4.4.9](#)
- non-SNA data chaining, [5.2.2](#)
- non-SNA data stream, [8.10](#)
- non-SNA environment
  - commands in, [10.2](#)
  - data pending states in, [10.5.2](#)
  - host acknowledgments in, [10.6](#)
  - inbound operations (INOPs) in, [10.4](#)
  - inbound transmissions in, [10.3](#)
  - indicators in, [10.6](#)
  - introduction to, [10.1](#)
  - normal read state in, [10.5.1](#)
  - processing of read command in, [10.8](#)
  - read commands in, [10.2.2](#)
  - read partition query structured fields, processing of, [10.9](#)
  - read states in, [10.5](#)
  - read-state transitions in, [10.5.3](#)
  - retry state in, [10.5.3](#)
  - write commands in, [10.2.1](#)
- nondetectable field, [1.4.1](#)
- nondisplay field, [1.4.1](#)
- nonloadable character sets, [4.2.1](#), [6.12.4.3](#)
- nonsecure magnetic-stripe cards, [7.2.2.2](#)
- normal read state, [3.8.5](#)
- normal read state, non-SNA environment, [10.5.1](#)
- Null structured field (Query Reply), [6.35](#)
- null suppression, [3.6.2.1](#)
- numeric fields, [1.4.1](#)
- numeric operation of the magnetic stripe reader, [7.2.2](#)

## O

Object Control structured field, 5.1, 5.36  
format of, 5.36.2

Object Data structured field, 5.1, 5.37  
format of, 5.37.2

Object Picture structured field, 5.1, 5.38

OEM Auxiliary Device structured field (Query Reply), 6.36  
format of, 6.36.2

OEM Data structured field, 5.1, 5.39  
format of, 5.39.3

operations  
in BSC mode, 9.1  
inbound, 3.7  
inbound (INOP), 3.9  
inbound, in BSC, 9.6  
read-modified, 3.6.2.1  
read, in SNA, 3.8  
short-read, 3.6.2.2  
write, 3.5

operator enter actions, 3.8.1

operator interface, DBCS, 12.3.3

orders, 3270  
codes for, 4.3  
data stream, 1.6  
DBCS, 12.7  
Erase Unprotected to Address, 4.3.9  
format control, 4.3.11  
Graphic Escape, 4.3.10  
hexadecimal index of, F.3  
in inbound data stream, 1.6  
insert cursor, 4.3.6  
introduction, 4.3  
Modify Field, 4.3.5  
permitted in inbound data stream, 4.3  
Program Tab, 4.3.7  
Repeat to Address, 4.3.8  
sent with write commands, 1.6  
Set Attribute, 4.3.4  
Set Buffer Address, 4.3.3  
Start Field, 4.3.1  
Start Field Extended, 4.3.2

outbound data stream, contents of, 1.2

outbound structured fields  
Activate Partition, 5.1, 5.5  
Create Partition, 5.1, 5.7  
Define Presentation Space Types, 5.7.4  
Destroy Partition, 5.1, 5.8  
Erase/Reset, 5.1, 5.9  
functions provided, 3.2  
introduction to, 5.1  
list of, with ID codes, 5.1  
Load Line Type, 5.1, 5.12  
Load Programmed Symbols, 5.1, 5.13  
Modify Partition, 5.1, 5.14  
operations performed within, 3.5.4  
Outbound Text Header, 5.1, 5.15  
Outbound 3270DS, 5.1, 5.16  
overview of, 5.4  
Present Absolute Format, 5.1, 5.17  
Present Relative Format, 5.1, 5.18  
Read Partition, 5.1, 5.19  
Reset Partition, 5.1, 5.21  
SCS Data, 5.1, 5.23  
Select Format, 5.25  
Select Format Group, 5.1  
Set Checkpoint Interval, 5.26  
Set MSR Control, 5.1, 5.27

- Set Reply Mode, [5.1](#), [5.30](#)
- Set Window Origin, [5.1](#), [5.31](#)
- Type 1 Text Outbound, [5.1](#), [5.32](#)
- Outbound Text Header structured field, [5.1](#), [5.15.2](#)
  - format of, [5.15.2](#)
- Outbound 3270DS structured field, [5.1](#), [5.16.2](#)
  - format of, [5.16.2](#)
- outbound/inbound structured fields
  - Data Chain, [5.1](#), [5.34](#)
  - Destination/Origin, [5.1](#), [5.35](#)
  - Object Control, [5.1](#), [5.36](#)
  - Object Data, [5.1](#), [5.37](#)
  - Object Picture, [5.1](#), [5.38](#)
  - OEM Data, [5.1](#), [5.39](#)
  - Save/Restore Format, [5.1](#), [5.40](#)
  - Select IPDS Mode, [5.41](#)
- outlining, field, [4.4.6.6](#)
- overview, [1.1](#)

## P

---

- spacing, [8.10.3](#)
- page presentation media descriptor, [6.18.7](#)
  - format, [6.18.7.2](#)
- Paper Feed Techniques structured field (Query Reply), [6.37](#)
  - format of, [6.37.2](#)
- parameters
  - auxiliary device, [6.2.5](#)
  - auxiliary device exception condition, [6.2.4](#)
  - buffer allocation, [6.9.4](#)
  - group, definition of, [5.2](#)
  - self-defining, in structured fields, [5.3](#)
  - symbol envelope table, [5.13.4](#)
  - terminator, [5.13.5](#)
- partition identifier (PID), [2.1](#)
- partition wait condition (PWAIT), [2.7](#)
  - and enter actions, [3.11.1](#)
- partitions, [2.1](#)
  - active, [2.4](#)
  - Clear function, [7.3.2](#)
  - explicit, [2.3](#), [2.3.2](#)
  - explicit partition state, [2.6](#)
  - implicit, [2.3](#), [2.3.1](#)
  - implicit partition state, [2.6](#)
  - introduction to, [2.1](#)
  - keyboard actions in, [7.3.6](#)
  - multiple, [2.4](#)
  - partition mode considerations, [8.6](#)
  - presentation space, [2.2](#)
  - PWAIT (partition wait condition), [2.7](#)
  - read functions for, [2.3](#)
  - read operations from, [3.7](#)
  - Read Partition Structured Field, [3.8.3](#)
  - scrolling, [7.3.7](#)
  - selecting fields in, [7.4.3](#)
  - system lock condition, [2.8](#)
  - the cursor in, [2.2.3](#)
  - TWAIT (terminal wait condition), [2.9](#)
  - 0 (zero), [2.3.2](#)
- PID
  - See partition identifier (PID)
- planes
  - See color
- Port structured field (Query Reply), [6.39](#)
  - format of, [6.39.2](#)

- Present Absolute Format structured field, [5.1](#), [5.17.2](#)
  - format of, [5.17.2](#)
- Present Relative Format structured field, [5.1](#), [5.18.2](#)
  - format of, [5.18.2](#)
- presentation plane
  - See presentation space
- presentation space, [2.2](#)
  - a conceptual view (figure), [2.2](#)
  - and viewport, relationship between, [2.2](#)
  - and viewport, without scrolling (figure), [2.2](#)
  - management of (figure), [2.6](#)
  - viewport, [2.2](#)
  - window and viewport, with scrolling (figure), [2.2.1](#)
- print complete, early, [8.9](#)
- printers
  - availability, [8.4](#)
  - considerations, [8.1](#)
  - copy initiation, [8.3](#)
  - display compatibility with, [8.5](#)
  - local copy function in an SNA environment, [8.2](#)
  - LUSTAT, [8.4](#)
  - overview, [8.1](#)
  - using the WCC byte with (table), [8.1](#)
- Procedure structured field (Query Reply), [6.40](#)
- processing of character attributes, [4.5](#)
- processing of enter actions, [3.11.4](#)
  - processing of read commands (alphanumeric), [3.12](#)
- Product Defined Data Stream structured field (Query Reply), [6.41](#)
- Program Attention keys, [7.3.1.10](#)
- Program Tab order
  - code for, [4.3](#)
  - what it does, [4.3.7](#)
- programmed symbols, [8.5.3](#)
- protected fields, [1.4.1](#)
- PT
  - See Program Tab order
- PWAIT
  - See partition wait condition (PWAIT)

## Q

---

- Query Reply structured field, [6.8.2](#)
  - format of, [6.8.2](#)
- Query Reply structured fields
  - Alphanumeric Partitions, [6.9](#)
  - Anomaly Implementation, [D.2](#)
  - Auxiliary Device, [6.10](#)
  - Begin/End of File, [6.11](#)
  - Color, [6.13](#)
  - DBCS-Asia, [6.17](#)
  - DDM (Distributed Data Management), [6.19](#)
  - Device Characteristics, [6.18](#)
  - Document Interchange Architecture, [6.20](#)
  - Field Outlining, [6.22](#)
  - Field Validation, [6.23](#)
  - Format Presentation, [6.24](#)
  - Graphic Color, [6.26](#)
  - Graphic Symbol Sets, [6.27](#)
  - Highlight, [6.28](#)
  - introduction, [6.7](#)
  - Line Type, [6.33](#)
  - MSR Control, [6.34](#)
  - Null, [6.35](#)
  - OEM Auxiliary Device, [6.36](#)
  - overview, [6.7](#)

Paper Feed Techniques, [6.37](#)  
Port, [6.39](#)  
Procedure, [6.40](#)  
Product Defined Data Stream, [6.41](#)  
Query Reply, [6.8](#)  
Reply Modes, [6.42](#)  
RPQ Names, [6.43](#)  
Save/Restore Format, [6.44](#)  
Segment, [6.45](#)  
Storage Pools, [6.47](#)  
Summary, [6.48](#)  
Text Partitions, [6.49](#)  
Transparency, [6.50](#)  
Usable Area, [6.51](#)  
3270 IPDS, [6.52](#)  
Query Reply, auxiliary devices and workstations, [11.3](#)

## R

---

### RA

See Repeat to Address order  
read acknowledgment, non-SNA, [10.7](#)  
read acknowledgment, SNA, [3.12.1](#)  
Read Buffer command, [1.5](#)  
    character mode, [3.6.1.4](#)  
    code for, [3.3](#)  
    field mode, [3.6.1.2](#)  
    in BSC environment, [9.4.1](#)  
    in read partition structured field, [3.2](#)  
    what it does, [3.6.1.1](#)  
read commands  
    extended field mode, [3.6.1.3](#)  
    in BSC environment, [9.4](#)  
    in non-SNA environment, [10.2.2](#)  
    in structured fields, [3.6.3](#)  
    inbound reply modes, [3.6.1](#)  
    processing of (alphanumeric), [3.12](#)  
    processing of, non-SNA environment, [10.8](#)  
    processing, in BSC environment, [9.10](#)  
    read acknowledgment, SNA, [3.12.1](#)  
    sent by application program, [3.6.1](#)  
Read Modified All command, [1.5](#)  
    character mode, [3.6.2.6](#)  
    code for, [3.3](#)  
    extended field mode, [3.6.2.6](#)  
    in read partition structured field, [3.2](#)  
    what it does, [3.6.2.5](#), [3.6.2.6](#)  
Read Modified command, [1.5](#)  
    character mode (figure), [3.6.2.4](#)  
    code for, [3.3](#)  
    extended field mode, [3.6.2.3](#)  
    field mode, [3.6.2.3](#)  
    in BSC environment, [9.4.2](#)  
    in read partition structured field, [3.2](#)  
    what it does, [3.6.2](#)  
read operations  
    application initiated, [3.8.2](#)  
    from partitions, [3.7](#)  
    in SNA, [3.8](#)  
    overview, [3.6](#)  
read partition structured field, [2.3](#), [3.8.3](#), [5.1](#), [5.19.2](#)  
    format of, [5.19.2](#)  
read state transitions, [3.8.5.4](#)  
read state transitions, BSC, [9.7.3](#)  
read state transitions, non-SNA environment, [10.5.3](#)

- read states, [3.8.5](#)
- read states in a non-SNA environment, [10.5](#)
- read states, BSC, [9.7](#)
- reader, magnetic, operation of, [7.2](#)
- reads, application initiated, [3.8.2](#)
- read, test request, [9.4.3](#)
- Recovery Data, [6.5.2](#)
  - format of, [6.5.2](#)
- refid=restore, keyboard, [3.12.1](#)
- Repeat to Address order
  - code for, [4.3](#)
  - what it does, [4.3.8](#)
- Reply Modes structured field, [6.42](#)
  - format of, [6.42.2](#)
- request/response unit (RU) chain
  - command codes in, [3.1](#)
  - commands to display, [3.1](#)
- reset actions, [B.0](#)
- reset partition structured field, [5.1](#), [5.21.2](#)
  - format of, [5.21.2](#)
- resolving conflicts between attributes, [4.4.4](#)
- retry read state, [3.8.5.3](#)
- retry states, BSC, [9.7.4](#)
- retry state, non-SNA environment, [10.5.3](#)
- retry, host, [3.8.4](#)
- RPQ Names structured field (Query Reply), [6.43](#)
  - format of, [6.43.2](#)
- RU chain
  - See request/response unit (RU) chain

## S

---

- Save/Restore Format structured field, [5.1](#), [5.40](#), [6.44](#)
  - format of, [5.40.2](#), [6.44.2](#)
- SBA
  - See Set Buffer Address order
- screens
  - formatted, [1.3](#)
  - unformatted, [1.3](#)
- scrolling, [2.2.1](#)
  - automatic, [7.3.7.4](#)
  - keyboard actions and, [7.3.7.2](#)
  - partitions, [7.3.7](#)
  - vertical, [7.3.7.1](#)
- SCS Data structured field, [5.1](#), [5.23.2](#)
  - format of, [5.23.2](#)
- secure magnetic-stripe cards, [7.2.2.2](#)
- Segment structured field (Query Reply), [6.45](#)
  - format of, [6.45.2](#)
- Select Format Group structured field, [5.1](#), [5.25.2](#)
  - format of, [5.25.2](#)
- Select IPDS Mode structured field, [5.41](#)
  - format of, [5.41.2](#)
- selecting fields in partitions, [7.4.3](#)
- selection rules and attribute types, [4.4.5](#)
- selection rules and attribute values, [4.4.6](#)
- selector pen
  - field format, [7.4.1](#)
  - operation of, [7.4](#)
- self-defining parameters in structured fields, [5.3](#)
- sense codes, SNA, [A.0](#)
- sense codes, structured field errors, SNA, [A.0](#)
- Set Attribute order, [4.3](#)
  - code for, [4.3](#)
  - what it does, [4.3.4](#)

set attribute order, DBCS, [12.6](#)  
Set Buffer Address order  
  code for, [4.3](#)  
  format of, [4.3.3](#)  
  lineal addressing of the presentation space, [2.2](#)  
  what it does, [4.3.3](#)  
Set Checkpoint Interval structured field, [5.1](#), [5.26.2](#)  
  format of, [5.26.2](#)  
Set MSR Control, [5.27.2](#)  
  format of, [5.27.2](#)  
Set MSR Control structured field, [5.1](#)  
set print density descriptor, [6.18.4](#)  
  format of, [6.18.4.2](#)  
Set Reply Mode structured field, [5.1](#), [5.30.2](#)  
  format of, [5.30.2](#)  
set text orientation descriptor, [6.18.8](#)  
  format, [6.18.8.2](#)  
Set Window Origin structured field, [5.1](#), [5.31.2](#)  
  format of, [5.31.2](#)  
SF  
  See Start Field order  
SFE  
  See Start Field Extended order  
shift in (SI), DBCS, [12.5](#)  
shift out (SO), DBCS, [12.5](#)  
short-read operation, [3.6.2.2](#)  
16 bit addressing  
  description of, [C.0](#)  
slices, [5.13.6](#)  
slot reader, magnetic, operation of, [7.2](#)  
SNA  
  See Systems Network Architecture (SNA)  
SO/SI interaction, DBCS, [12.6](#)  
spanning, [5.2.1](#)  
special applications, [D.1](#)  
Start Field Extended order  
  code for, [4.3](#)  
  format of, [4.3.2](#)  
  what it does, [4.3.2](#)  
Start Field order  
  code for, [4.3](#)  
  what it does, [4.3.1](#)  
state transitions, display, [2.6](#)  
states, display, [2.6](#)  
storage pool characteristics, [6.47.3](#)  
Storage Pools structured field (Query Reply), [6.47](#)  
  format of, [6.47.2](#)  
stripe codes, [7.2.2.1](#)  
stripe codes and application program codes, [7.2.2.1](#)  
structured fields  
  Activating Partitions, [2.5](#)  
  commands within, [3.2](#)  
  Create Partition, [2.3.2](#)  
  Destroy Partition, [2.5](#)  
  errors, SNA sense codes for, [A.0](#)  
  grouping, [5.2](#)  
  hexadecimal index of, [F.5](#)  
  in the 3270 data stream, [1.7](#)  
  inbound structured fields  
    Exception/Status, [6.2](#)  
    Inbound Text Header, [6.3](#)  
    Inbound 3270DS, [6.4](#)  
    overview, [6.1](#)  
  non-SNA data chaining, [5.2.2](#)  
  outbound structured fields  
    Activate Partition, [5.1](#), [5.5](#)  
    Create Partition, [5.1](#), [5.7](#)  
    Define Presentation Space Types, [5.7.4](#)  
    Destroy Partition, [5.1](#), [5.8](#)

Erase/Reset, [5.1](#), [5.9](#)  
introduction to, [5.1](#)  
list of, with ID codes, [5.1](#)  
Load Line Type, [5.1](#), [5.12](#)  
Load Programmed Symbols, [5.1](#), [5.13](#)  
Modify Partition, [5.1](#), [5.14](#)  
Outbound Text Header, [5.1](#), [5.15](#)  
Outbound 3270DS, [5.1](#), [5.16](#)  
overview of, [5.4](#)  
Present Absolute Format, [5.1](#), [5.17](#)  
Present Relative Format, [5.1](#), [5.18](#)  
Read Partition, [5.1](#), [5.19](#)  
Recovery Data, [6.5](#)  
Reset Partition, [5.1](#), [5.21](#)  
SCS Data, [5.1](#), [5.23](#)  
Select Format, [5.25](#)  
Select Format Group, [5.1](#)  
Set Checkpoint Interval, [5.1](#), [5.26](#)  
Set MSR Control, [5.1](#), [5.27](#)  
Set Reply Mode, [5.1](#), [5.30](#)  
Set Window Origin, [5.1](#), [5.31](#)  
Type 1 Text Inbound, [6.6](#)  
Type 1 Text Outbound, [5.1](#), [5.32](#)  
outbound/inbound structured fields  
Data Chain, [5.1](#), [5.34](#)  
Destination/Origin, [5.1](#), [5.35](#)  
general discussion of, [5.1](#)  
Object Control, [5.1](#), [5.36](#)  
Object Data, [5.1](#), [5.37](#)  
Object Picture, [5.1](#), [5.38](#)  
OEM Data, [5.1](#), [5.39](#)  
overview of, [5.33](#)  
Save/Restore Format, [5.1](#), [5.40](#)  
Select IPDS Mode, [5.41](#)  
Set Checkpoint Interval, [5.1](#)  
Query Reply structured fields  
Alphanumeric Partitions, [6.9](#)  
Anomaly Implementation, [D.2](#)  
Auxiliary Device, [6.10](#)  
Begin/End of File, [6.11](#)  
Color, [6.13](#)  
DBCS-Asia, [6.17](#)  
DDM, [6.19](#)  
Device Characteristics, [6.18](#)  
Document Interchange Architecture, [6.20](#)  
Field Outlining, [6.22](#)  
Field Validation, [6.23](#)  
Format Presentation, [6.24](#)  
Graphic Color, [6.26](#)  
Graphic Symbol Sets, [6.27](#)  
Highlight, [6.28](#)  
introduction, [6.7](#)  
Line Type, [6.33](#)  
MSR Control, [6.34](#)  
Null, [6.35](#)  
OEM Auxiliary Device, [6.36](#)  
overview, [6.7](#)  
Paper Feed Techniques, [6.37](#)  
Port, [6.39](#)  
Procedure, [6.40](#)  
Product Defined Data Stream, [6.41](#)  
Query Reply, [6.8](#)  
Reply Modes, [6.42](#)  
RPQ Names, [6.43](#)  
Save/Restore Format, [6.44](#)  
Segment, [6.45](#)  
Storage Pools, [6.47](#)  
Summary, [6.48](#)  
Text Partitions, [6.49](#)

- Usable Area, [6.51](#)
- 3270 IPDS, [6.52](#)
- read commands in, [3.6.3](#)
- read partition, [2.3](#), [3.8.3](#)
- read partition query, in BSC, [9.11](#)
- read partition query, processing of, non-SNA, [10.9](#)
- RU chains in, [3.1](#)
- self-defining parameters, [5.3](#)
- spanning, [5.2.1](#)
- types of, [1.7](#)
- variable length, [5.1](#)
- Summary structured field (Query Reply), [6.48](#)
- symbols, programmed, [8.5.3](#)
- system lock, [3.11.2](#)
- system lock condition, [2.8](#)
- Systems Application Architecture
  - support of, [E.1](#)
  - functions required, [E.1](#)
- Systems Network Architecture (SNA)
  - local copy function, printers, [8.2](#)
  - read acknowledgment, [3.12.1](#)
  - read operations in, [3.8](#)
  - sense codes, [A.0](#)

## T

---

- table
  - symbol envelope, [5.13.4](#)
- terminal wait condition, [2.9](#)
  - and enter actions, [3.11.1](#)
- terminator bits, [5.13.11](#)
- terminator self-defining parameter, [5.13.5](#)
- test card, [7.2.3](#)
- test request read, [9.4.3](#)
- Text Partitions structured field (Query Reply), [6.49](#)
- 3270 PC Application-to-Application
  - description, [D.2.4](#)
- 3275/3277-compatible mode
  - operation, description of, [7.2.1](#)
- 3270 IPDS
  - description, [6.52](#)
- transmissions, inbound., [9.5](#)
- transmissions, input, [8.10.4](#)
- transparency, [1.4.3](#)
- transparent mode, in BSC, [9.1](#)
- trigger, [4.4.9](#)
- trigger attribute
  - affirmative reply to, [4.4.9](#)
  - negative reply to, [4.4.9](#)
  - programming notes for, [4.4.9.1](#)
- TWAIT
  - See terminal wait condition
- Type 1 Text Inbound structured field, [6.6.2](#)
  - format of, [6.6.2](#)
- Type 1 Text Outbound structured field, [5.1](#), [5.32.2](#)
  - format of, [5.32.2](#)

## U

---

- unprotected fields, [1.4.1](#)

## V

---

variable length structured fields, [5.1](#)  
vertical dimensional parameters descriptor, [6.18.6](#)  
  format of, [6.18.6.2](#)  
vertical scrolling, [7.3.7.1](#)  
viewport, [2.1](#), [2.2](#)  
  and presentation space, relationship between, [2.2](#)  
  and presentation space, without scrolling (figure), [2.2](#)  
  presentation space and window, with scrolling (figure), [2.2.1](#)

## W

---

WCC

  See Write Control Character (WCC)

window

  on the presentation space, [2.2](#)  
  presentation space and viewport, with scrolling (figure), [2.2.1](#)

Write command, [1.5](#)

  what it does, [3.5.1](#)

write commands

  code for, [3.3](#)  
  format of (figure), [3.1](#)  
  in BSC environment, [9.3](#)  
  in non-SNA environment, [10.2.1](#)  
  orders sent with, [1.6](#)

Write Control Character (WCC)

  bit definitions (table), [3.4](#)  
  byte, for use with printers (table), [8.1](#)  
  following write commands, [1.5](#)  
  functions, order of operation, [3.4](#)  
  reset actions (for displays), table, [3.4](#)

write operation

  overview of, [3.5](#)

write structured field command, [1.5](#)

  code for, [3.3](#)  
  data stream, format of, [3.5.4](#)  
  nonsupport in ASCII environment, [3.3](#)  
  what it does, [3.5.4](#)

## Numerics

---

12-bit addressing, [C.0](#)

  description of, [C.0](#)

14-bit addressing, [C.0](#)

  description of, [C.0](#)

16-bit addressing, [C.0](#)

3270 data stream

  attention identifiers in, [1.2](#)  
  attributes, types of, [1.4](#)  
  commands, [1.5](#), [3.1](#)  
  format of, [1.2](#)  
  inbound, contents of, [1.2](#)  
  inbound, format of, [3.6](#)

inbound, orders in, 1.6  
inbound, orders permitted in, 4.3  
IPDS, 8.10  
non-SNA, 8.10  
orders, 1.6  
orders and attributes, 4.1  
outbound, contents of, 1.2  
overview, 1.2  
processing, DBCS, 12.3.2  
sense codes (SNA), A.0  
SNA format, 1.1  
structured field errors, sense codes for, SNA, A.0  
structured fields in, 1.7  
Write Structured Field, format of, 3.5.4  
3270 PC Application-to-Application, D.2.4  
3275/3277-compatible mode operation, 7.2.1

---

## COMMENTS Readers' Comments

3270 Information Display System  
Data Stream Programmer's Reference

Publication No. GA23-0059-07

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Be sure to print your name and address below if you would like a reply.

International Business Machines Corporation  
Information Development  
Department E02  
P.O. Box 12195  
Research Triangle Park, North Carolina 27709-9990

Name . . . . . \_\_\_\_\_  
Company or Organization \_\_\_\_\_  
Address . . . . . \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
Phone No. . . . . \_\_\_\_\_

---