Virtual Machine
Remote Spooling Communications Subsystem
Networking

General Information

Version 3 Release 2

GH24-5218-03

Virtual Machine
Remote Spooling Communications Subsystem
Networking

General Information

Version 3 Release 2

GH24-5218-03

Note:

Before using this information and the product it supports, read the information in “Notices” on page vii.

Fourth Edition (July 1999)

This edition applies to Version 3, Release 2, Modification 0 of IBM® Virtual Machine Remote Spooling Communications Subsystem Networking (VM/RSCS) (product number 5684-096) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces GH24-5218-02.

Contents

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes to the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Corporation
Mail Station P300,
522 South Road
Poughkeepsie, NY 12601-5400
U.S.A.
Attention: Information Request
```

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities on non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM

for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, or other countries, or both:

ACF/VTAM	Advanced Function Printing
APL2	AS/400
BookManager	CICS/VM
DFSMS/VM	ESA/390
ESCON	GDDM
IBM	IBMLink
IPDS	Intelligent Printer Data Stream
Library Reader	Language Environment
OfficeVision	OpenEdition
Print Services Facility	PSF
S/390	System/36
Virtual Machine/Enterprise Systems Architecture	VM/ESA
System/370	VTAM

Other company, product, and service names may be trademarks or service marks of others.

Preface

This book contains a high-level description of the VM Remote Spooling Communications Subsystem Networking (RSCS) licensed program. It is designed to offer a product overview for managers and planners to evaluate RSCS.

Who Should Read This Book

This book describes RSCS in sufficient detail to help new customers, RSCS customers migrating from a previous version, and IBM representatives to determine if RSCS is suitable for particular installations.

What You Should Know Before Reading This Book

This book introduces basic RSCS and networking concepts. It is not intended to be used as a reference for programming or operating activities. This book assumes that you are familiar with the Virtual Machine/Enterprise Systems Architecture (VM/ESA) operating system and its associated terminology. Familiarity with Systems Network Architecture (SNA) and Transmission Control Protocol/Internet Protocol for VM (TCP/IP) terms is also helpful.

What This Book Contains

This book is divided into several sections to help you locate the information you need about RSCS.

- Part 1, “Introduction” contains the following topics:
  - Chapter 1, “Introducing RSCS Network Support” on page 3 introduces RSCS and discusses networking concepts. It defines and briefly explains networking terms such as nodes and links. It also explains RSCS networking by giving examples of specific types of nodes and links in a RSCS network. Within a RSCS network, you can establish Systems Network Architecture (SNA) connections, non-SNA connections, and Transmission Control Protocol/Internet Protocol for VM (TCP/IP) connections.
  - Chapter 2, “RSCS Environments” on page 17 explains how RSCS can be used in both a single-system or a multi-system environment. It describes the advantages of both types of systems.
  - Chapter 3, “RSCS Users” on page 25 describes the two types of RSCS users: general users and operators. Most general users are not aware that they are using RSCS because they indirectly use it through CMS commands that send data. RSCS commands enable users to control their outgoing and incoming files. This chapter also describes the four types of RSCS operators and their functions.
  - Chapter 4, “RSCS Meets Your Business Needs” on page 27 shows you that RSCS is easy to install, operate, and use. It describes some tasks you can do with RSCS and directs you to other RSCS books where you can find more information.

  - Chapter 5, “Customizing RSCS” on page 31 describes the RSCS exit facilities and how you can use them at your installation. It offers an overview of the RSCS Exit Facility, Transmission Algorithms, ASCII Printer and Plotter Exits, Gateway Programming Interface, LPR Exits, LPD Exits, UFT Exits, UFTD Exits, and the Loadable Link Driver Facility. It also describes the exits available to the VM RSCS Data Interchange Manager.
- Part 2, “Features and Requirements” contains the following topics:
  - Chapter 6, “RSCS Version 3 Enhancements” on page 39 summarizes the improvements that have been made to RSCS Version 3, Release 2 and previous releases of RSCS.
  - Chapter 7, “RSCS Requirements” on page 59 describes the requirements for using RSCS. It includes information about product requirements; it also describes the tasks you must perform when using RSCS.
  - Chapter 8, “RSCS Library Guide” on page 63 summarizes the other publications in the RSCS library.
- Part 3, “Reference Information” contains the following topics:
  - “Glossary and Abbreviations” on page 69 defines many of the terms, abbreviations, and acronyms used in this book.
  - “Bibliography” on page 81 provides a list of books and order numbers in the RSCS library and in other product libraries that can help you when using RSCS.
  - “Using the Master Index” on page 85 explains how to use the master index of the RSCS library. The actual master index is provided in the following section.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other VM/ESA documentation:

- Visit our home page at:

  `http://www.ibm.com/s390/vm/related/rscs`

  There you will find the feedback page where you can enter and submit your comments.
- Send your comments by electronic mail to one of the following addresses:

Format	Address
Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL
IBMLink™	GDLVME(PUBRCF)

  Be sure to include the name of the book, the form number (including the suffix), and the page, section title, or topic you are commenting on.
- Fill out the form at the back of this book and return it by mail, by fax, or by giving it to an IBM representative.

Part 1. Introduction

The following sections contain information to introduce you to RSCS.

If you are already familiar with RSCS Version 3 and its related concepts, you should review Part 2, “Features and Requirements” on page 37 for information about RSCS features and product requirements.

Chapter 1. Introducing RSCS Network Support

VM Remote Spooling Communications Subsystem Networking (RSCS) is a networking program that enables users on one system to send messages, files, commands, and jobs to other users within a network. RSCS connects nodes (systems, devices, and workstations) using links. These links allow data, consisting mainly of CP spool files, to be transferred between these nodes: local and remote, adjacent and nonadjacent.

As its name implies, and as Figure 1 shows, RSCS enables you to exchange data with other users on other remote systems.

Remote A separate system or device that you access through a telecommunication line.

Spooling Storing and retrieving data on an auxiliary direct access storage device (DASD) so that it is convenient for later processing or output.

Communications Interaction or information (data) exchange with others.

Subsystem A system that runs under the control of another system.

Networking Data processing systems and devices connected with other systems and devices.

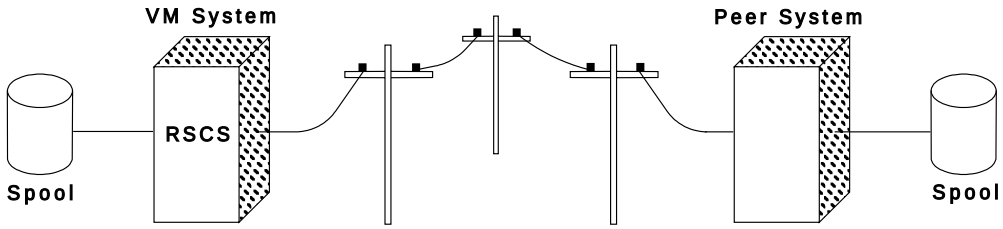

Figure 1. RSCS Transferring Spool Data Over Telecommunications Lines

RSCS is a subsystem that can:

- Handle data being sent to, from, or through its VM/ESA system
- Store and retrieve input and output data files on the VM/ESA system spool
- Use communications equipment to transfer data between its VM/ESA system and remote users, devices, and other systems.

Running under the Group Control System (GCS) component of VM/ESA, RSCS uses the spooling facilities of VM/ESA to store and retrieve data. VM/ESA handles data transfer *within* its system by means of spooling. RSCS extends VM/ESA's spooling capabilities, handling data transfer *between* its VM/ESA system and outside sources. Data is stored on a spool after RSCS receives it and until RSCS can forward it to its destination. RSCS uses communications equipment to transfer data between the local VM/ESA system and other systems or remote locations.

The following sections provide an overview of how RSCS works and the type of connections that you can establish within an RSCS network.

How RSCS Fits into Your VM/ESA System

RSCS runs in a virtual machine and relies on the Group Control System (GCS) component of VM/ESA for supervisor services. The RSCS virtual machine must be defined as part of a GCS *virtual machine group*. GCS allows group members to share the following:

- A common storage area for information exchange
- Multitasking services
- General I/O (input/output) services
- Other simulated-OS services.

For more information about GCS, see the *VM/ESA: Group Control System* reference.

How RSCS Manages Files

RSCS uses the VM system spool to manage file transfer. Like VM/ESA, RSCS uses the system spool for temporary file storage. Figure 2 shows VM/ESA using the system spool to store files moving within the local system (or node); that is, between its users and real devices.

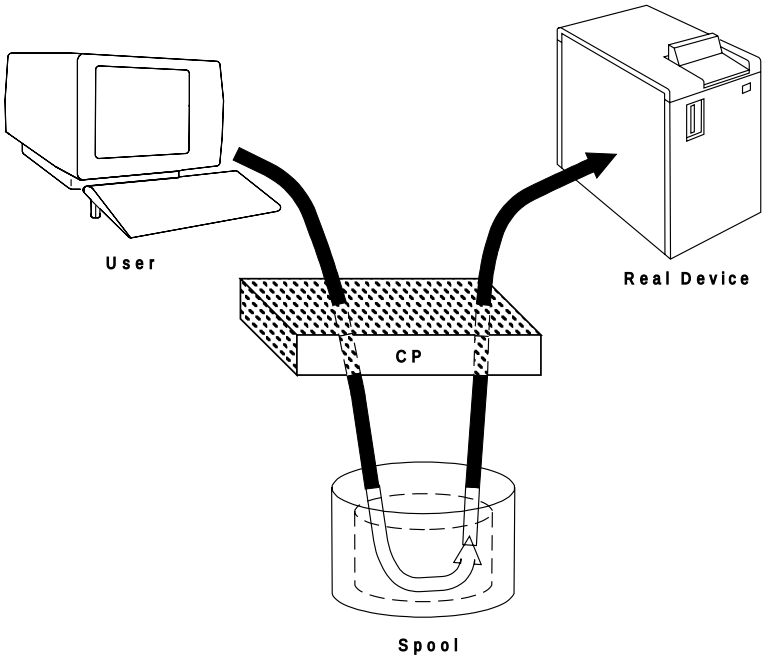

Figure 2. VM Use of the System Spool

RSCS extends VM spooling. It uses the spool to store files being transferred *between* its local system and remote users, devices, or systems. As Figure 3 on page 5 shows, these files are passed or “spooled” to RSCS for processing.

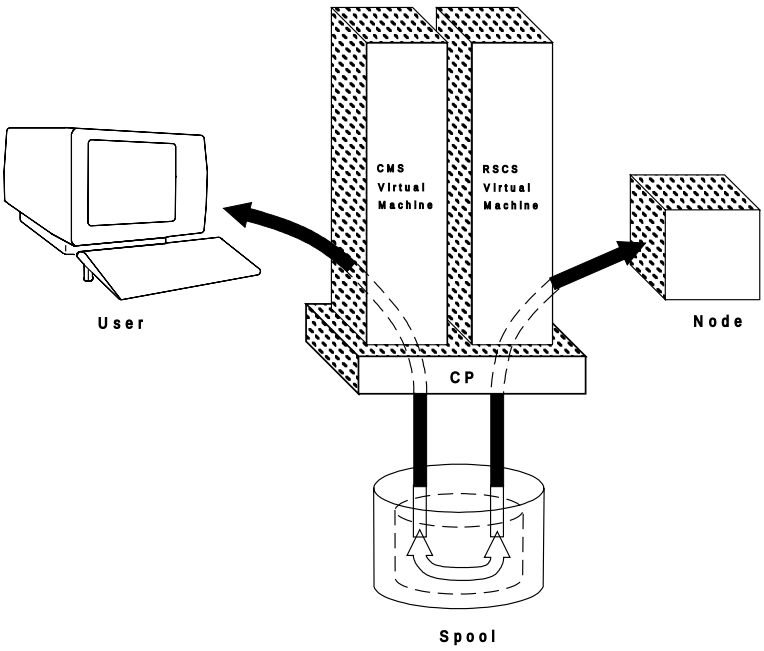

Figure 3. RSCS's Use of the System Spool

To manage files spooled to remote nodes, RSCS relies on *tag information*. A spool file "tag" becomes a part of each data file spooled to RSCS. The tag contains information that describes where the file came from (origin information) and where it is going (destination information). Figure 4 shows an example of origin and destination information in a tag.

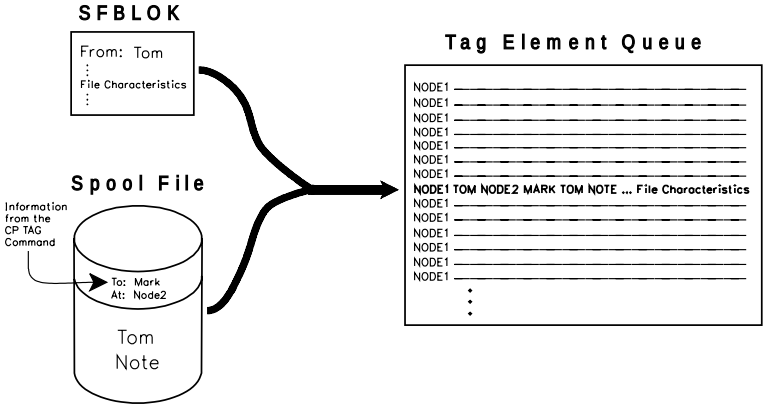

Figure 4. RSCS Spool Files Contain Tag Information

When RSCS receives a file, it records information from the spool file tag into a *tag slot*. RSCS uses the information in the tag slot to finish transferring the file to its destination.

Supported Nodes

Networks consist of systems and Input/Output (I/O) devices connected by communications equipment. The connections between various points in a network are called **links**. The points at which links meet or end are called **nodes**. Figure 5 on page 6 shows a simple network.

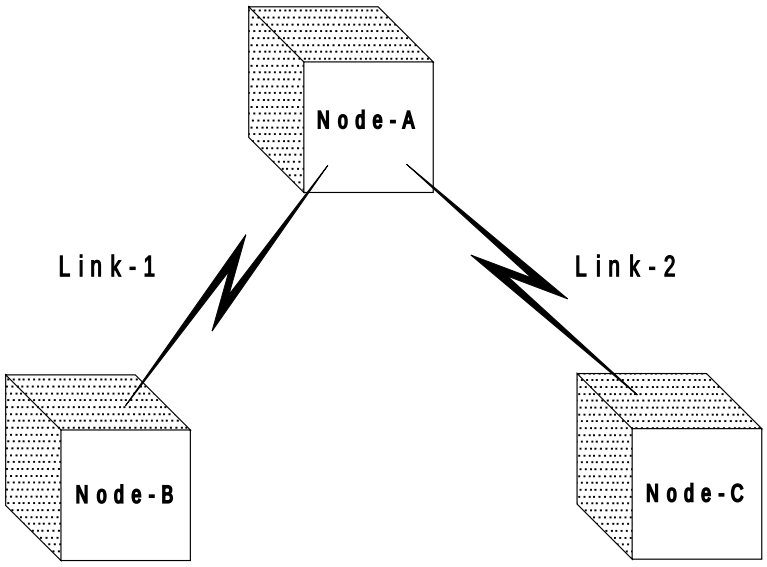

Figure 5. A Simple RSCS Network

Nodes in an RSCS network can be devices, such as workstations and printers, or systems, including VM/ESA and non-VM peer systems. A **peer** is an equivalent system with which data can be mutually exchanged. In a network, nodes are considered to be **local** or **remote**.

A **local** node is the system you are using at that particular time. Because RSCS runs on VM/ESA, its local node is a VM/ESA system. A **remote** node is another system connected to your local system by communications equipment over RSCS-defined links. Remote nodes are either **adjacent** or **nonadjacent**.

An **adjacent** node is one that RSCS can communicate with directly. In other words, RSCS can transfer data to it without going through another node first. Conversely, for **nonadjacent** nodes the communication is indirect; data passes from RSCS to an intermediate (adjacent) node before the nonadjacent node receives it.

Figure 6 on page 7 shows the distinction between remote and local and adjacent and nonadjacent nodes. In Figure 6 on page 7:

- Node-A is the local node
- Node-B, Node-C, and Node-D are remote nodes
- Node-B and Node-C are adjacent to Node A. They are directly connected by Link-1 and Link-2 respectively.
- Node-A and Node-D are nonadjacent nodes that are indirectly connected by Link-2, adjacent Node-C, and Link-3

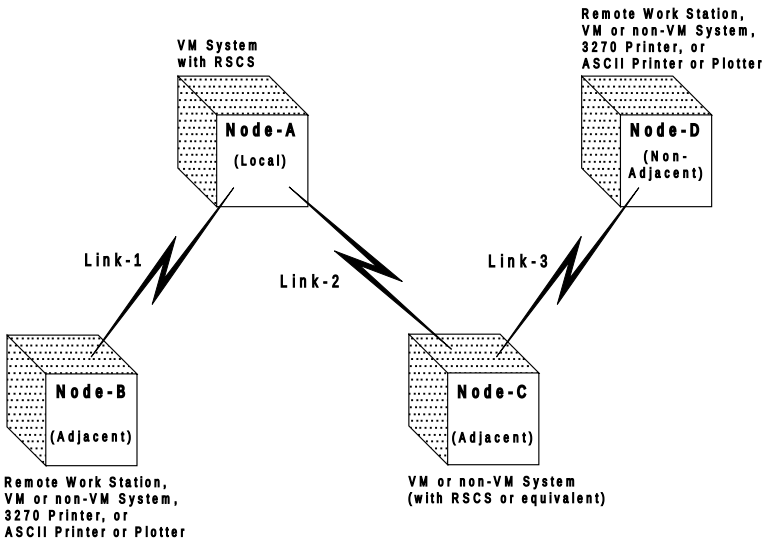

Figure 6. Node Characteristics in a Simple Network

Types of Nodes

A node is either a **system node** or a **station node**. A station node can originate and receive information. It can be either a computer, workstation, or printer. A system node, however, must be a computer. Besides originating and receiving information, system nodes can also *relay* it between two other nodes. If, because of its position in a network, a node relays information, it is sometimes called a **store-and-forward node**.

In Figure 6, Node-C represents a store-and-forward node. It must be a system node because it must relay information between nodes. If a node cannot relay information because of its position in a network, it is sometimes called an **end node**. In Figure 6, both B and D represent end nodes. They can be either system nodes or station nodes.

RSCS can communicate with system nodes that are running under the control of NJE-compatible subsystems. NJE-compatible subsystems include:

- AS/400 Communications Utilities Release 2
- JES2 Release 1.3.3, or later
- JES3 Release 1.3.1, or later
- MVS/BDT (MVS/Bulk Data Transfer) Release 2.1.0, or later.
- RSCS Networking Version 1, Release 3
- RSCS Networking Version 2
- RSCS Networking Version 3 Release 1.0 or later
- VSE/POWER Version 2

RSCS can communicate with station nodes that are:

- ASCII printers or plotters (ASCII is an abbreviation for American National Standard Code for Information Interchange)
- Computers running under the control of a system that can provide Multi-Leaving protocol
- IBM 3270 Information Display System Printers
- Line printer router (LPR) daemons in a TCP/IP network
- LPR clients in a TCP/IP network

- Unsolicated File Transfer (UFT) daemons in a TCP/IP network
- UFT clients in a TCP/IP network
- Workstations running under the control of RJE

Supported Link Types

A link in an RSCS network is a connection between two adjacent nodes. RSCS transfers data between its system and other nodes over links. Each link is associated with a programming routine, called a **driver**, that manages the transmission and reception of files, messages, and commands over the link.

The way that a driver manages the data is called a **protocol**. All file transmission between networking nodes uses NJE protocol. 3270 printers use 3270 data streams, workstations use RJE protocol, and ASCII printers use data streams appropriate to that printer. Systems Network Architecture (SNA) provides one set of protocols that governs communications on links.

The method RSCS uses for sending data to a node varies, depending on the type of connection that is used to establish a link. As the following sections describe, RSCS 3.2 can support non-SNA, SNA, and Transmission Control Protocol/Internet Protocol for VM (TCP/IP) connections.

Non-SNA Connections

Non-SNA connections between RSCS nodes are usually established using binary synchronous communication (BSC) or channel-to-channel (CTC) type lines. These link are associated with virtual line addresses. RSCS has direct control of I/O activity on the link and must respond to all error conditions that occur on the link.

Over non-SNA connections, RSCS communicates directly with adjacent nodes and indirectly with nonadjacent nodes. Direct communication means that RSCS determines where to move data and how to get it there. Indirect communication means that RSCS sends a file to an intermediate system (adjacent node) where it is stored temporarily and then forwarded on to its destination. This destination could be the nonadjacent node or another intermediate system. This method of storing files, between the time they are sent and the time they are received, is called **store-and-forward transmission.**

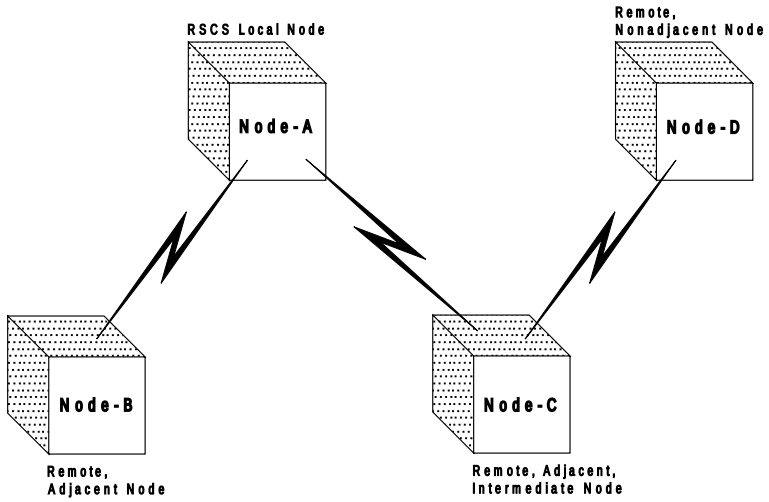

Figure 7. RSCS Communication Across Non-SNA Links in a Network

For example, in Figure 7, RSCS resides on the system at Node-A. From Node-A, RSCS can communicate with Node-B, Node-C, and Node-D. When RSCS communicates with Node-B or Node-C, the communication is direct. Communication with Node-D is indirect. Because the communication is indirect, the store-and-forward transmission method is used. This means that when a file is sent from Node-A to Node-D it temporarily is stored at Node-C. Then, the networking program at Node-C (which may or may not be another RSCS system) forwards the file on to Node-D.

RSCS recognizes the following types of non-SNA links:

ASCII	A link to a remote or local ASCII printer or plotter using an ASCII control unit
GATEWAY	A link using an installation-defined protocol
LISTPROC	A list processor
MRJE	A link to a Multi-leaving Remote Job Entry workstation using a BSC adapter
NJE	A link to a remote Network Job Entry system using a 3088, CTC adapter, ESCON CTC, or BSC adapter
NOTIFY	A note generator
RJE	A link to a Remote Job Entry workstation using a BSC adapter
3270P	A link to a remote or local 3270 Information Display System Printer that is channel-attached or using a BSC adapter

SNA Connections

For SNA connections, RSCS deals with *where* and calls on another product, VTAM, to deal with *how*. SNA connections, which are also called session drivers, are VTAM-controlled devices that are logically connected to RSCS; they perform data transfer on SNA links. VTAM directly processes all I/O and errors that may occur; it then provides RSCS with information about these transactions. These connections are also referred to as LU_LU (logical unit-to-logical unit) sessions.

RSCS communicates directly with both adjacent and nonadjacent nodes through VTAM by means of SNA links. On SNA connections, VTAM takes care of physically moving data between its origin and destination. To accomplish this, VTAM must have an accurate view of the network's physical configuration so that it can select paths for moving the data between nodes. RSCS, however, does not become involved with the physical movement of data on SNA links. While VTAM identifies the paths, RSCS simply identifies the destinations.

Because RSCS and VTAM divide the work involved with data transmission, they have different physical and logical views of nodes connected by SNA links. For example, nodes that appear physically nonadjacent to VTAM seem adjacent to RSCS. As Figure 8 on page 10 shows, through VTAM, RSCS can establish *direct* communications with nodes that are both physically adjacent and physically nonadjacent.

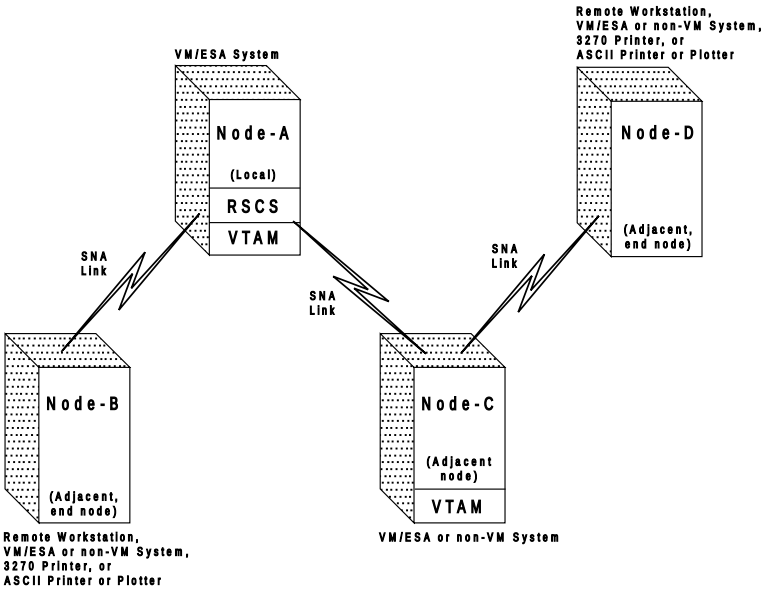

Figure 8. RSCS Communication Across SNA Links in a Network

In Figure 8, RSCS resides on the system at **Node-A**. From Node-A, RSCS can communicate with **Node-B**, **Node-C**, and **Node-D**. If this were a non-SNA network (like the one in Figure 7 on page 8), RSCS would be communicating directly with Node-B and Node-C and indirectly with Node-D. However, this is an SNA network with VTAM residing on Node-A and Node-C. Because of VTAM, RSCS can communicate directly with all the nodes in this network. For example, a file sent from Node-A, destined for Node-D, is transmitted directly to Node-D with VTAM's help. It is not temporarily stored at Node-C as with non-SNA links.

RSCS recognizes the following types of SNA links:

SNANJE	A link to a remote Network Job Entry system
SNARJE	A link to a System/36 Remote Job Entry workstation
SNA3270P	A link to a remote or local 3270 Information Display System Printer

TCP/IP Connections

RSCS 3.2 provides several links that enable RSCS to interact with printers, workstations, or peer NJE nodes within a TCP/IP network. RSCS recognizes the following types of TCP/IP connections:

LPD	A connection that enables RSCS to receive print files from a TCP/IP network for distribution in the RSCS network.
LPR	A connection sends print files to a remote line printer daemon.
TCPASCII	A connection to an ASCII printer or plotter attached to a terminal server.
TCPNJE	A connection to a remote peer NJE system in a TCP/IP network.
TN3270E	A link to a remote 3270 Information Display System Printer that is attached within a TCP/IP network.
UFT	A connection used to send files to a UFT client in a TCP/IP network.
UFTD	A connection that enables RSCS to receive files from a UFT server in a TCP/IP network for distribution in the RSCS network.

These TCP/IP connections use TCP to establish the protocol layer to an IP network; these connections are associated with dotted decimal addresses. This network provides connectivity through routers and gateways to a client, server, or NJE peer system in a TCP/IP network. As Figure 9 on page 11 shows, TCP services the connections to the IP network; it also provides RSCS with information about the flow of data.

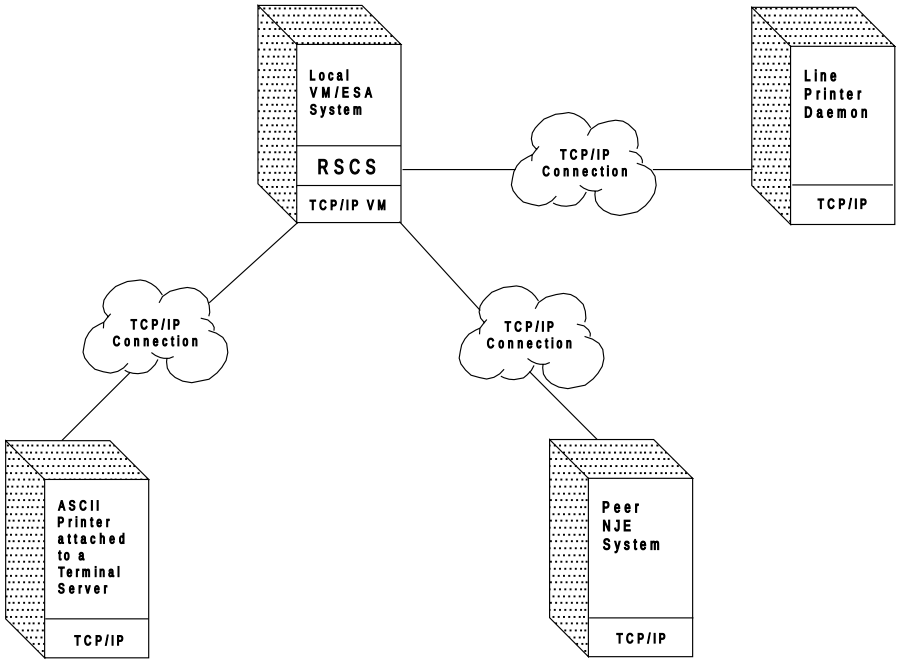

Figure 9. RSCS Communication Across TCP/IP Connections

RSCS Configuration File

The **configuration file** tells RSCS about the network hardware and software arrangement. Statements in the configuration file describe (among other things):

- The name of RSCS's own VM/ESA system (local node)
- The names of links to adjacent nodes
- Communication paths (routes) to nonadjacent nodes through adjacent nodes
- How to communicate with the nodes (based on whether they are 3270 printers, workstations, peer systems or ASCII devices).

RSCS loads the configuration file during its initialization process. RSCS then builds internal tables based on information from the configuration file statements that govern operations at the network. RSCS's initial view of the network, established by the configuration file, can be altered by means of RSCS commands which can add, delete, and redefine links between nodes while RSCS is still running.

After RSCS has been initialized, it knows what your network looks like. If your network should change, you have two ways to tell RSCS about these changes. You could change the configuration file and initialize RSCS again, but if the change is temporary, this is not necessary. Instead, you can alter RSCS's initial view of the network by using certain commands that dynamically add, delete, and redefine links between nodes while RSCS is still running. These commands cause changes in the internal tables that, in turn, let RSCS quickly make changes to the network's

configuration without shutting down and starting up again. These commands are described in the *VM/RSCS: Operation and Use* book.

How RSCS Transfers Files

RSCS transfers files using **link drivers**. While a link driver is active, it handles data transfer between its link and the VM system spool. The link drivers use tag slot information to complete file transfer.

There are three categories of link drivers: **line drivers**, **session drivers**, and **TCP/IP connections**. Table 1 shows the various link drivers and their associated nodes.

Line drivers perform data transfer on non-SNA links. They are called “line” drivers because they handle actual I/O on communications lines. They transmit and receive files over these lines between nodes associated with specific drivers.

Session drivers perform data transfer on SNA links. They are called “session” drivers because they enter into LU_LU (logical unit-to-logical unit) sessions with VTAM. They send and receive files, through VTAM, between nodes associated with specific drivers.

TCP/IP connections perform data transfer to peer NJE nodes and devices that reside in a TCP/IP network. They are called “connections” because the links use TCP to establish the protocol layer to an IP network; these connections are associated with dotted decimal addresses in a TCP/IP network. TCP services the connections to the IP network and provides RSCS with information about the flow of data.

Table 1. Types of Link Drivers and Associated Nodes

Driver Type	Driver Name	Associated Node
line	ASCII	Remote or local ASCII printers
line	GATEWAY	Defined by your installation
line	LISTPROC	None
TCP/IP connection	LPD	Remote line printer router within a TCP/IP network
TCP/IP connection	LPR	Remote line printer daemon within a TCP/IP network
line	MRJE	Remote multi-leaving workstations
line	NJE	Remote (peer) systems
line	NOTIFY	None
line	RJE	Remote workstations
session	SNANJE	Remote (peer) systems using VTAM
session	SNARJE	Remote System/36 RJE workstation using VTAM
session	SNA3270P	Remote or local 3270 printers using VTAM
TCP/IP connection	TCPASCII	Remote ASCII printers in a TCP/IP network
TCP/IP connection	TCPNJE	Remote (peer) systems within a TCP/IP network
TCP/IP connection	TN3270E	Remote 3270 printer attached within a TCP/IP network.
TCP/IP connection	UFT	UFT daemon within a TCP/IP network
TCP/IP connection	UFTD	UFT client within a TCP/IP network
line	3270P	Remote or local 3270 printers

List Processor Links

The list processor link driver, LISTPROC, is an RSCS facility that you can use to process a spool file containing a list of destinations. A list processor enables a file with a distribution list to be transmitted over a network to the destinations listed in the distribution list using the minimum number of copies of the file.

The list processor link driver operates on a specially-defined link. Unlike other types of links, RSCS will not let you delete it once you define it. RSCS also prevents you from defining any other LISTPROC-type links by ignoring the new definition. The existing RSCS configuration file statements and commands define and start the link ID and line driver. Internally, the list processor appears to RSCS as an NJE line driver driving a line wrapped back upon itself. Data is read from the spool by the transmitting side of the line driver and written back into new spool files by the receiving side of the line driver.

Networking Links

The networking link drivers (GATEWAY, LISTPROC, NJE, SNANJE, and TCPNJE) use a special technique called **multistreaming** for transferring files. Multistreaming allows more than one file to be transmitted concurrently over the same link. For example, if files A and B are being transmitted at the same time, a portion of file A will be transmitted, then a portion of file B, followed by another portion of file A. This type of intermixing can improve traffic flow on the network. With multistreaming, small files are not kept waiting while a large file is being transmitted. Both large and small files are transmitted at the same time.

The LISTPROC, NJE, SNANJE, and TCPNJE link drivers support up to 7 transmission streams and the GATEWAY link driver supports up to 32 transmission streams.

Notify Links

The NOTIFY link driver is also a special type of link. It is designed to store files, generate notes, and purge files. In other books in the RSCS library, we show examples of how you can use a NOTIFY link to handle misdirected files in your network. Using the ROUTE configuration file statement or command, you can direct files with unknown user IDs and node IDs to a NOTIFY link. You can define the link to hold files for a specified period of time, create a note reflecting the problem, and purge the file when the period of time has elapsed.

Printer Links

The printer link drivers (ASCII, LPD, LPR, SNA3270P, TCPASCII, TN3270E and 3270P) enable RSCS to communicate with a variety of printer and plotter devices. RSCS can support IBM 3270 Information Display Printers, ASCII printers or plotters, or a print daemon in a TCP/IP network. The SNA3270P-, TN3270E-, and 3270P-type links permit the production of GDDM (Graphical Data Display Manager) Extended 3270 data stream output or IPDS (Intelligent Printer Data Stream) output through GDDM or PSF/VM (Print Services Facility/VM). The ASCII- and TCPASCII-type links send data streams of ASCII characters and control sequences to ASCII printers and plotters. In addition, the LPD- and LPR-type links enable RSCS to send data streams to a line printer daemon and receive them from LPR clients, so that print files can be exchanged between a TCP/IP network and a NJE network.

UFT Links

The UFT- and UFTD- type links enable RSCS to send files to a UFT daemon and receive them from a UFT client allowing for the exchange of files between a TCP/IP network and a NJE network.

Workstation Links

Each workstation link (RJE, MRJE, and SNARJE) enables RSCS to communicate with one remote workstation in the network. The communication protocols RSCS uses varies according to the type of link that is established. RJE-type links use binary synchronous communications (BSC) to communicate with a single workstation. An MRJE-type link uses a multi-leaving protocol that enables it to operate in host mode or in workstation mode. For SNARJE-type links, SNA protocols are used to communicate with a workstation.

VM RSCS Data Interchange Manager

In addition to the various types of links described in the previous sections, RSCS also features the VM RSCS Data Interchange Manager (RSCS Interchange). The RSCS Interchange server, which runs as a separate virtual machine, handles the exchange of mail between users on a VM/ESA system and users in a TCP/IP network environment who use Simple Mail Transfer Protocol (SMTP) mail functions.

SMTP is an electronic mail protocol that supports client (sender) and server (receiver) functions. RSCS Interchange handles mail traffic destined for SMTP from VM/ESA or other NJE systems, and mail destined to VM/ESA or other NJE systems from SMTP. These networks use different mail formats. For example, you can send mail using the CMS NOTE command, PROFS, or OfficeVision. However, RSCS Interchange allows users with different mail formats to communicate with each other, without changing the format of their VM/ESA note or message.

For additional information, see the *RSCS Data Interchange Manager Installing, Managing, and Using* book.

Summary of RSCS Communications

Table 2 summarizes the types of links that RSCS can support in your network. The table includes the type of communication the links enable and the types of nodes they connect.

Table 2 (Page 1 of 3). Summary of RSCS Link Types and Their Roles

Link Category	Function	Link Type	Driver Type	Node and Communication
List Processor	Provides mechanisms to send one file to many destinations using the minimum number of physical transmissions.	LISTPROC	Line	None

Table 2 (Page 2 of 3). Summary of RSCS Link Types and Their Roles

Link Category	Function	Link Type	Driver Type	Node and Communication
Networking	Provides for communication between RSCS and an adjacent NJE-compatible subsystem.	GATEWAY	Line	Allows a connection to a NJE-compatible subsystem using protocols defined by your installation.
		NJE	Line	Each link controls one communication adapter, for a BSC line, a CTCA, ESCON CTCA, or a 3088.
		SNANJE	Session	Allows logical unit, type 0 sessions with SNA systems. A SNANJE link uses a full duplex protocol; both session partners can simultaneously send and receive.
		TCPNJE	TCP/IP connection	Allows communications, using standard TCP/IP sockets, with an NJE peer node that resides in a TCP/IP network.
Notify	Generates and sends notes to users when RSCS finds a problem with a file. If a file cannot be delivered, Notify informs its originator and holds the file until it can be processed. Notify purges the file after a time limit expires.	NOTIFY	Line	None

Table 2 (Page 3 of 3). Summary of RSCS Link Types and Their Roles

Link Category	Function	Link Type	Driver Type	Node and Communication
Printer	Connects a system node and a station node (an IBM 3270 Information Display System printer, an ASCII printer or plotter, or a line printer daemon).	ASCII	Line	Transmits data streams of ASCII characters and control sequences to ASCII printers and plotters connected to a host VM/ESA system.
		LPD	TCP/IP connection	Enables RSCS to receive print data streams from a line printer router in a TCP/IP network
		LPR	TCP/IP connection	Enables RSCS to send print data streams to a line printer daemon in a TCP/IP network.
		SNA3270P	Session	Transmits files from a host VM/ESA system to a 3270 printer through VTAM using types 0, 1, or 3 logical unit protocols.
		TCPASCII	TCP/IP connection	Transmits data streams of ASCII characters and control sequences to ASCII printers and plotters that are attached to a terminal server within a TCP/IP network.
		TN3270E	TCP/IP connection	Transmits files from a host VM/ESA system to a 3270 printer, which is attached within a TCP/IP network.
		3270P	Line	Transmits files from a host VM/ESA system to a 3270 printer, which is locally channel-attached or is remote through a BSC adapter.
UFT	TCP/IP connection using the unsolicated file transfer protocol to send files.	UFT	TCP/IP connection	Enables RSCS to send data streams to a UFT daemon in a TCP/IP network.
UFTD	TCP/IP connection using the unsolicated file transfer protocol to receive files.	UFTD	TCP/IP connection	Enables RSCS to receive data streams from a UFT client in a TCP/IP network.
Workstation	Connects a system node (or host) and a station node that is a workstation or another computer that is acting as a workstation.	MRJE	Line	Provides multi-leaving protocol for a remote workstation or as a remote workstation, using a BSC adapter.
		RJE	Line	Provides BSC line protocol only to a single remote workstation.
		SNARJE	Session	Transmits files between a host VM/ESA system and an SNA System/36 RJE workstation using LU T1 sessions.

Chapter 2. RSCS Environments

RSCS can be used in either a single-system or a multi-system (network) environment. The configuration of a network depends both on the size and the nature of the business.

Single-system Environment

Typically, a small business operates in a single-system network. This network consists of a single, main computer that is centrally located and has a number of local and remote devices connected to it. Remote devices are those connected to the system by communications equipment. Remote connection allows for geographic separation between the system and the device. (They might be on different floors, in different buildings, or in different cities.) Remote users are those who access the system over remote connections.

In a single-system environment, RSCS running under VM/ESA can have output printed on a 3270 Information Display System printer or an ASCII device that several users share (regardless of input method or location).

3270 Printers

3270 printers are those printers that connect to the system by the same type of control unit as 3270 terminals. This group includes models that produce both alphanumeric and graphic (monochrome and color) output.

These printers range from basic function, low-cost models to more costly, extended function models. Basic function, low-cost 3270 printers help businesses provide their employees with timely printed output. Their basic function provides for day-to-day working copy. The low cost makes it feasible for a business to own several and distribute them among individual departments that are located away from the system, near users. Members in or near the department can share the printer and the benefits of having their output printed when and where it is convenient. The more costly, extended function 3270 printers provide for the special needs (graphics output, for example) of a business.

ASCII Devices

RSCS can communicate with ASCII devices connected to an IBM 7171 ASCII Device Attachment Control Unit or a 9370 ASCII Subsystem Controller in transparency mode. RSCS can also communicate with ASCII devices that are attached to a terminal server in a TCP/IP network.

With this RSCS support, user programs can transmit data streams of ASCII characters and control sequences rather than EBCDIC data streams to ASCII printers and plotters. RSCS uses the transparency mode of the controller to allow the user's program to exercise the many features of an ASCII device such as type quality or type size which are not supported by the 3270 emulation provided by the controller.

Sharing Printers

Scheduling and managing real devices is a necessary part of data processing. These are services VM/ESA gives to its users. VM/ESA controls real devices; users do not. Users do not have to wait for real devices to become available. They work as they need to, using virtual devices on their virtual machines.

Along with scheduling and managing real devices, the system provides for the more efficient shared use of those devices. They are not controlled by individual users, but by the system to work for everyone. Sharing is possible, under VM/ESA, because of spooling.

When users “print” output on their virtual printers, for example, they are creating output files which the system stores on the spool. The spool, which has a large capacity for holding data, can hold output files from many different users for the same printer. When the required *real* printer is available, the system selects which file to print next. Only then does the spool file become printed on paper.

The system manages the flow of files to real printers. Users are concerned only with their virtual printers, which are always available to them. They can work continuously, run jobs that produce output at any rate, in any order. The system controls the real printers. It schedules the actual printing of an output file on a real printer such as a 1403, 3800, 4245, or 4248 printer. In this way, many users can share a single real printer.

Sharing 3270 Printers

VM/ESA spooling cannot support 3270 printers; however, RSCS can support these printers. Businesses that want the flexibility and convenience offered by 3270 printers can support them through RSCS.

RSCS supports both remote *and* local 3270 printers and manages them, through the system spool. RSCS extends VM/ESA's spooling capabilities to include 3270 printers.

The system controls other types of printers, but leaves control of 3270 printers to RSCS. Users direct their output for a 3270 printer to RSCS. When they “print” output, the system stores it in a spool file for RSCS. When the required real 3270 printer is available, RSCS selects which file to print next and sends it along to the printer. Only then does the spool file become printed on paper. Figure 10 on page 19 shows several users sharing a single real (RSCS-controlled) 3270 printer.

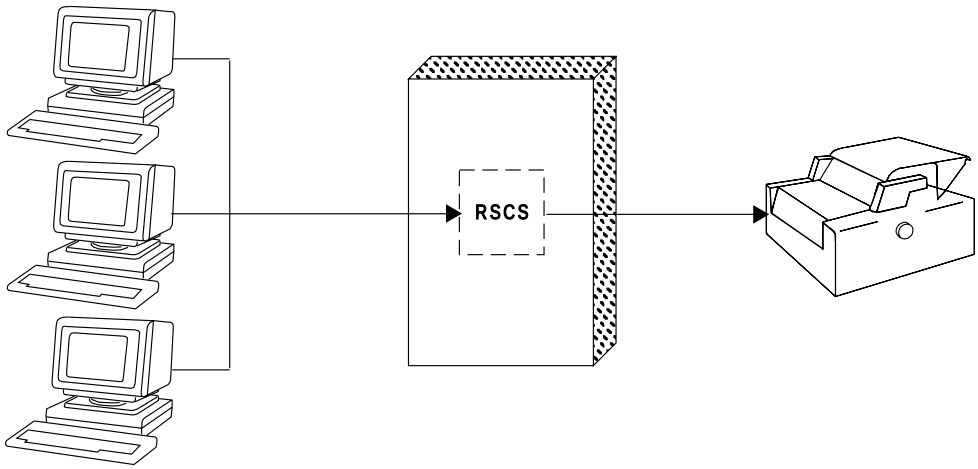

Figure 10. Through RSCS, Several Users Can Share a Single 3270 Printer

Submitting Jobs from Remote Locations

The following sections describe how you can use RSCS to submit different types of jobs from remote locations in your network.

Remote Workstations

In RSCS terms, a remote workstation consists of one or more devices through which users can send input to and receive output from a **host** system. The host is the system used for processing jobs. It may be the same system to which a remote workstation is physically connected. Or, in a multi-system network, the host may be another system in the network. This process is called Remote Job Entry, and so, the remote workstations are often called RJE workstations.

Input and output are in the form of files. The files are created *before* they are sent to or from the remote workstation. Input files can be jobs and data that users have created or programs they have written. Output files can be formatted reports or results from compiling programs.

Using a remote workstation, a user can send an input file to the host system for processing. The host creates an output file as the result of its processing and returns that file to the user. RSCS moves the data from the user to the system and from the system to the user through the remote workstation.

Remote Workstation Location

A remote workstation can be anywhere. It does not have to be near the system, but it can be situated, instead, near its users—where it is convenient for their work. Remote workstations might be located in individual users' offices. So, without leaving their work area users can:

- Create files (or use existing files)
- Send jobs to the system for processing
- Do other work at their desks while jobs are being processed
- Receive output when job processing is done
- Use results immediately for the next phase of their work.

Or, a remote workstation might be centrally located, where it is convenient for several people to share—in the same building and on the same floor with its users.

Advantages of RSCS in a Single-system Environment

In a single-system environment, RSCS offers greater efficiency, productivity, and flexibility.

ASCII printer and plotter support allows users the advantage of using ASCII data streams rather than EBCDIC data streams. RSCS supports these devices through the ASCII- and TCPASCII-type links. This support allows users to take advantage of ASCII printer type size and quality which 3270 emulation does not support.

3270 printers complement rather than replace other (system-controlled) printers. They do work that other printers cannot do, allowing the other printers to be used more efficiently—for high-volume work and high-quality copy.

RSCS 3270 printer support means that:

- Users do not have to go after printouts or have them delivered from some other location. RSCS moves output data to a printer near users, so they can retrieve it as soon as it is printed.
- Neither the system nor individual users need to be involved with managing the shared use of a printer. RSCS can handle several users' output to a single device.

Remote workstations let employees use their company's computer to do their work even if they are geographically removed from it.

- Users do not have to carry (or have someone else carry) data to and from the computer. RSCS handles that—moving the data between users and the system. People stay where they are, where they work. Only the data moves.
- Users can work on the system whenever and as often as needed. Users can get results, make corrections, and try again in the same day. Or, they can get results and move on to the next task.

RSCS gives you the flexibility to align your resources according to the needs of your business. It lets you use devices more efficiently, which saves employees' time and energy for other activities. This means improved productivity in your employees and your business. Figure 11 on page 21 shows RSCS supporting Remote Job Entry and users sharing an ASCII and a 3270 printer.

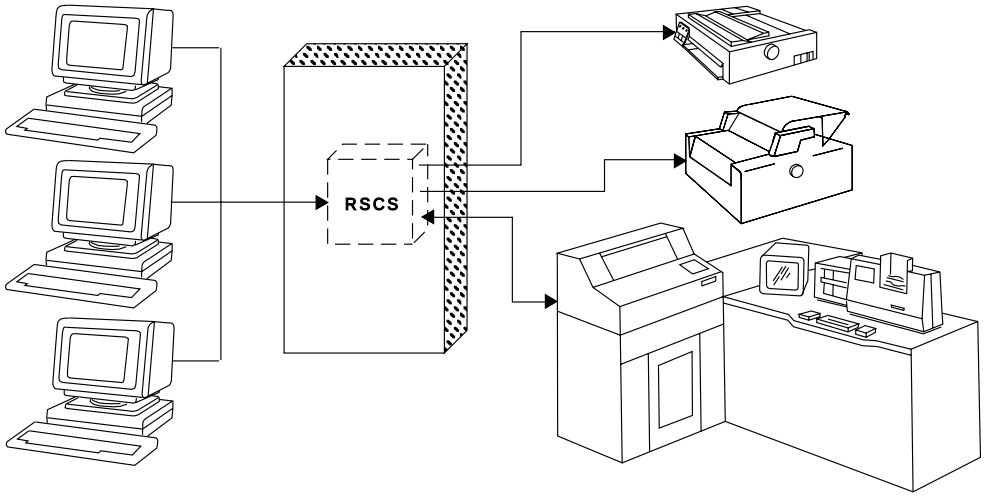

Figure 11. RSCS Support in a Single-system Network. RSCS supports remote job entry, users sharing a 3270 printer, and users sharing an ASCII printer.

Multi-system Environment

A multi-system environment expands the benefits of a single-system environment. Typically, a large business or one that has more than one geographic location operates in a multi-system network. This network consists of two or more systems connected to each other physically (by communications equipment) and logically (in a defined relationship, as peer or host, to other systems).

The systems may be at the same or at different geographic locations. To increase processing power, a business may choose to add a second processor at a site and connect it to their existing processor (rather than to replace their existing processor with one that has a larger processing capacity). Besides being connected to one another, each system in this network may have several local and remote devices connected to it.

In a multi-system network, because the systems are interconnected, data can be moved through and between them—from any system, to any system. RSCS running under VM/ESA (in addition to what it can do in a single-system environment) can do *networking*.

To users, networking means they can:

- Exchange data with users on the same system
- Exchange data with systems and users at other locations
- Send jobs to other systems for processing
- Direct processed output to devices, such as printers and punches, that are connected to another system.

Data Exchanges

Exchange means two-way communication. The *data* RSCS handles falls into two categories — *messages* and *files*. Although messages are usually smaller pieces of data, the main distinction between the two is how RSCS handles them.

Messages are not spooled (stored temporarily); they are sent directly. If a message cannot be sent or its recipient is not there, the message is not delivered. (This is like calling someone on the telephone when the line is busy or there's no answer.) RSCS tells the sender when it cannot deliver the message. RSCS handles commands like messages.

Files are spooled, so they can always be delivered. If a file cannot be sent right away, it is held until it can be sent. When it arrives at its destination, it is placed in a user's virtual reader. So, it is delivered whether or not the recipient is there. (This is like sending someone a letter. It will be delivered to the mailbox even if no one is home.)

RSCS handles user exchanges and operational exchanges. The following sections describe both, starting with the more frequent user exchanges.

User Exchanges

VM/ESA has the facilities to allow its users to exchange messages and files with other users on the *same* system. RSCS allows users to exchange data as easily with users on *other* systems and locations. RSCS moves data from one place to another in either direction—from system to system, user to user.

User exchanges can be business correspondence (electronic mail).

- Users can send data they've created to an associate at another regional office.
- Users can send weekly reports to a supervisor at a central location.

User exchanges can also be used for short reminders and notices (instead of telephone messages).

All of these exchanges can be done between users on different systems and at different locations.

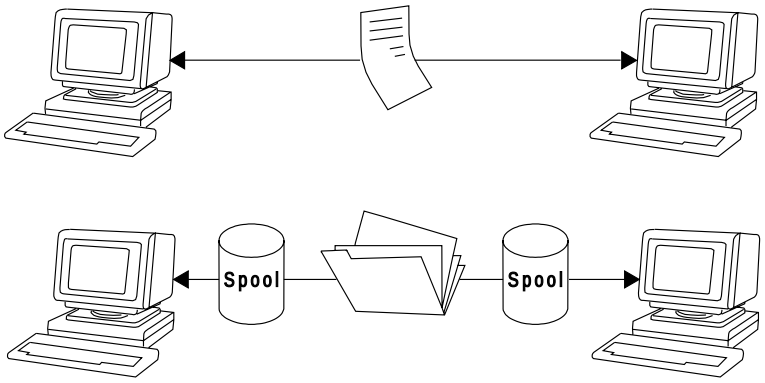

Figure 12. RSCS Allows Users on Different Systems to Exchange Messages and Files

Operational Exchanges

Operational exchanges deal with system operations. VM/ESA provides the Programmable Operator Facility (PROP) which can be programmed to handle a number of routine system operations. Systems can run largely unattended. Sometimes they may need a person for special and manual operations, but they do not need the constant attention of a skilled operator.

RSCS adds the option of geographic separation between system and operator. Using the PROP, a business can have an operator of one system oversee the operation of a second system at a remote location. RSCS sends information on special operations from the PROP on the remote system to the operator and sends the operator's responses back to the remote system. Using RSCS in this way, it is possible for one operator to oversee the operation of several systems, even in different cities or states.

Sending Jobs to Other Places for Processing

Sending jobs to other places for processing is like sending jobs from a remote workstation, only broader in scope. Users can send jobs not only to their own system, but to and from other systems in the network. This makes resources available to users wherever they are in the network. They can take advantage of the following items:

- A special configuration or piece of equipment connected to another system
- Programs or applications that are on other systems
- Greater processing capacity on another system
- Data from common data bases maintained on other systems.

RSCS puts users in touch with the resources they need by moving data to and from the selected resource. The resources of a business are available to more of its employees to help them do their jobs.

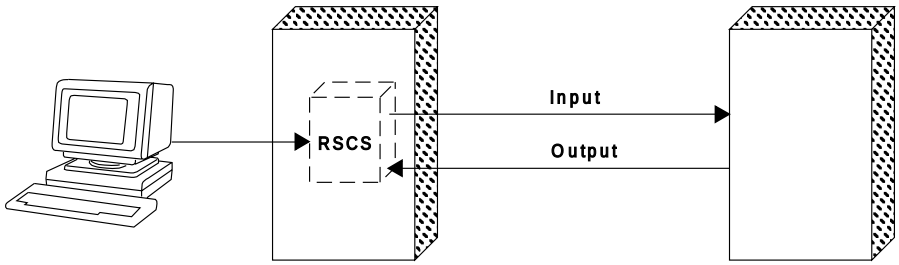

Figure 13. Users Can Send Jobs to Other Systems for Processing

Sending Output to Other Places

The *other places* in this case are other systems (other than the one the user is logged onto) or a device connected to another system. The output may be destined for a printer, a punch, or another user.

When printed output is needed, a user can send the output to a user on another system for printing. That is, instead of printing the output and mailing it, it is mailed first (using RSCS) and printed at its destination.

Or, for another example, a specialized printer in a multi-system network may be connected to another system. RSCS can send the data to the specialized printer on the other system.

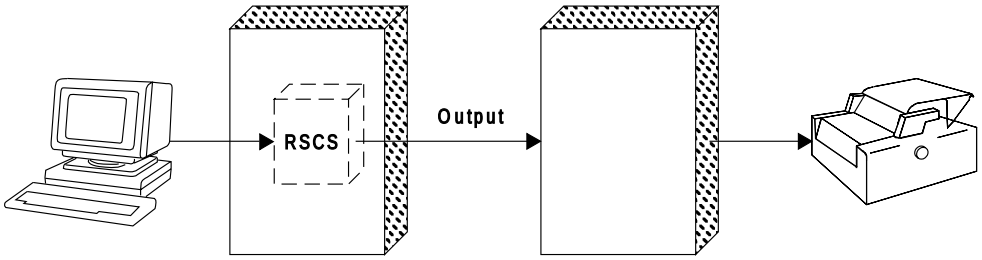

Figure 14. Users Can Send Output Files to Other Systems for Printing

Advantages of RSCS in a Multi-system Environment

In a multi-system environment, RSCS offers greater efficiency, productivity, and flexibility. Earlier, you learned the benefits RSCS offers in a single-system environment through its 3270 printer, remote workstation, or ASCII support. Any system in the network with which RSCS can communicate can be used by your employees.

RSCS gives you many options in using your computing resources. Businesses can distribute computing systems, designed to the specific needs of a given department. Multi-system environments offer:

- Specialized device configuration
- Large processing capacity
- Special set of applications and programs.

Any of your employees can have access to the resources within your business network from their own location. However, you control their access to your resources at a corporate or site level.

Employees can get data they need from other systems. When a piece of work passes from one phase to another, moving as it does from department to department, RSCS can transfer data from one employee to the next, from one system to another system. Employees can:

- Correspond electronically
- Use programs on another system to process their jobs
- Send jobs from a remote workstation at their location to the local or to a remote system
- Direct output to RSCS-controlled 3270 printers or ASCII devices from jobs they have submitted to either local or remote systems
- Send or receive output for other system-controlled printers through RSCS
- Send an output file they have created for another employee to print on that employee's system.

You can buy and locate resources (processors, computer programs, I/O devices) to fit your business needs—by departments or regions. Because these resources can be shared, you can distribute the workload of your business and improve your employees access to these resources. This can lead to greater efficiency and productivity in your business.

Chapter 3. RSCS Users

There are two types of RSCS users: general users and operators. This chapter explains how they use RSCS.

General Users

Most users are not aware that they are using RSCS because they use it indirectly. When a VM/ESA user enters the TELL, NOTE, or SENDFILE commands and the destination is not the local system, CMS uses RSCS to send the data. This is also true for Professional Office System (PROFS) and OfficeVision users. These and other VM/ESA commands that use the network do not require users to know about RSCS in order to use them.

Users can use RSCS commands to transmit messages and to inquire about the status of specific files, links, and systems. Users can also control files that have been given to RSCS to transmit over the network or that RSCS is transmitting to them. The *VM/RSCS: Operation and Use* book tells you how to use these commands.

Operators

There are four types of RSCS operators: console operators, system-authorized alternate operators, link-authorized alternate operators, and remote workstation operators. The operator categories are determined by their authority level and the type of work they do. During the day, one person may perform all four operator roles; if that person is interested in using RSCS as a tool with which to do work, he or she may also be performing the user role. RSCS recognizes the following types of operator authority levels:

RSCS Console Operator
is the person logged on to the RSCS virtual machine; this person controls your local RSCS node. For high-traffic nodes, the RSCS console is constantly receiving messages, making it difficult for a dedicated operator to work. For this reason, in many installations, RSCS is a disconnected virtual machine. Several users may then be designated as system-authorized alternate operators.

A System-Authorized Alternate Operator
has unlimited control over your local RSCS node.

A Link-Authorized Alternate Operator
is authorized to control a single link and the information that passes over that link to and from a single workstation or printer. Link operators usually are also workstation or printer operators but have added capability. Depending on the particular model of workstation or printer in use, its operator may have little or no ability to communicate with RSCS. However, if operators are also link operators, they can easily process commands and have almost as much ability to control the link and the information that passes over it as the local RSCS operator does.

Remote Work Station Operator
controls a remote workstation that is part of the RSCS network. That is, the remote workstation is a separate remote node and is connected to the local node by an RSCS link.

Chapter 4. RSCS Meets Your Business Needs

You can use RSCS to move your data between your computer system and its users, wherever they are. If they are remote from the system, RSCS handles the movement of data between the system and remote resources. It also means that remote (like local) resources can be shared among several users for greater efficiency. RSCS gets the files for the system, then sends them on for processing, either to its own VM/ESA system or to another system. When processing is done, RSCS, if requested, returns the results. RSCS can send output files to printers it controls, to system-controlled printers, or to other systems, as directed by users.

With these functions, RSCS gives you great flexibility in managing data and resources within your network. With support for non-SNA, SNA, and TCP/IP connections, RSCS can meet a wide range of business needs. With all of this, RSCS remains:

- Easy to install
- Easy to operate
- Easy to use.

Easy to Install

Note

- RSCS Version 3 Release 2.0 is preinstalled on VM/ESA Version 2 Release 3.0 or later in a disable state. For more information on installation and tailoring steps, see the RSCS Program Directory.
- TCP/IP is preinstalled on VM/ESA Version 2 Release 3.0 or later in a disable state. For more information on installation and tailoring steps, see the TCP/IP Program Directory.

RSCS is easy to install because:

- It can be properly planned for.

  The *VM/RSCS: Planning and Installation* book, described on page 63, contains information to help you adapt RSCS to your network configuration and business needs.

- The installation procedure is simple to perform.

  The RSCS base code, sample exit routines, and feature code are installed and serviced using the Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E) component of VM/ESA.

  – Verify prerequisites for product installation
  – Install RSCS, with or without replacing an earlier release
  – Migrate to a new release, with or without replacing an earlier release
  – Apply service to a product
  – Delete a product

  The *RSCS Program Directory* contains the specific instructions on how to install RSCS using VMSES/E.

- Choices for customizing the product are clearly defined.

The *VM/RSCS: Exit Customization* book, described on page 64, contains information to help you tailor RSCS to your business needs.

If you will be using SNA connections within your RSCS network, you will need a system programmer (or equivalent) to perform VTAM installation and configuration. Similarly, if you will be using TCP/IP connections, a system programmer should also be available to install and configure TCP/IP for VM on your system.

Easy to Operate

RSCS is designed to run without regular operator intervention. After it is initialized, the RSCS virtual machine can be disconnected so the operator is available to perform other tasks. The initialization tasks can also be automated; a feature in VM/ESA lets virtual machines start automatically as part of the VM/ESA system initialization process.

Sometimes, the operator needs or wants to be involved with operations; for example, he or she may need to check status or make adjustments for performance reasons. To find information about RSCS status, the operator can use the QUERY command. This one command provides an easy way to view any information available to RSCS. The data RSCS returns is presented in easy-to-read columnar format.

RSCS lets you authorize another user, called an alternate operator, to perform operational tasks. This can help you combine system and service machine operations at your site. It also lets you assign users at remote locations to control devices at their locations. This on-site control is not only more practical, but a more efficient way to manage remote operations.

You can automate the operation of your RSCS network to reduce the need for manual intervention and to improve the level of service that it provides to its users. The types of automating possible fall into these categories:

- Automatic link restarts

  You can use an exec that will automatically restart non-SNA links when they deactivate because of a signoff received from the other side of the connection or an unrecoverable error found on a link.

- Packaged RSCS execs for operator usage

  You can provide RSCS execs that you place on any RSCS accessible disk to simplify an operator's work, some examples are:

  – Package complex commands or sequences of commands into one, easy-to-invoke exec

  – Access GCS functions, such as reaccessing disks, that cannot be processed directly from within RSCS

  – Allow signon passwords or phone numbers for links to be defined dynamically in execs rather than requiring the operators to know them.

- Packaged CMS execs for operator usage

  You can provide your operators with tools that can perform several steps in analyzing the state of your network.

- Automatic load balancing across the network

You can use the Slowdown facility to automatically slow down and resume the incoming traffic flow on your networking links based on the size of RSCS's file inventory.

- Service virtual machine that monitors RSCS

  You can automate many time-driven tasks that otherwise you would have to do manually. This allows you the ability to automate such daily tasks as changing shift settings, closing RSCS consoles, and driving exits at prearranged times.

- Monitoring RSCS

  You can write a program to monitor the operation of your RSCS machine. This monitoring operation could periodically check the connectivity of links and tell operators of links that remain inactive for an extended time, check files active on links and inform operators if these files appear stuck, and periodically check the depth of file queues and report any anomalies. These automated monitoring activities could greatly reduce the number of informational messages your operators would receive from RSCS and reduce the need for operators to manually run programs to periodically monitor RSCS.

The *VM/RSCS: Operation and Use* book describes RSCS operation tasks and procedures used in various environments. Its content is described on page 64.

Easy to Use

Many VM/ESA users *use* RSCS without even knowing it. Any time they communicate or send files to users at other locations (using the CMS NOTE, TELL, or SENDFILE commands), they are *using* RSCS indirectly through VM/ESA.

PROFS and OfficeVision users use RSCS to communicate with users at other locations.

Users can enter commands to RSCS directly to do these tasks if they wish. They are also given control over files they own while the files are still in transit. After sending files, users can enter commands to find where the files are in route and even to delete files. The *VM/RSCS: Operation and Use* book describes RSCS user tasks and procedures for various environments.

Some Other Features

RSCS includes mechanisms for problem recovery, both dynamic and operator-initiated. It also has diagnostic aids to help isolate problems with faulty data transmission equipment or procedures. In addition, the RSCS accounting facilities for tracking data transfer are consistent with those used by VM/ESA.

Chapter 5. Customizing RSCS

RSCS is designed to be flexible, but customizing and modifying RSCS is intended for experienced system programmers. System programmers can customize it to accommodate your installation and network needs. Using exit routines and control files, such as the configuration file and the events file, and the destination identifier file, they can set up and tailor the RSCS network. For information about these files, see the *VM/RSCS: Planning and Installation* book.

System programmers can use the exit facility to modify RSCS processing to meet any special functional requirements. The *VM/RSCS: Exit Customization* book describes how to customize RSCS using the exit facility.

Only an experienced system programmer should attempt to write RSCS exit routines. Writing exit routines requires a thorough knowledge of system programming, of RSCS programming conventions, and of RSCS internals. If, without having this knowledge, you attempt to write exit routines, you risk seriously degrading the performance of your system and causing system error.

In certain areas of RSCS processing, you may want to customize RSCS to a greater extent than that made possible by the standard options, RSCS operator commands, RSCS control statements, and RSCS exits. While you are not constrained from altering the RSCS source code, direct alteration can eventually result in problems. For example, you would have to adapt IBM maintenance code before applying it to your altered RSCS source code and, during migration, there would be no simple way of transferring your alterations to a new release. To avoid possible problems, use the RSCS exit facility to its fullest extent; excessive alteration of RSCS code is not recommended.

What Is an Exit?

An **exit** is a procedure outside the main RSCS program that, under certain conditions, receives control from RSCS. An exit consists of an exit point and an exit routine. An **exit point** is a location in RSCS source code from which processing passes to an exit routine. An **exit routine** is the code that you supply to customize standard RSCS processing.

RSCS provides the following categories of installation-wide exits, which you can use to customize specific processing areas. Each exit category can be used to customize different areas of RSCS.

- ASCII printer and plotter exits

  ASCII printer and plotter exits specify the protocol used when an ASCII- or TCPASCII-type link communicates with a particular kind of ASCII device.

- Gateway programming interface link driver

  The Gateway link driver exit and the associated Gateway Programming Interface (GPI) specify how RSCS uses a customer-defined protocol.

- Loadable link driver

  The loadable link driver facility allows customers to write an RSCS link driver without using the GPI.

- LPD

  LPD exits customize data streams that are received from a line printer client in a TCP/IP network.

- LPR exits

  LPR exits build device-specific data streams that are sent to a line printer daemon in a TCP/IP network.

- RSCS exit facility

  The RSCS exit facility allows customers to alter and extend the processing of standard RSCS functions.

- RSCS interchange exits

  RSCS interchange exits tailor the processing of the XCHANGE user ID.

- Transmission algorithms

  Transmission algorithms specify the way that RSCS allocates multiple streams to send files over a networking link.

- UFT

  UFT exits build data streams that are sent to a UFT daemon in a TCP/IP network.

- UFTD

  UFT exits customize data streams that are received from a UFT client in a TCP/IP network.

ASCII Printer and Plotter Exits

The ASCII-type links use the ASCII printer and plotter exits to customize device-specific data streams that are sent to printers and plotters. These devices are connected to RSCS through a controller, such as the IBM 7171 ASCII Device Attachment Control Unit or the 9370 ASCII Subsystem controller.

TCPASCII-type links also use these exits to customize device-specific data streams. However, in this case, the ASCII printer or plotter is attached to a terminal server in a TCP/IP network.

RSCS also supplies sample exit routines for the ASCII printer and plotter exits. Use these samples to choose the appropriate protocol for an ASCII printer or plotter. Or, create your own processing modules, using the sample modules as guides.

ASCII printer and plotter exit routines remain independent of RSCS code. The printer and plotter processing you can modify includes: initialization, TAG record processing for new spool files, output record translation, message handling, attention interrupt processing, and end-of-file processing.

You only need to code ASCII printer and plotter exit routines when your installation uses the ASCII- or TCPASCII-type link driver. If your installation uses one of these link drivers, you must provide the exit routine for the link driver or use one of the sample exit routines that is provided with RSCS for the specific ASCII device. See the *VM/RSCS: Exit Customization* book for more information.

Gateway Programming Interface

The Gateway Programming Interface (GPI) lets you create routines that enable RSCS to communicate with systems using various network protocols. The GPI is made up of GATEWAY-type links, gateway programs, and gateway service macros. Gateway programs, like other exit routines, are external from the rest of the RSCS code. For more information about this area of RSCS customization, see the *VM/RSCS: Exit Customization*.

LPD Exits

The LPD-type link drivers use exits to customize data streams that are received from a LPR client in a TCP/IP network. These exits may also be used to control which node and user ID will receive the data stream.

The LPD exits are called at seven points in the operational cycle of the LPD-type link drivers:

- Control file processing
- Data file processing
- End-of-file processing
- Initialization
- Print command processing
- Print job command processing
- Termination

You only need to code LPD exit routines if your installation uses LPD-type links. If your installation uses an LPD-type link, you must provide an exit routine for the line driver. RSCS also provides sample LPD exit routines, which you can use, or modify as needed, for the LPD-type links defined in your network.

LPR Exits

The LPR-type link drivers use exits to customize data streams that are sent to a line print daemon in a TCP/IP network. These exits may also be used to control which remote host and port in a TCP/IP network will receive the transmission.

The LPR exits are called at six points in the operational cycle of the LPR-type link drivers:

- Control file processing
- End-of-file processing
- Initialization
- Record processing
- TAG record processing for spool files
- Termination

You only need to code LPR exit routines if your installation uses LPR-type links. If your installation uses an LPR-type link, you must provide an exit routine for the line

driver. RSCS also provides sample LPR exit routines, which you can use, or modify as needed, for the LPR-type links defined in your network.

Loadable Link Driver Facility

The loadable link driver facility provides a facility to define additional types of link drivers to RSCS. You must supply the routine for the new type of driver. This routine is also separate from the RSCS code. Like other exit routines, you can use RSCS facilities, such as macros and other routines, within your link driver routine. See the *VM/RSCS: Exit Customization* book for more information.

RSCS Exit Facility

The RSCS exit facility allows users to alter and extend the processing of standard RSCS functions. It is an interface between RSCS and your program modifications. When you use the RSCS exit facility, you can modify certain RSCS processes without directly altering RSCS source code. Your exit routines remain independent of RSCS code, making maintenance and migration to new RSCS releases easier.

Using the RSCS exit facility, you can customize the following areas of RSCS processing:

- Initialization and termination
- Spool file processing
- Network job entry (NJE) header processing
- Separator page output
- Output page accounting
- Commands
- Messages
- Dump processing

The RSCS exit facility can have 256 exit points, which are identified by a number from 0 to 255. RSCS contains 48 IBM-defined exits. The *VM/RSCS: Exit Customization* book identifies each IBM-defined exit point by its number. You do not need to write and use exit routines as part of standard RSCS processing. Use of the RSCS exit facility is optional. If you do not use an exit point, RSCS continues its standard processing.

RSCS Interchange Exits

RSCS Interchange also supports several different types of exits. There are four areas that you can code exit routines for processing of RSCS Interchange.

- Accounting

  This exit lets you create accounting information to audit file traffic on the link.

- Commands

  This exit lets you process commands that are issued to the RSCS Interchange server.

- Format Recognition

  This exit enables you to have the RSCS Interchange server recognize and process additional types of mail.

- Security

  This exit lets you enhance security by restricting the users who are authorized to send mail through RSCS Interchange.

For additional information about these exits, see the *RSCS Data Interchange Manager Installing, Managing, and Using* book.

Transmission Algorithms

The spool manager and networking link driver tasks use transmission algorithms to determine the stream on which a **multistreaming** link sends a file. Multistreaming allows RSCS to send more than one file at a time over a networking link (GATEWAY, LISTPROC, NJE, SNANJE, or TCPNJE). RSCS can send several small files at the same time it sends a large file. Each file is sent concurrently on different transmission streams. This reduces the time small files wait to be sent on a link.

RSCS supplies two transmission algorithms. You do not need to provide additional transmission algorithms. However, as the needs of your installation change, you can modify one of these transmission algorithms or create your own, packaged in a separate load library. For more information about transmission algorithms, see *VM/RSCS: Exit Customization*.

UFT Exits

The UFT-type link drivers use exits to customize data streams that are sent to a UFT daemon in a TCP/IP network. These exits may also be used to control which remote host and user ID in a TCP/IP network will receive the transmission.

The UFT exits are called at six points in the operational cycle of the UFT-type link drivers:

- Command processing
- End-of-file processing
- Initialization
- Record processing
- TAG record processing for spool files
- Termination

You only need to code UFT exit routines if your installation uses UFT-type links. If your installation uses an UFT-type link, you must provide an exit routine for the line driver. RSCS also provides sample UFT exit routines, which you can use, or modify as needed, for the UFT-type links defined in your network.

UFTD Exits

The UFTD-type link drivers use exits to customize data streams that are received from a UFT client in a TCP/IP network. These exits may also be used to control which node and user ID will receive the file.

The UFTD exits are called at six points in the operational cycle of the UFTD-type link drivers:

- Command processing
- Connect processing
- Data processing
- End-of-file processing
- Initialization
- Termination

You only need to code UFTD exit routines if your installation uses UFTD-type links. If your installation uses an UFTD-type link, you must provide an exit routine for the line driver. RSCS also provides sample UFTD exit routines, which you can use, or modify as needed, for the UFTD-type links defined in your network.

Part 2. Features and Requirements

The following sections describe the features and updates available with RSCS. It also contains information about product and customer requirements for using RSCS. This section also describes the other books that are available in the RSCS library.

Chapter 6. RSCS Version 3 Enhancements

The following sections describe the major enhancements offered by each release of RSCS Version 3. Review the following table to determine which section, or sections, you should review if you are unfamiliar with RSCS or if you are migrating from a previous release of RSCS.

Migrating from RSCS	Review Section
Before Version 3 Release 1.0	"Version 3 Release 1.0 Enhancements" "Version 3 Release 1.1 Enhancements" on page 54 "Version 3 Release 2 Enhancements" on page 54
Version 3 Release 1.0	"Version 3 Release 1.1 Enhancements" on page 54 "Version 3 Release 2 Enhancements" on page 54
Version 3 Release 1.1	"Version 3 Release 2 Enhancements" on page 54

The *VM/RSCS: Planning and Installation* book contains additional details about migration considerations for RSCS.

Version 3 Release 1.0 Enhancements

This section describes the improvements made in RSCS Version 3 Release 1.0 (RSCS 3.1.0) and how Version 3 compares to RSCS Version 2 Release 3 (RSCS 2.3). The improvements are categorized as follows:

- Routing

  Improved routing mechanisms increase traffic throughput in usual situations and decrease rerouting in problem situations. For more details, see "Routing Improvements" on page 40.

- Performance

  Performance enhancements improve RSCS efficiency and speed by allowing more files to transfer in less time. This greater speed increases user efficiency through faster data availability. For more information, see "Performance Improvements" on page 42.

- Usability

  Usability enhancements give general RSCS users more control over their own files. For more details, see "Usability Improvements" on page 44.

- Operation

  Operation enhancements make authorized RSCS operator intervention easier. For more information, see "Operation Improvements" on page 47.

- Customization

  New exit facilities make RSCS more adaptable and flexible, giving you more control over your data. For more details, see "Customization Improvements" on page 49.

- Programming

  Additional enhancements that can assist your system programmer are described in "Programming Improvements" on page 51.

Routing Improvements

Improved routing mechanisms allow your programmers to set up routes to increase traffic through-put in usual and problem situations. In problem situations, these improved routing mechanisms greatly reduce or eliminate the need to reroute files. The routing mechanism has been revised to support the following features:

- Multiple links that you can use to route traffic to remote nodes.
- Single alternate links that you can define for routing traffic to a remote node.
- Nodes that you can group for common routing purposes.
- Multi-destination files, produced by the list processor, that can be transmitted more efficiently in an SNA network.
- Multiple links between any two systems are now allowed.

Multiple Primary Links

RSCS 3.1.0 supports two or more primary links that you can use for routing traffic to a remote node. All links can send traffic to that remote node at once. If any one of the links is down, RSCS distributes the traffic on the other links. This support provides:

- Intervention-free backup routing
- Intervention-free load balancing; traffic will preferentially send on links that are running faster
- Increased capacity, because more than one link can be used to send traffic.

Alternate Link

You may define a single alternate link for routing traffic to a remote node. RSCS will use this alternate link for traffic for that node *only* if none of the primary links are connected. This further adds to the benefits provided by the support for multiple primary links.

Routing Groups

You can group and route nodes in a common way. For instance, you might form all nodes in one site into a single group. Groups can be further grouped into higher-level groups – perhaps a city, state, region, or country. The highest-level groups are routed through primary and alternate links. The grouping mechanism provides the following benefits:

- It makes setting up routes easier.
- In the few cases where operator intervention *is* required, it can be done on a group-by-group basis.
- Statistical information is kept and can be queried on a group by group basis.

Multi-Destination Files in an SNA Network

Support has been added to improve the way RSCS fans out multi-destination files. In Version 2, copies of a file that were bound for destinations routed through a common link were routed identically. Therefore, the minimum number of RSCS links was used to route traffic. In an SNA network, this is not the ideal procedure, since different *RSCS links* may share the same VTAM-managed *telecommunications lines*. Support has been added to indicate which links share telecommunications lines and, therefore, minimize the amount of physical traffic generated by multi-copy files.

Multiple Links Between Two Systems

It is no longer necessary for the node to which a link connects to sign on with the same name as the link. Therefore, it is possible for you to define multiple links between two nodes. There can be any number of non-SNA links and at most one SNA link.

Relationship to Version 2 Backup Routing

Important Note for Users

The concept of routing has changed between RSCS Version 2 and RSCS Version 3. If you are migrating from RSCS Version 2, please be sure to read this entire section on routing and the next section entitled "Effects Throughout RSCS." Also, see the description of the ROUTE statement in the *VM/RSCS: Planning and Installation* book or the description of the ROUTE command in the *VM/RSCS: Operation and Use* book.

In RSCS Version 2, you could define a backup route only for a node to which there was a direct link. RSCS 3.1.0 alternate routing improves on this situation in several ways:

- The use of multiple primary routes provides you with not only *backup* routing (that is, one link starts when another is inactive), but also with *load balancing*. When two or more links can share the traffic to a single destination as they are able, there is load balancing. Backup routing is simply an extreme case of load balancing.
- In Version 2, the backup link would start only when the primary link was inactive. In RSCS 3.1.0, it starts when none of the primary links is *connected*. This eliminates a situation in the Version 2 backup routing scheme, where a primary link could be *active*, but not *connect*, for a long time. When in that state, no traffic could transmit and the backup link, even if defined, would not start.
- Since multiple primary and alternate links are specified for destination *nodes*, not for links, it is now possible to provide backup routing for non-adjacent nodes and adjacent ones.
- The provision of routing groups lets you reroute many nodes using a single command. This is an improvement over Version 2 where, if an intermediate node should fail, every node routed *through* it had to be rerouted individually.
- The change in concept from backup for a link, to routing for a node, has created a slight incompatibility between Version 2 and RSCS 3.1.0 configuration files. These old statements will be accepted by RSCS 3.1.0, but they will probably not have the desired effect. Before migrating to Version 3, find and replace with equivalent Version 3 statements, or delete any Version 2-style backup routes.

Effects Throughout RSCS

The revision of the routing mechanism has brought about two major changes:

- Routing is now done *entirely* based on destination node; whether a node has the same name as a link is now no longer relevant.
- A file is now no longer associated directly with a link while it is waiting to be sent.

These changes have other affects throughout RSCS, including:

- The QUERY SYSTEM ROUTES command has been replaced with QUERY SYSTEM NODES and QUERY SYSTEM GROUPS. QUERY commands for individual nodes and groups have also been added.
- Files are represented internally not only by tag queue elements, but by shadows. Each file has a single tag but one shadow for every link on which it is enqueued.
- Besides the QUERY FILE command, there is now also a QUERY QUEUE command. The QUERY FILE scans through tags and will tell you everything about a file except the link it is enqueued on and its position in the queue; QUERY QUEUES scans through shadows and may produce more than one answer per file. It will tell you the link a shadow is enqueued on and its position in the queue.
- Exit routines that are related to queueing control will probably need to be rewritten for use with RSCS 3.1.0. To highlight the differences associated with the new routing mechanisms to exit writers, names of relevant RSCS internal subroutines have been changed so that exits that the use them will not assemble correctly. This serves as a warning and will enable exit writers to recode their exits appropriately.

Performance Improvements

The performance enhancements improve RSCS's speed and efficiency while sending files under usual circumstances (steady-state operation), during initialization, and when performing events, such as link activation or deactivation.

File Transmission

In RSCS Version 2, at least five telecommunications buffers were needed to send a single file, even if that file was much smaller than an individual buffer. In RSCS 3.1.0, slight protocol changes have been introduced to allow a file to transmit in a single buffer. In fact, if multi-streaming is in effect and several small files are eligible to transmit, more than one file can be fit inside a single buffer.

This support substantially reduces the number of I/O operations or VTAM requests issued by RSCS and, therefore, directly reduces processor utilization. It also increases file throughput because up to four line *turn-arounds* can be eliminated from a file transmission. This is particularly effective on medium- or low-speed, long-distance, lines.

The protocol improvements must be carried out by the sending and receiving nodes. If only the transmitter does the improvements, a slight increase in performance can be obtained. If only the receiver does the improvements, no improvement can be obtained.

Command and Message Transmission

RSCS Version 2 transmitted commands and messages as priority traffic, but the NJE protocol required them to go in a buffer of their own. This meant that a small amount of command and message traffic could have a disproportionate impact on file throughput.

These protocol enhancements also enable RSCS to send commands and messages in the same buffer as file traffic. This reduces the number of I/O or

VTAM requests and, therefore, reduces processor utilization. It reduces line utilization and cuts down the average transmission time for both files and messages.

Like the file improvements, the command and message improvements rely on protocol enhancements. Throughput improvements for commands and messages are greatest when both nodes (that is, the node sending the file and the node receiving the file) use these enhancements.

However, RSCS 3.1.0 can also communicate with nodes running different levels of networking programs. As such, RSCS 3.1.0 also provides a facility to delay command and message transmission for a limited time. This facility reduces the impact on file throughput that command and message traffic can cause; however, command and message traffic in the network may be delayed.

Multi-Streaming Scheduler

RSCS Version 2 was the first version of RSCS to introduce multi-streaming support. This provided improvements in average file delivery times in the case where, for instance, a large file would be transmitting for some time but a small file arrived for transmission while the larger one was being sent. Without multi-streaming, the small file would have to wait until the larger one had finished.

To improve multi-streaming, RSCS 3.1.0 has implemented an advanced mechanism for selecting the order in which streams are transmitted. This mechanism usually results in file delivery sooner than RSCS Version 2.

Reorder Process

The process by which RSCS reorders its file queue in response to various events has been completely rewritten:

- A different method is used to implement the reorder caused by different events, which eliminates unnecessary work on a reorder.
- Reorders for link activation and deactivation, the most common cause of reorders, are now particularly efficient–they cost virtually nothing.
- The process of enqueueing and removing a shadow from a link has been made much more efficient.

These enhancements affect all RSCS installations. If your installation has small file inventories, queue handling is more efficient. For installations that handle large volumes of data, RSCS 3.1.0 is able to process much larger file inventories. RSCS's handling of the reorder, enqueueing, and removal processes no longer limits overall system throughput.

Working Set Size

Storage allocation mechanisms have been changed so that similar data areas are allocated close together in virtual storage. In particular, allocation for tags and shadows, which are constantly being allocated and deallocated as RSCS operates, is contiguous.

This minimizes the number of page faults RSCS experiences, which contributes to greater RSCS throughput and reduces its working set size. It also results in greater system throughput on the processor on which RSCS is running, since more real storage and more processor time are available for other virtual machines.

Quick Storage Allocation

RSCS now carries out its own storage allocator, which allows storage to be set apart in regions allocated to control blocks of equal size. Use of 4k page alignment, elimination of unnecessary supervisor calls, and use of faster algorithms that work when allocating equal-sized areas of storage, make this scheme far more efficient than direct use of GETMAIN and FREEMAIN.

Message Subscriptions

Message subscription requests are handled in a direct way by chaining them from the link table to which they apply. It is thus no longer necessary to scan the entire AUTH table whenever a message is issued.

Reroute Table Scanning Mechanism

The reroute table scanning mechanism has been changed from a sequential scan to a doubly-indexed lookup. Besides increasing the usability of the reroute mechanism, it greatly enhances its performance and increases the number of reroutes that can effectively be in effect at any time.

Compression and Decompression on Networking Links

The routines for compressing and decompressing data as it is transmitted and received have been optimized. This enhancement significantly improves RSCS performance, as these routines process every transmission buffer.

Direct Pointers Instead of Symbolic References

In RSCS Version 2, many data areas contained a *symbolic reference* to a data area of another type. For example, a port table entry would contain the name of the link it was currently being used by. To look up the associated link table, a subroutine would have to be called to scan the link table to find a matching link.

In RSCS 3.1.0, such references have been replaced by direct pointers (for example, the link table address is contained in the port table entry) so that only a single load instruction is required.

Message Formatting

The message builder in RSCS 2.3 introduced National Language Support but, much processing was required to allow for multiple national languages, honoring message subscriptions, and interaction with exits. The RSCS 3.1.0 message builder performs these and other functions; it has also been redesigned to more efficiently build the text of each message. Improved coding interfaces to the message builder also result in further efficiency improvements.

Usability Improvements

The usability enhancements continue to enhance RSCS's ability to delegate as much power as possible to the general user. Without any special authorization, users can query many aspects of an RSCS system and have full control of their own files. This support can greatly reduce the necessity for authorized intervention. In turn, authorized system operators also benefit from this support as they rarely need to intervene to solve a user's problem.

Improved Query Command

The QUERY command has been completely rewritten. All the Version 2 forms of the QUERY command still exist (except for QUERY SYSTEM ROUTES), but most have been greatly enhanced. These improvements enable any RSCS user to display information about many features of RSCS operation, including:

- Options set by the OPTIONS statement
- Local time zone
- Link start parameters
- Counts of files sending, receiving, queued, held, and looping, by link and by group.

The QUERY command has been improved in the following ways:

- The QUERY command now covers many more basic types of query.
- For queries that refer to complete RSCS tables, the concept of *filters* has been introduced to limit the number of objects that can be displayed from that table. For example, a QUERY SYSTEM LINKS command filter can let you display only the SNA links in starting status.
- The information displayed about any object that passes the filters can be selected at will from several *columns* and is displayed in an easy-to-read columnar format on the screen, with headings on each column and only data in the display itself.
- Commands can self-propagate through the network so that a query about the path to a destination node, or files traveling to a particular node, can be automatically processed on every node. From the node on which the command was first entered to the destination node, the command need only be typed once.

  Self-propagating queries propagate on all the links that can be used to reach a destination node. Although this means that a self-propagating query can reach a particular node by multiple routes, mechanisms have been carried out to ensure it only processes once on any node.

An additional feature of the QUERY command is that QUERY responses never include the message number. This makes them easier to read and gives an installation more freedom in choosing whether to have RSCS produce the numbers for other messages. Often, these message numbers can provide useful diagnostic information in difficult problem determination circumstances.

Columnar Messages

Responses to the QUERY command, and many other commands that give many lines of output, now use columnar messages. This message format keeps all heading text at the top and right side, and display the actual data in neatly aligned columns. This makes these displays much easier to read.

Command Response Interface

A Command Response Interface (CRI) has been designed to enable execs to enter commands to one or more RSCS machines simultaneously. Because the end of the command responses are clearly identified, you can easily determine the responses from each message. You can also use the *language-independent* message format, which is easy for execs to parse and removes a dependency on the national language used for other messages.

The CRI offers the following benefits to users:

- It makes it easy for installations to write and install execs on their systems that provide the users with facilities for querying RSCS and manipulating their files. This means the users do not need to learn the details of RSCS command formats.
- Such execs need never wait until a timeout period has expired before deciding that all the responses to a command have been received. They can use the end-of-command response marker and so provide a quicker response to their users.
- Because execs can be written in a language-independent way using the CRI, installations can feel freer to use the users' own natural language for RSCS's usual messages and responses.

National Language Support

The National Language Support (NLS) introduced in RSCS 2.3 has been extended in RSCS 3.1.0. All messages are now translatable. The RSCS 2.3 message repository has been split into a *conversion repository* and a *translation repository*, which only needs to be translated to provide support for a new national language. This makes the job of translators much easier.

The RSCS 2.3 MFORMAT EXEC has been succeeded by two compilers: one for the conversion repository and another for the translation repository. Compiler performance and diagnostics are improved, and the syntax of the repositories is more friendly.

The CRI makes NLS usage more attractive to individual installations because it makes it possible to write execs that should work on any RSCS 3.1.0 installation, regardless of the language used to enter messages.

In addition to messages, the notes produced by NOTIFY-type links are also included in national language support.

User Control of Files Destined to Them

Users have been able to control files they originate since RSCS Version 2. In RSCS 3.1.0, they can also control files directed to them. This control covers the PURGE, FLUSH, CHANGE, and TRANSFER commands.

Better Information on Active Files

Queries for active files now cover both sending and receiving files. The progress of a file is indicated by the number of blocks that have *been* transmitted or received, not the number that remain. This removes a source of possible error from the messages, especially for files whose total record count cannot be known in advance.

Notification of Misdirected Files

In RSCS Version 2, misdirected files – addressed to unknown nodes or user IDs – would either be transferred back to the user who originated them, purged, or spooled to RSCS's punch or print output queue on a destination system. In the latter cases, the originating user was very rarely aware of what had happened, and the presence of the file on RSCS's output queues wasted spool resource on the destination node.

In RSCS 3.1.0, a NOTIFY link driver has been added to hold files, generate notes, and purge files. You can implement NOTIFY-type links to act as misdirected file handlers. See the *VM/RSCS: Planning and Installation* book for more information about setting up NOTIFY-type links.

Detection of Looping Traffic

In large and complex networks, routing loops can easily arise. RSCS 3.1.0 can detect routing loops:

- For commands and messages, immediate routing loops can be detected, where node **A** routes traffic to node **B**, and **B** routes it to **A** again.
- For files, immediate loops can be detected. Also, nonimmediate loops are detected using a **hop-count** mechanism where the number of links a file has traversed reaches an improbably large value.

Looping files are held in *looping* status and can be released by system operators when the routing problem has been fixed. Looping commands and messages are flushed. Originators of looping traffic are informed; so are system operators in certain instances.

Improved CPQUERY Command

The CPQUERY command has been improved to display, at a glance, the number of spool files owned by RSCS. It can also now distinguish between a user who is not logged on and one who is not in the CP directory.

Friendlier Command Formats

Command keywords can now be up to 16 characters long, which allows them to be much more significant. Because of abbreviation support, it is rarely necessary to type the entire keyword.

Operation Improvements

The operation improvements and the others described in this section, combined with the routing and usability enhancements already described, contribute to the ease of operation of RSCS 3.1.0. Operation enhancements:

- Reduce the need to intervene in RSCS's operation when problems arise.
- Give improved notification of error or problem situations
- Make intervention, when necessary, much easier
- Automate routine tasks.

Slowdown Mechanism

In Version 2, operational problems often resulted from RSCS's spool filling up. This would cause files on neighboring systems to go into hold status and possibly even propagate the problem to those systems. Very complex operator intervention was required to recover from such situations.

RSCS 3.1.0 introduces the *slowdown* mechanism. With this support, an RSCS system nearing its spool limit can alert neighboring systems and prevent them from sending files. The mechanism can be controlled so that less critical links stop receiving traffic before more critical ones do. When spool utilization becomes acceptably low again, RSCS informs neighboring systems that they can resume

sending files. This support requires NJE protocol enhancements to be carried out on both sides (local and remote nodes) of a link.

Stopping Incoming Files

The same protocol enhancements that carry out slowdown can be used manually to prevent a neighboring system from sending files. The HOLD command has been enhanced to cover both outgoing and incoming files.

Event Scheduler

The RSCS event scheduler enables you to provide different service levels at different times of day. For example, large files might be prevented from transmitting during prime shift or console logs might be closed every 15 minutes.

The scheduler can schedule execution of RSCS commands, posting of event control blocks (ECBs), or execution of internal subroutines. It can schedule on-off or repeating events, which can be programmed according to day of week or local or national holidays. They can be controlled by an *events file*, which is read at initialization time, or a by command while RSCS is running.

Improved Recovery Facilities

Recovery facilities have been generally improved:

- Only one restart exec, called RECOVER, is now needed for all non-SNA links. It receives the name of the failed link as a parameter.
- The retry facility has been improved. When a link deactivates, the first retry now takes place within a minute.
- It is possible to have RSCS automatically restart itself after a system abend. This required manual intervention in RSCS Version 2.
- The SHUTDOWN command now lets you specify a restart immediately after RSCS has shut down, eliminating the need to force and autolog RSCS.

Message Subscription

Automated operation is made much easier by improved facilities to subscribe to RSCS messages:

- The SET command has been enhanced. In Version 2, link operators could subscribe to messages for their link, and full operators could subscribe to every message. Now, full operators can choose to subscribe only to messages on a link.
- A new SETMSG command enables full operators to subscribe to messages by number rather than by link.

  Operators cannot subscribe to private messages, such as those issued when users enter the TELL command.
- Both subscription types can be specified in the configuration file.
- CRI options can be used, including language-independence, in messages resulting from subscriptions.

These improvements enable operators and system programmers to subscribe to only the messages they are interested in, and to write automation packages to react to messages appropriately.

CHANGE, PURGE, and TRANSFER Commands

These commands have been enhanced to cover all files on a link, files of a given class, files for a particular destination, or looping files, as well as files identified by spool ID as in Version 2.

Better Treatment of Link Authorized Operators

Some commands have different syntaxes according to whether the originator is a link authorized operator (authority over any file on the link concerned) or a general user (authority over only their own files).

In Version 2, link authorized operators are not able to control their own files on any link other than the one for which they are authorized. In RSCS 3.1.0, they can choose whether to enter the operator syntax or the general user syntax. If they enter the operator syntax, they must refer to the link for which they are authorized. If they enter the general user syntax, they have the same control over their own files as any other general user does.

CP Command Always Gives Return Code

Many CP commands give no console output, especially when running (as most RSCS systems do) with IMSG OFF. In RSCS 3.1.0, the CP command (for CP-authorized operators) now always completes with a return code message so you can assess the success of the command.

Ordered Files Stay Ordered

In RSCS Version 2, an ordered file would go to the top of a link's queue but could be removed from that position by a new file arriving, by any kind of reorder, or by a re-start of RSCS. In RSCS 3.1.0, an ordered file stays at the top of a link's queue, even after a re-start.

Customization Improvements

RSCS 3.1.0 introduces additional ways to customize RSCS. This enhanced customization support provides the following benefits:

- Eliminates the need to modify RSCS source code
- Simplifies the programming involved in such customization
- Simplifies the writing of portable *packages* that can be developed at one installation and used at other installations.

At the same time, many of the reasons that Version 2 installations needed to write customization have been eliminated by RSCS 3.1.0 product features.

Improved Exit Facility

RSCS 3.1.0 has doubled the number of IBM-defined exit points that are available to customers. Also, the functions on many of the Version 2 exits have been enhanced. New and improved exits include facilities to:

- Implement complete security systems
- Implement new commands
- Implement shift management
- Easily inspect and change NJE headers
- Inspect the individual records of files being received
- Add installation-defined link parameters
- Control dump processing after abends

- Edit notes produced by NOTIFY drivers
- Use NOTIFY-type links to process or reject misdirected files
- Log, suppress, and select language for messages
- Effect reroutes according to criteria that cannot be specified with the reroute command or statement
- Efficiently implement multiple RSCS virtual machines on a single processor.

Gateway Programming Interface

A Gateway Programming Interface (GPI) has been defined which, given a single exit routine, can act as an NJE link over another medium – for example, TCP/IP.

Loadable Link Drivers

RSCS supports many different types of links that support a variety of devices and needs. However, if your installation has other link requirements, RSCS 3.1.0 can also support new link types that are defined using a LINKTYPE statement. These links can now be loaded from their own load library instead of being linked into RSCS.

Loadable Transmission Algorithms

Transmission algorithm 1 has been improved, but if you need your own transmission algorithm (for example, to work with a shift manager), you do not need to modify the RSCS load library to implement it. RSCS 3.1.0 supports transmission algorithms, which can be referred to by name and loaded when the link initializes.

More Routines Available from CRV

Programming RSCS exits must still be done in basic assembler language (BAL). However, to simplify the writing of exit routines, more routines, queue anchors, and ECBs have been added to the Common Routines Vector (CRV). The CRV, which is available to every exit, refers to the following type of RSCS routines:

- Command processing entry points
- Command element queue anchors
- Node, link, and group lookup routines
- NETDATA formatting
- Event manager
- Utility routines, including:
  - Generalized hashing
  - Module loading
  - Token-by-token parsing
  - Quick storage allocation
  - Format and enter messages.

Improved Macros

The following programming macros also have been improved:

- RMOD provides a mechanism to specify how later RCALLs will work.
- RCALL can find a selected routine in the CRV table before calling it. See the *VM/RSCS: Exit Customization* book for information about the CRV routines.
- RENTRY has an improved save area chaining mechanism that causes an immediate abend when a save area chain is exhausted. In Version 2, no immediate abend would result, but pointers would become corrupted resulting, later, in extremely difficult-to-diagnose problems.
- RMSG has been introduced to make issuing messages easier.

Packaging Facilities

Enhancements to the EXIT command and configuration file statement, and provision of the imbedding mechanism in the configuration file, make it easier to install an exit package that was not locally written.

Programming Improvements

The enhancements in this section generally benefit systems programmers, but other RSCS users can also benefit from these improvements.

Configuration File Improvements

The configuration file improvements mentioned here are in addition to the others described above. They make it easier to write, read and maintain configuration files, and provide a mechanism for protecting your RSCS system against configuration file errors.

Format Improvements: The general format of the configuration file has been improved:

- REXX-style comments enclosed in /* */ are allowed on any line.
- Statement text can be continued onto other lines using a # character.
- Additional file segments can be imbedded in the RSCS configuration using an IMBED statement. Multiple levels of imbeds are allowed, and imbedded files can be referred to by file ID or FILEDEF name.

LINKDEFINE Statement: A LINKDEFINE statement, which is more flexible than the old LINK statement, has been added to RSCS 3.1.0. The LINKDEFINE statement lets you specify keywords rather than positional parameters and lets you specify more than one LINKDEFINE statement per link ID.

Error Tolerance: The new TOLERANCE statement lets you designate certain sections of the configuration file as critical. If any errors arise while these sections are being processed, RSCS will not start. This prevents errors in RSCS configuration files from causing service problems in a network.

DEST Statement and DEST File: The DEST file is no longer necessary; a DEST statement has been added to the configuration file.

OPTION Statement: Several more option names have been added to the OPTION statement. To make it more usable, multiple OPTION statements can be specified in the configuration file, with different keywords on each one.

On-Line Statistics

A new QUERY option gives, for each link, detailed online measurements of traffic volumes, I/O times, and error codes.

Improved Diagnostics

The improved diagnostic facilities described in the following sections make RSCS problem determination much easier.

Finding Out Storage Locations: A query command has been added to indicate the address at which RSCS is loaded and the address of its CVT. System query commands all have an option to give the base address of the data areas they are querying, so that the in-storage form of the data areas can easily be displayed by those who have RSCS CP authority.

Link Tracing: The facilities for tracing I/O and SNA activity on links have been enhanced:

- In Version 2, it was possible to trace only the data actually sent and received on a link. In RSCS 3.1.0, NJE records can be traced without the extra layer of protocol required for transmission.
- Tracing of failed link start-ups has been considerably simplified by adding the ability to specify, in the configuration file, a default destination for link traces.

Internal Trace: The Version 2 GTRACE mechanism has been replaced by a comprehensive internal trace that uses a wrap-around trace table to trace various events. RSCS can now wrap the trace round in storage, can dump it to a dump user ID, or show every record using the GTRACE facility.

Events that can be traced, include: RSCS commands, diagnose instructions, exit points, file traffic, I/O activity, NJE records, storage allocation and deallocation, inter-task communication, SNA activity, spool activity, and task creation and end. Activities can be traced for all tasks or for specific task types and name.

Abend Symptom Display: When a task abends, a symptom summary is displayed at the console. Also, an exit can be used to perform further abend processing.

More Eye-Catchers: Eye-catchers have been added throughout the RSCS load module to indicate subroutines and main module entry points. They are also included in many new data areas and in all storage pages allocated by RSCS's new quick storage allocator.

Improved Transmission Algorithms

The transmission algorithms supplied with RSCS have been improved, and the facilities for writing more advanced ones are also better. Transmission algorithms can now be specified by name and by number. Named transmission algorithms can be written without modifying any RSCS code. A transmission algorithm can also now block individual files from being transmitted. This makes it easier to manage service levels. For example, large files can be blocked from transmitting during the busiest hours of the day.

The NJE protocol for SNA, CTC, and BSC links allows a maximum of 14 streams. For GPI links, however, RSCS allows transmission algorithms to use up to 32 streams.

With RSCS 3.1.0, transmission algorithm 1 now allows up to seven streams (only two could be used in Version 2) and provides flexible means to specify which file sizes can transmit on which streams.

Improved Rerouting

The REROUTE command has been improved to include commands, files, messages, and rerouting events can be hidden from traffic originators.

Job Header Options

With RSCS Version 2, exits were often coded to set values in the job header, job name, programmer name, building, and room fields. In RSCS 3.1.0, these fields can be set globally by system options or for individual files by options on the TAG command.

Selectable Compression

In RSCS Version 2, data is always compressed on SNA and BSC links, and never on CTC links. In RSCS 3.1.0, these remain the defaults, but parameters have been added to networking links to allow these defaults to be overridden.

Printer Improvements

Printer Pool Support: RSCS's use of routing groups and multiple primary links can be exploited to address a *pool* of printers by a single name. A file will print on the first printer in the pool that is able to print it.

Files with multiple copies will print on the same printer, unless this printer malfunctions during the print job; in which case the remaining copies will be printed on another printer in the pool.

Sharing on Token Rings: Additional 3270 printer control sequences are now transmitted to indicate the end of a print job. This means that printers can effectively be shared on a local basis on a token ring. With Version 2, printers could be shared but only between hosts running VTAM.

Simulated Forms Control Buffer (FCB) Support: FCBs are used by CP-attached system printers to control the spacing of print lines on a form. 3270 printers do not directly support FCBs, but to provide a simulation of CP printers, RSCS 3.1.0 can simulate the effects of FCBs on 3270 printers.

Improved Programming Practices

The RSCS supervisor and main utility modules have been almost completely rewritten for RSCS 3.1.0. The result shows as more routines accessible from the CRV. Some routines are specific to RSCS; others are general-purpose utilities that RSCS uses. The new utility routines also simplify the writing of exit routines for RSCS 3.1.0.

RSCS 3.1.0 also provides a new command and statement parser. This new support enables RSCS to completely parse all commands and configuration file statements before handing them over to an appropriate processing routine for execution. This support simplifies the command and statement processors and has contributed to the increased number of commands and options available in RSCS 3.1.0.

Second-Level Routing on DESTs Removed

In Version 2, if a DEST name matched a printer name, files could not be successfully received to that DEST and processed by PSF before being printed on a printer. This resulted in many bypasses for group 1 3812 and 3816 devices. In RSCS 3.1.0, this second-level routing has been eliminated. Bypasses will not be affected but it is now possible to remove them.

Pending File Mechanism Removed

Previous versions of RSCS allowed the system programmer to specify a pool of less than 10,000 TAG slots to conserve virtual storage. If a file arrived but no TAG slot could be allocated for it, the file would be put into *pending* state. RSCS would enter a degraded mode of operation that caused severe performance and operational difficulties.

RSCS 3.1.0 ignores any TAGS statement in the configuration file and allocates 10,000 TAG slots automatically. Although this can result in increased *virtual* storage utilization, great care has been taken to keep *real* storage utilization down, so that even in small storage-constrained systems, real storage use because of queued spool files is less in Version 3 than in Version 2.

VM RSCS Data Interchange Manager

The VM RSCS Data Interchange Manager (RSCS Interchange) is available as a separate feature of RSCS 3.1.0. This support enables users in an RSCS network to easily send notes to users in a TCP/IP network who use SMTP note facilities.

Version 3 Release 1.1 Enhancements

This section describes the enhancements provided by RSCS Version 3 Release 1.1 (RSCS 3.1.1). This support is offered in addition to the support provided by RSCS 3.1.0.

VMSES/E Support

This support enables customers to install and service RSCS 3.1.1 using the Virtual Machine Serviceability Enhancements/Extended (VMSES/E) component of VM/ESA. The step-by-step installation and service procedures are described in the *RSCS Program Directory*, which is automatically supplied with the product order.

3270 Print Driver Enhancements

This support provides consistency between double-byte character set (DBCS) and non-DBCS print streams. It also enables RSCS to support the APL2 character set and improves error condition handling for print files.

RSCS Interchange Availability

The RSCS Data Interchange Manager (RSCS Interchange), which was a separately-orderable feature of RSCS 3.1.0, is now included with RSCS 3.1.1. In addition, RSCS Interchange can also be installed and serviced using the VMSES/E component of VM/ESA.

Version 3 Release 2 Enhancements

In addition to the support provided by RSCS 3.1.0 and RSCS 3.1.1, RSCS Version 3, Release 2 (RSCS 3.2) provides the following enhancements.

31-bit Addressing Support

RSCS 3.2 has been enabled to use 31-bit addressing, which allows RSCS to use storage above the 16MB line. This support relieves storage constraints and enables you to define more links to the RSCS virtual machine. As part of this support, RSCS and the RSCS Interchange server must be initialized in an ESA mode virtual machine on a VM/ESA system.

All modules, macros, and sample programs supplied with RSCS 3.2 are 31-bit enabled. In addition, RSCS can now support virtual addresses up to X'FFFF'. Some RSCS messages and trace headers will also display 4-digit addresses.

All user-written exit routines used with RSCS 3.2 and RSCS Interchange must also be enabled for 31-bit addressing. These exit routines will run in an ESA mode virtual machine. The *VM/RSCS: Exit Customization* book contains more information to help you migrate or write exit routines.

Transmission Control Protocol/Internet Protocol Support

RSCS 3.2 provides the following enhancements to enable interaction with hosts or connections within a TCP/IP network.

- The LPD (line printer daemon) link drivers enable RSCS to accept files from an IP network. The files can then be distributed on the local system or routed to a remote node in the RSCS network.

  RSCS 3.2 also adds LPD exits, which are used with the LPD- type links. These exit points let you customize the data streams received. You can also use the LPD exits to specify where the file will be delivered within the NJE network.

- The LPR (line printer router) link drivers allow RSCS to send print files to TCP/IP line printer daemons within a TCP/IP network. The files can then be printed anywhere within a TCP/IP network.

  RSCS 3.2 also adds LPR exits, which are used with the LPR-type links. These exit points let you customize the data streams that are sent over the link according to the type of printer that will be used. You can also use the LPR exits to specify where the file will be printed (remote host and port) within a TCP/IP network.

- RSCS 3.2 provides a socket interface that enables customers to create other RSCS TCP/IP link drivers.

- The TCPASCII link drivers allow RSCS to send output to ASCII printers that are attached to terminal servers within a TCP/IP network. This enhancement enables RSCS to support ASCII printers that are attached to workstations and use TCP/IP as the transport medium.

  The TCPASCII-type links are similar to the existing ASCII-type links; however, an ASCII control unit is not required to communicate with the printer. In addition, except for a small difference in the data passed to the Attention interrupt processing exit, the TCPASCII-type links have been designed to support the current ASCII printer and plotter exits.

- The TCPNJE link drivers enable RSCS to send data to another peer NJE node within a TCP/IP network.

- The TN3270E link drivers allow RSCS to send print files to Information Display System Printers attached within a TCP/IP network.

- The UFT (Unsolicited File Transfer) link drivers allow RSCS to send files to TCP/IP UFT daemons within a TCP/IP network. The files can then be delivered to a user at a remote node within a TCP/IP network.

  RSCS 3.2 also adds UFT exits, which are used with the UFT- type links. These exit points let you customize the data streams that are sent over the link.

- The UFTD (Unsolicited File Transfer Daemon) link drivers enable RSCS to accept files from TCP/IP UFT clients within a TCP/IP network. The files can then be distributed on the local system or routed to a remote node in the RSCS network.

  RSCS 3.2 also adds UFTD exits, which are used with the UFTD- type links. These exit points let you customize the data streams that are received. You can also use the UFTD exits to specify where files will be delivered within the NJE network.

Exit Facility Enhancements

In addition to the new LPR exits, RSCS 3.2 further enhances the exit facility by adding several new IBM-defined exit points. These exits points are designed to help you customize the processing of SNA3270P- and 3270P-type printer links.

- Exit 44 Link Termination
- Exit 45 Output Page Accounting
- Exit 46 Verification of Page Accounting
- Exit 47 Driver Initialization
- Exit 48 Verification of Output Page Error

For example, you can use these exit routines to obtain accounting information about the number of pages printed on a particular link. For more information about these new exit points, see *VM/RSCS: Exit Customization*.

Enhanced VMSES/E Support

RSCS 3.2 and RSCS Interchange are completely installed and serviced using the VMSES/E component of VM/ESA. The step-by-step instructions are listed in the *RSCS Program Directory*, which is automatically supplied with your product order. The *RSCS Program Directory* also contains instructions to help you verify the installation of the RSCS and RSCS Interchange code.

In addition, the sample exit routines supplied with RSCS 3.2 have been enabled to use the VMSES/E component of VM/ESA. All sample exit routines supplied with RSCS 3.2 have been combined into one load library. In addition, RSCS 3.2 supplies the macro library and control files to help you use these sample exit routines. This support simplifies the installation and service of these sample exit routines.

Enhanced Workstation Link Support

RSCS 3.2 provides the following enhancements to the RJE- and MRJE-type workstation link drivers.

- RJE-Type Link Enhancements

  The POLL, MSG, and OPR parameters have been added for RJE-type links. These options provide greater flexibility for communicating with various types of remote workstations.

- MRJE-Type Link Enhancements

  The TPASS parameter has been added to the MRJE-type links to enhance security when the remote node signs on the host system.

Additional RSCS Enhancements

The following enhancements have also been introduced to RSCS 3.2.

- Enhanced Forms Control Support for Autostart Links

  This support enables you to specify form and mode options when defining an autostart link with the DEFINE or LINKDEFINE statements. Previously, these options could only be specified on the START command, which did not permit autostart links to use forms control.

- 5-digit Spool ID Support

  A new parameter provided on the OPTION statement allows you to select if the origin spool ID of files will be displayed as 5-digits.

- NOTIFY-type Link Enhancements

  The new NETDATA parameter has been added to NOTIFY-type links. This parameter enables you to specify if the NOTIFY-type links send notes in NETDATA format or as punch files.

VM RSCS Data Interchange Manager Enhancements

In Version 3 Release 2, the following enhancements have also been made to the VM RSCS Data Interchange Manager (RSCS Interchange).

- RSCS Interchange has also been enabled to use 31-bit addressing. The sample exit routines supplied with RSCS Interchange have also been enabled to use 31-bit addressing. In addition, these exit routines can now be installed and serviced using the VMSES/E component of VM/ESA.
- Support has been added to enable you to specify, and query, the format of the “From:” header of files that are sent to remote domains. When users send notes through SMTP, this support enables any replies to be returned through a router at your site.

Chapter 7. RSCS Requirements

This section describes the product operational requirements for RSCS 3.2. It also describes the tasks that you may need to perform to support RSCS operations.

RSCS Product Requirements

The following sections provide an overview of the requirements for operating RSCS and RSCS Interchange on your VM/ESA system.

RSCS 3.2 Requirements

RSCS 3.2 must run in an ESA mode virtual machine on one of the following levels of the VM/ESA licensed program:

- VM/ESA Version 1, Release 2.1 (5684-112)
- VM/ESA Version 1, Release 2.2 (5684-112)
- VM/ESA Version 2, Release 1.0 (5654-030) or later

Note: RSCS Version 3 Release 2.0 is preinstalled with VM/ESA Version 2 Release 3.0 or later.

RSCS runs as an application of the GCS component of VM/ESA. RSCS uses GCS for task management and I/O operations; it also uses GCS to communicate with CP or other virtual machines, such as VTAM.

The *VM/RSCS: Planning and Installation* book contains information about how to define the RSCS virtual machine to your VM/ESA system. This is information includes:

- A sample VM/ESA directory entry
- How to define devices to RSCS
- Requirements for each supported type of communication (non-SNA, SNA, and TCP/IP)
- Virtual storage considerations

Table 3 lists the approximate amount of DASD space, in cylinders or blocks, needed to store the required and optional files for RSCS 3.2. The *RSCS Program Directory* contains additional information about these requirements.

Table 3. Approximate Version 2 Release 4.0 DASD Requirements

Device Type	Storage Requirements	
	Required Files	Optional Files
9345, 3380	120	85
3390	105	75
3375	188	138
3350	152	115
FB-512 blocks	133830	106200
SFS 4K blocks	14250	12775

To communicate with users in a System Network Architecture (SNA) network, RSCS requires ACF/Virtual Telecommunications Access Method (VTAM). One of the following releases of VTAM must also be installed on your VM/ESA system:

- ACF/VTAM Version 3 Release 4.1 for VM/ESA (5684-095), or later
- ACF/VTAM Version 4 Release 2 for VM/ESA (5654-010), or later

RSCS and VTAM share the same GCS virtual machine group (note that the VTAM release installed on your system must support the ESA version of GCS). Under GCS, VTAM creates an interface that RSCS uses to exchange information with other VTAM applications in the SNA network. VTAM selects the path that RSCS uses to communicate within the SNA network. It also allows RSCS to send and receive data across SNA-type links.

To enable RSCS to communicate within an IP network, Transmission Control Protocol/Internet Protocol for VM (TCP/IP) must also be installed on the VM/ESA system. For TCP/IP connections, TCP/IP Version 2 Release 2 for VM (5735-FAL) (and APAR PN40284, PTF UN42195), or later, is required.

Notes:

1. If RSCS 3.2 is installed on a VM/ESA Version 2 Release 1.0 system and TCP/IP connections will be used, TCP/IP Version 2 Release 3 for VM is required.
2. TCP/IP is preinstalled on VM/ESA Version 2 Release 3.0, or later, systems in a disabled state.

RSCS Interchange Requirements

Like RSCS, RSCS Interchange also runs in an ESA mode virtual machine that must be defined to your VM/ESA system. A link between RSCS and the RSCS Interchange server must also be defined. In addition, the RSCS Interchange server must be able to communicate with, and be defined to, a target system that is capable of supporting SMTP mail.

The *RSCS Data Interchange Manager Installing, Managing, and Using* book and the *RSCS Program Directory* contain more information about RSCS Interchange requirements; this information includes disk sizes and examples to help you define and prepare the RSCS Interchange server.

RSCS Customer Responsibilities

This section describes the tasks your personnel will need to perform to support RSCS. These tasks fall into three categories — those done *once* (initially), those that are *ongoing* (daily, weekly, monthly), and those that are *occasional* (as needed).

One-Time Tasks

Note: RSCS Version 3 Release 2.0 is preinstalled with VM/ESA Version 2 Release 3.0 or later.

These tasks are usually done by system programmers, support personnel, and administrators.

- Planning

Consider designating someone to plan for installing RSCS on your system and making it operational. Planning will ensure a smoother process.

- Installation

  First, someone must install the product.

  This is done by a well-defined, step-by-step procedure that is described in the *RSCS Program Directory*, which is provided with the RSCS 3.2 product order. For more information to plan for RSCS installation, see *VM/RSCS: Planning and Installation*.

  Note: RSCS Version 3 Release 2.0 is preinstalled with VM/ESA Version 2 Release 3.0 or later. Most install steps are complete. Enable, tailoring, test and placing into production steps still need to be completed.

- Customization

  Someone at your installation must define the components and characteristics of your RSCS network. Tasks include defining links and routes between nodes and devices. The *VM/RSCS: Planning and Installation* book describes how to set up an RSCS network and manipulate file traffic.

  Optionally, your installation can write and install RSCS exit routines. You might use these exit routines to track or perform accounting functions for data traffic on, and through, your system. The *VM/RSCS: Exit Customization* book provides specific information to help you customize RSCS.

- Operation and Administration

  Automating RSCS will simplify many aspects of RSCS operation and use. You can do most of this through exec procedures. Someone must create such procedures, so they are there for use by RSCS, operators, and users. You can find some suggestions of ways to automate your operation in “Easy to Operate” on page 28.

Use the *VM/RSCS: Planning and Installation* and *VM/RSCS: Exit Customization* books as resources for these tasks.

Ongoing Tasks

Ongoing tasks are those activities that are generally performed by RSCS operators and users. These types of tasks are described in more detail in the *VM/RSCS: Operation and Use* book.

- Operation

  Operation includes both the operation of the RSCS system and the operation of its remote devices. You can and, for convenience, should assign remote (on-site) operators to control remote devices. Although RSCS is designed to run without regular operator intervention, there are tasks for an operator to perform. For example, after initialization, the RSCS operator will have to monitor operations, make dynamic adjustments, and react to unusual situations.

- Use

  VM/ESA and other products use RSCS to handle communications between systems. Because installations may create automatic procedures, users need not be aware they are using RSCS. However, users can interact with RSCS directly to perform tasks, such as: sending messages and files to other systems, monitoring the status of files in transit, and using remote workstations.

Occasional Tasks

The following tasks are usually performed by support personnel.

- Adjusting and Refining Customization

  Occasionally, you may need to make adjustments because of long-term changes in the RSCS operating environment. For example, execs associated with RSCS or installation-created exit routines may need to be adjusted because of changes in the following areas:

  – User population
  – Network configuration
  – Designated responsibilities
  – Volume of data traffic.

  Use the *VM/RSCS: Planning and Installation* and *VM/RSCS: Exit Customization* books as resources for these tasks.

- Diagnosis

  At times, there may be problems with errors on communications lines, slow response, or overloads. Someone will need to recognize symptoms and follow procedures for collecting data to find a solution or report the problem. Use the *VM/RSCS: Diagnosis Reference* as a resource for this task. The *VM/RSCS: Operation and Use* book, which provides information about how to use the RSCS QUERY command, is also helpful.

- Service

  Sometimes it is necessary for someone to apply service to your system—either as a preventive measure or to fix a problem. The *RSCS Program Directory* provides specific information to help you apply any required service updates to RSCS or RSCS Interchange.

Chapter 8. RSCS Library Guide

This section describes the documentation available with RSCS. It offers an abstract of the RSCS books and provides ordering information.

RSCS Library

You can learn more about the RSCS licensed product by using the other books in the library. You can order individual RSCS books from your IBM Branch Office.

Title	Order Number
Evaluation	
VM/RSCS: Licensed Program Specifications	GH24-5223
VM/RSCS: General Information	GH24-5218
Planning, Installation, and Service	
VM/RSCS: Planning and Installation	SH24-5219
VM/RSCS: Data Interchange Manager Installing, Managing, and Using	SH24-5248
End Use	
VM/RSCS: Operation and Use	SH24-5220
VM/RSCS: Reference Summary	SX24-5257
Customization	
VM/RSCS: Exit Customization	SH24-5222
Diagnosis and Reference	
VM/RSCS: Diagnosis Reference	SC24-5881
VM/RSCS: Messages and Codes	SH24-5221

In addition, a copy of *RSCS Program Directory* is automatically supplied with your product order.

VM/RSCS: Planning and Installation

The *VM/RSCS: Planning and Installation* book is written for people who are responsible for installing, servicing, and customizing RSCS for new users. It is divided into two sections: planning and installation.

- Planning

  This section describes how to prepare to install RSCS. Each chapter contains a checklist of items that you should consider before performing a task or using RSCS on your system. The requirements and considerations for the following tasks are discussed:

  - Installation
  - Customization
  - Operation
  - Administration
  - Diagnosis
  - Migration

- Installation

This section contains overview information about the installation and service process for RSCS. The detailed step-by-step instructions are listed in the *RSCS Program Directory*, which is supplied with your product order.

This section also contains information to help you customize and define your RSCS network after the product code is installed.

In addition, the *VM/RSCS: Planning and Installation* book also contains the syntax and function descriptions of each RSCS configuration file statement. This book also contains many examples to demonstrate how you can define your RSCS network. Examples include: a directory entry for the RSCS virtual machine, how to define a complex RSCS network, and VTAM logon mode table entries.

VM/RSCS: Operation and Use

The *VM/RSCS: Operation and Use* book describes how to operate and use RSCS on your system. This book introduces you to RSCS — what it does, who can use it, and how it interacts with other products, such as: VM/ESA, VTAM, and TCP/IP. The book contains instructions for performing many RSCS tasks, including:

- Starting and stopping RSCS
- Managing links and ports
- Monitoring the progress of your files
- Manipulating files in the network
- Automating network operations

Appendixes also describe the syntax and functions of the RSCS commands and link operational parameters.

VM/RSCS: Messages and Codes

The *VM/RSCS: Messages and Codes* book contains an explanation of the RSCS messages and abend codes. The messages, which are listed numerically, contain an explanation, a description of the system actions, and recommendations for any further action, if appropriate. An alphabetic cross-reference of the message text is also provided. This book also contains an explanation of columnar messages returned when you enter QUERY commands. In addition, the abend codes that GCS or RSCS returns when they detect a programming error are also explained.

VM/RSCS: Exit Customization

The *VM/RSCS: Exit Customization* book describes how you can customize RSCS processing to meet any functional requirements of your facility. You can use several categories of exits to customize specific areas of RSCS processing:

- ASCII printer and plotter exits
- Gateway link driver
- Loadable link driver
- LPD exits
- LPR exits
- RSCS exit facility
- Transmission algorithms
- UFT exits
- UFTD exits

This book explains each category and helps you select the type of customizing vehicle that will best suit your needs. It includes information on the standard

conventions that you must use when writing exit routines. It also includes examples that you can use in writing your exit routines.

VM/RSCS: Diagnosis Reference

The *VM/RSCS: Diagnosis Reference* describes the internal functions of RSCS. This book is intended for people who are responsible for customizing, diagnosing, and modifying RSCS. These readers must be familiar with RSCS programming techniques and operating procedures. They also must be familiar with the concepts, terminology, and use of other licensed programs, such as: VM/ESA, VTAM, and TCP/IP for VM.

The *VM/RSCS: Diagnosis Reference* is organized into three parts:

- Part 1, Functional Overview, describes the functions, tasks, and structures that enable RSCS to process files, messages, and commands in the network.
- Part 2, Diagnostic Aids, contains information to help you collect and process diagnostic information for RSCS.
- Part 3, Reference Directories, contains information about the RSCS modules and data area structures.

RSCS Data Interchange Manager Installing, Managing, and Using

The *RSCS Data Interchange Manager Installing, Managing, and Using* book contains information to help you use RSCS Interchange. Topics include:

- Planning considerations
- Installation and service overview
- Customization
- Administering and using RSCS Interchange

This book also describes the commands, configuration file statements, and messages associated with RSCS Interchange.

VM/RSCS: Reference Summary

The *VM/RSCS: Reference Summary* is a booklet that summarizes a variety of information about RSCS and RSCS Interchange. The *VM/RSCS: Reference Summary* shows the syntax and a brief description of the following items:

- RSCS commands
- RSCS configuration file statements
- Link characteristics and operational parameters
- RSCS Interchange commands
- RSCS Interchange configuration file statements
- Commonly-used CP and CMS commands
- Diagnostic subcommands
- Terminal signon card formats

RSCS Program Directory

The *RSCS Program Directory* contains all the detailed instructions for installing, servicing, enabling, and verifying RSCS. It also contains information about the layout of the RSCS product tapes. In addition, the *RSCS Program Directory* contains installation and service information for the VM RSCS Data Interchange Manager.

The *RSCS Program Directory* is supplied automatically with your product order. You should ensure that a copy is available before you start any installation or service procedures. You should also keep a copy available for future reference.

To obtain additional copies or to ensure you have the latest copy of the *RSCS Program Directory*, you can order the PTF UV98320 through the IBM Support Center or through S/390 SoftwareXcel.

Part 3. Reference Information

This section contains additional reference information to help you use and learn more about RSCS.

- “Glossary and Abbreviations” on page 69 contains a list of RSCS-related terms and abbreviations.
- “Bibliography” on page 81 contains a list of additional publications that are helpful when using RSCS.
- “Using the Master Index” on page 85 describes how to use the master index, which is supplied in the following section, to find information in the RSCS library.

Glossary and Abbreviations

A

abend. (1) Abnormal end of task. (2) Synonym for *abnormal termination*.

abnormal termination. The ending of processing before planned termination. Synonymous with *abend*.

accept. Allowing a connection to the user's virtual machine from another virtual machine or from the user's own virtual machine.

access method. A technique for moving data between main storage and I/O devices.

ACF. Advanced communications function.

ACF/VTAM. Advanced Communications Function for Virtual Telecommunications Access Method.

activate. To make a resource of a node ready to perform the functions for which it was designed. Contrast with *deactivate*.

active. The state a resource is in when it has been activated and is operational. Contrast with *inoperative*.

adjacent. In RSCS, two nodes are adjacent if they are connected by only one link.

Advanced Communications Function (ACF). A group of IBM program products (principally VTAM, TCAM, NCP, and SSP) that use the concepts of SNA, including distribution of function and resource sharing.

Advanced Communications Function for Virtual Telecommunications Access Method (ACF/VTAM). An IBM licensed program that controls communications and flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

all points addressable (APA). The ability to refer to any pixel on the printable area of the screen. (Pixel is a dot that forms part of an image on the screen.)

alphameric. Synonym for *alphanumeric*.

alphanumeric. A character set that contains letters, digits, and usually other characters, such as punctuation marks. Synonymous with *alphameric*.

alternate link. In RSCS, a link to transmit data over when the status of none of the primary links defined in the routing of a node is connect. Contrast with *primary link*.

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 graphic characters.

APA. All points addressable.

apply. When servicing a product or component, to generate an auxiliary control structure from a PTF.

area. A term acceptable for DASD space when there is no need to differentiate between space on count-key-data devices and FB-512 devices. See *DASD space*.

ASCII. American National Standard Code for Information Interchange.

ASCII printer and plotter exits. An exit that allows files to be transmitted to printers and plotters connected to RSCS through a controller, such as the IBM 7171 ASCII Device Attachment Control Unit, or by a terminal server in a TCP/IP network.

ASCII-type link. A type of RSCS link that connects a VM system and an ASCII printer or plotter using an ASCII control unit. Contrast with *TCPASCII-type link*.

assembler language. A source language that includes symbolic machine language statements in which there is a one-to-one correspondence with instruction formats and data formats of the computer.

authorized operator. An operator, other than the RSCS central operator, who is authorized to control the RSCS virtual machine or individual links.

automatic logon. A process when an operator's or user's virtual machine is initiated by someone other than the user of that virtual machine. For example, the primary VM system operator's virtual machine is automatically started during VM initialization. A privileged user can enter the AUTOLOG command to start some other (disconnected) virtual machine.

available. In VTAM, pertaining to an LU that is active, connected, enabled, and not at its session limit.

B

BDT. Bulk data transfer. See *MVS/Bulk Data Transfer*.

binary digit. Either of the digits 0 or 1 when used in the pure binary numeration system. Synonymous with *bit*.

binary synchronous communication (BSC). Communication using binary synchronous line discipline in which transmission of binary-coded data between stations is synchronized by timing signals generated at the sending and receiving stations.

bit. (1) Either of the binary digits 0 or 1. See *byte*. (2) Synonym for *binary digit*.

block. A unit of DASD space on FB-512 devices. For example, FB-512 devices can be the IBM 9332, 9335, 9313, 9370, and 3310 DASD using fixed-block architecture.

BSC. Binary synchronous communication.

buffer. An area of storage, temporarily reserved for performing input or output, into which data is read, or into which data is written.

build. In the installation and service of a product, to do the necessary steps to produce executable code or systems. This is often called the *build process*.

byte. A unit of storage, consisting of eight adjacent binary digits that are operated on as a unit and constitute the smallest addressable unit in the system.

C

changes. In installation and service, IBM and original equipment manufacturer (OEM) supplied service for their programs. In the IBM service process, there are many ways users can receive information they need to fix (change) a portion(s) of a product they are running on a VM system. These include PTFs, APARs, user modifications, and information received over the phone. All these types of information are called *changes*.

channel. A path in a system that connects a processor and main storage with an I/O device.

channel-attached. Pertaining to the attachment of devices directly by data channels (I/O channels) to a host processor. Synonymous with *local-attached*.

channel-to-channel adapter (CTCA). A hardware device that connects two channels on the same computing system or on different systems.

channel-to-channel (CTC) device. A hardware device that connects two channels on the same computing system or on different systems. CTC devices include CTCAs, ESCON CTCAs, and 3088 MCUs.

CMS. Conversational Monitor System.

columnar messages. A series of messages formatted into rows and columns of information. Columnar messages consist of two parts: header lines and body lines. Usually, they are followed by a summary message.

command response interface (CRI). A utility that lets users specify which method RSCS should use to send command responses, which language RSCS should use to build the message text, and what type of signature RSCS should include with each response. When RSCS sends the command responses, each line of the response is numbered and a message 001 is added to the end of the responses generated for each command.

common storage. A shared segment of reentrant code that contains free storage space, the GCS supervisor, control blocks, and data that all members of a virtual machine group share.

communication adapter. A program or a part of a program that handles communications with other programs.

communication link. Synonym for *data link*.

communication vector table (CVT). An RSCS control block that contains pointers to information used by RSCS modules.

component. A collection of objects that together form a separate functional unit. A product may contain many components (for example, VM has components of AVS, CP, CMS, GCS, REXX, TSAF, Dump Viewing Facility, and VMSES/E). A component can be part of many products.

concurrently. Concerning a mode of operation that includes doing work on two or more activities within a given (short) interval of time.

configuration. (1) The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units. The term may refer to a hardware or a software configuration. (2) The devices and programs that make up a system, subsystem, or network.

configuration file. A file used by RSCS to define the configuration of a computer network.

connect. Establishing a path to communicate with another virtual machine or with the user's own virtual machine.

connection. Synonym for *physical connection*.

console. A device used for communications between the operator or maintenance engineer and the computer.

console spooling. Synonym for *virtual console spooling*.

Control Program (CP). A component of VM that manages the resources of a single computer so multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an IBM System/370.

control unit. A device that controls I/O operations at one or more devices.

conversation. A connection between two transaction programs over an LU-LU session that lets them communicate with each other while processing some transaction. The programs establish a conversation, send and receive data in the conversation, and then terminate the conversation.

Conversational Monitor System (CMS). A virtual machine operating system and component of VM that provides general interactive time sharing, problem solving, program development capabilities, and operates only under the control of the VM Control Program (CP).

corrective service. Service that IBM supplies on tape to correct a specific problem.

CP. Control Program.

CP directory. Synonym for *VM directory*.

CRI. Command response interface.

CRV. Common routines vector.

CTC. Channel-to-channel.

CTCA. Channel-to-channel adapter.

customization. The process of defining and activating a configuration and changing system processing to meet user requirements.

CVT. Communications vector table.

cylinder. In a disk pack, the set of all tracks with the same nominal distance from the axis about which the disk pack rotates.

D

DASD. Direct access storage device.

DASD space. (1) Area allocated to DASD units on CKD devices. (2) Area allocated to DASD units on FB-512 devices. Note that *DASD space* is synonymous with *cylinder* when there is no need to differentiate between CKD devices and FB-512 devices.

data area. A storage area used by a program or device to hold information.

data link. (1) The equipment and rules (protocols) used for sending and receiving data. Synonymous with *communication link*. (2) In SNA, synonym for *link*.

deactivate. To take a resource of a node out of service, rendering it inoperable, or to place it in a state in which it cannot perform the functions for which it was designed. Contrast with *activate*.

default. An attribute, value, or option that is assumed when none is explicitly specified.

delimiter. (1) A flag that separates and organizes items of data. Synonymous with *separator*. (2) A character that groups or separates words or values in a line of input. Usually one or more blank characters separate the command name and each operand or option in the command line. In certain cases, a tab, left parenthesis, or backspace character can also act as a delimiter.

direct access storage device (DASD). A storage device in which the access time is effectively independent of the location of the data.

directory. See *VM directory*.

disk. A magnetic disk unit in the user's CMS virtual machine configuration. Also called a virtual disk.

display. (1) To present information for viewing, usually on a terminal screen or a hardcopy device. (2) A device or medium on which information is presented, such as a terminal screen. (3) Deprecated term for *panel*.

dump. (1) Computer printout of storage. (2) To write the contents of part or all of main storage, or part or all of a minidisk, to auxiliary storage or a printer. See *abend dump*.

duplex. Pertaining to communication in which data can be sent and received at the same time. Synonymous with *full duplex*. Contrast with *half duplex*.

E

EBCDIC. Extended binary-coded decimal interchange code.

edit. A function that makes changes, additions, or deletions to a file on a disk. These changes are interactively made. The edit function also generates information in a file that did not previously exist.

element. In reference to installation and service of a product, a file provided on a program update tape (PUT) as input to the build process. An element is the smallest serviceable unit of a component. Several files can be associated with a given element, but each file has the same file name. See *build*.

emulation. The use of programming techniques and special machine features to permit a computing system to execute programs written for another system.

entry point. An address or label of an instruction performed on entering a computer program, a routine, or a subroutine. A program can have several different entry points, each corresponding to a different function or purpose.

event. An occurrence of significance to a task; typically, the completion of an asynchronous operation, such as an I/O operation.

events file. The configuration file used to initialize the RSCS event scheduler. This file contains a list of event definitions and special dates that define the initial scheduled events when RSCS is initialized.

exit. (1) An instruction in a program, routine, or subroutine that causes control to pass to another program, routine, or subroutine. See *exit point*. (2) A routine, normally user-supplied, that receives control from the system when a certain event such as an abnormal-end exit occurs.

exit facility. See *RSCS exit facility*.

exit point. An instruction in a program, routine, or subroutine that causes control to pass to another program, routine, or subroutine. After the other program, routine, or subroutine has finished processing, control usually returns to the instruction in the original program, routine, or subroutine that caused control to pass.

exit routine. (1) A routine that receives control when a specified event occurs, such as an error. (2) Any of several types of user-written routines. See *ASCII printer plotter exits*, *RSCS exit facility*, *transmission algorithms*, *LPD exit, LPR exit, UFT exit,* and *UFTD exit*.

extended binary-coded decimal interchange code (EBCDIC). A set of 256 characters, with each character represented by 8 bits.

F

facility. A service provided by an operating system for a particular purpose.

FCB. Forms control buffer.

feature. A particular part of an IBM product that a customer can order separately.

file access mode. A file mode number that designates whether the file can be used as a read-only or read/write file by a user. See *file mode*.

file ID. A CMS file identifier that consists of a file name, file type, and file mode. The file ID is associated with a particular file when the file is created, defined, or renamed under CMS. See *file name*, *file type*, and *file mode*.

file mode. A two-character CMS file identifier field containing the file mode letter (A through Z) followed by the file mode number (0 through 6). The file mode letter indicates the minidisk or SFS directory on which the file resides. The file mode number indicates the access mode of the file. See *file access mode*.

file name. A one- to eight-character alphanumeric field, containing A through Z, 0 through 9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is part of the CMS file identifier and serves to identify the file for the user.

file tag. (1) In VM, a character string associated with a VM spool file that contains information pertaining to that spool file. (2) In RSCS, the file tag contains the destination link ID and transmission priority of the file.

file type. A one- to eight-character alphanumeric field, containing A through Z, 0 through 9, and special characters $ # @ + - (hyphen) : (colon) _ (underscore), that is used as a descriptor or as a qualifier of the file name field in the CMS file identifier.

forms control buffer (FCB). In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output. The FCB is analogous to the punched-paper, carriage-control tape that IBM 1403 Printers use.

full duplex. Synonym for *duplex*.

G

gateway. (1) The combination of machines and programs that provide address translation, name translation, and SSCP rerouting between independent SNA networks to allow those networks to communicate. A gateway consists of one gateway NCP and at least one gateway SSCP. (2) In VM, the LU name of a VM system or TSAF collection that is a source for communications to an SNA-defined network or the target of communications from an SNA-defined network.

GATEWAY-type link. A type of RSCS link that connects a VM system to another node using an installation-defined protocol.

GCS. (1) Group Control System.

group. (1) In RSCS, synonym for *routing group*. (2) In VM, synonym for *virtual machine group*.

Group Control System (GCS). A component of VM, consisting of a shared segment that the user can IPL and run in a virtual machine. It provides simulated MVS services and unique supervisor services to help support a native SNA network.

guest. An operating system running in a virtual machine managed by a VM control program. Contrast with *host*.

guest operating system (GOS). A second operating system that runs on the user's primary operating system. An example of a GOS is VSE running on VM to support VM/VCNA.

H

half duplex. In data communication, pertaining to transmission in only one direction at a time. Contrast with *duplex*.

help panel. An online display that tells you how to use a command or another aspect of a product. See *task panel*.

host. A VM control program in its capacity as manager of a virtual machine in which another operating system is running. Contrast with *guest*.

host system. A data processing system that prepares programs and the operating environments for use by another computer or controller.

HPO. High Performance Option.

I

I/O. Input/output.

IBM-defined exit. An exit point that has already been placed and designed into the RSCS code by IBM.

imbed. (1) To make something an integral part of. (2) To place or fix firmly in surrounding matter.

initialize. To set counters, switches, addresses, or contents of storage to starting values.

inoperative. The condition of a resource that has been active, but is not. The resource may have failed, received an INOP request, or is suspended while a reactivate command is being processed.

input/output (I/O). (1) A device whose parts can do an input process and an output process at the same time. (2) 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.

interaction. A basic unit that records system activity, consisting of acceptance of a line of terminal input, processing of the line, and a response, if any.

interface. A shared boundary between two or more entities. An interface might be a hardware or software component that links two devices or programs together.

interrupt. A suspension of a process, such as execution of a computer program, caused by an external event and done in such a way that the process can be resumed.

invoke. To start a command, procedure, or program.

J

JES2. A job entry subsystem for MVS.

JES3. A job entry subsystem for MVS.

job. A unit of work within a network. It consists of all data beginning with a job header control record and ending with a job trailer control record.

job header. A control record that provides general information relating to a job as a whole.

K

keyword. (1) A name or symbol that identifies a parameter. (2) A part of a command operand that consists of a specific character string (such as FEATURE=).

L

line driver. An RSCS task (program) that permits communication between RSCS and a non-SNA type of remote station. Contrast with *session driver*.

link. A connection, or ability to communicate, between two adjacent nodes in a network. Synonymous with *data link*.

link driver. Term that refers collectively to all tasks that establish a link between the local RSCS virtual machine and a remote node. In RSCS, there are two types of link driver tasks: line and session driver tasks. See also *line driver* and *session driver*.

link ID. Link identifier.

link table. An RSCS control block that contains the definitions for all the links in a system.

load. In installation and service, to move files from tape to disk, auxiliary storage to main storage, or minidisks to virtual storage within a virtual machine.

load module. A computer program in a form suitable for loading into main storage for execution.

local. (1) Two entities (for example, a user and a server) are said to be local to each other if they belong to the same system within a collection or to the same node within an SNA system. (2) In RSCS, channel-attached; that is, connected through only a channel, not through an RSCS-defined link. Local refers to a computer, I/O devices, and virtual machines that communicate with one another only over channels. Contrast with *remote*.

local service. Changes manually applied to a product or component (that is, not using the program update service or corrective service procedures). See *user modification*.

local-attached. Depreciated term for *channel-attached*.

logical unit (LU). In VM, an entity addressable within an SNA-defined network, similar to a node within a VM network. LUs are categorized by the types of communication they support. A TSAF collection in an SNA network is viewed as one or more LUs.

logon. The procedure by which a user begins a terminal session. See *automatic logon*.

LPD exit. A type of RSCS exit that enables RSCS to adjust information for print files that are received over a LPD-type link from a LPR client in a TCP/IP network.

LPD-type link. A type of RSCS link that enables RSCS to receive print data streams from a LPR client in a TCP/IP network.

LPR exit. A type of RSCS exit that enables RSCS to adjust information for print files that are sent over a LPR-type link to a line printer daemon in a TCP/IP network.

LPR-type link. A type of RSCS link that enables RSCS to send print data streams to a line printer daemon in a TCP/IP network.

LU_T1. Logical Unit Type 1 (session).

M

machine. A synonym for a virtual machine running under the control of VM.

macro. Synonym for *macrodefinition* and *macroinstruction*.

macrodefinition. A set of statements that defines the name of, format of, and conditions for generating a sequence of assembler language statements from a single source statement. Synonymous with *macro*.

macroinstruction. In assembler language programming, an assembler language statement that causes the assembler to process a predefined set of statements called a macrodefinition. The statements usually produced from the macrodefinition replace the macroinstruction in the program. Synonymous with *macro*.

message. Data sent from a source application to a target application program in a conversation.

message repository. A source file that contains message texts for a VM component or user application. The message texts in a repository file can be translated and used to support national languages.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

module. (1) A unit of a software product that is discretely and separately identifiable with respect to modifying, compiling, and merging with other units, or with respect to loading and execution. For example, the input to, or output from, a compiler, the assembler, the linkage editor, or an exec routine. (2) A nonrelocatable file whose external references have been resolved.

MRJE. Multi-leaving remote job entry.

MRJE-type link. A type of RSCS link that connects a VM system and an MRJE work station using a BSC adapter.

multi-leaving. Fully synchronized two-directional transmission of a variable number of data streams between terminals and a computer using BSC facilities.

Multiple Virtual Storage (MVS). (1) An alternative name for OS/VS2. (2) An IBM program product whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system controlling the execution of programs.

multistreaming. Concurrent transmission of parts of several files so that small files are not held up waiting for transmission of large files. RSCS uses multistreaming on up to seven streams on NJE- or SNANJE-type networking links.

multitasking. Providing services for many tasks that are active at the same time.

MVS. Multiple Virtual Storage.

MVS/Bulk Data Transfer (BDT). An IBM licensed program that (1) copies sequential or partitioned data sets from a JES3 or JES3 computer complex to another JES2 or JES3 computer complex within an SNA network, and (2) lets JES3 computer complexes participate in an SNA NJE network.

N

netdata. The name of the format that sends a file when the NEW option of the CMS SENDFILE command is specified.

network. (1) An interconnected group of nodes. (2) Any set of two or more computers, workstations, or printers linked in such a way as to let data be transmitted between them.

network identifier (network ID). The network name defined to NCPs and hosts to indicate the name of the network in which they reside. It is unique across all communicating SNA networks.

network job entry (NJE). A facility for transmitting jobs, sysout data sets, operator commands and operator messages, and job accounting information from one computing system to another. NJE is supported by JES2, JES3, RSCS, VSE/POWER, and BDT.

network node. Synonym for *node*.

network operator. (1) A person or program responsible for controlling the operation of all or part of a network. (2) The person or program that controls all the domains in a multiple-domain network.

networking. In a multiple-domain network, communication among domains.

NJE. Network job entry.

NJE-type link. A type of RSCS link that connects a VM system and a peer NJE system using a 3088, CTCA, ESCON CTC, or BSC adapter.

node. (1) A single processor or a group of processors in a teleprocessing network. (2) A computer, workstation, or printer, when it is participating in a network. (3) In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. Synonymous with *network node*.

nonprogrammable terminal. In RSCS, a device such as an IBM 2770, 2780, 3770, or 3780 that performs a hardware-defined fixed set of functions; such as printing, reading, and punching data.

nonprogrammable terminal line driver. An RSCS line driver that exchanges data with a nonprogrammable terminal.

NOTIFY-type link. A type of RSCS link that generates notes, retains files, and purges them after a pre-specified number of days.

O

object code. Compiler or assembler output that is executable machine code or is suitable for more processing to produce executable machine code. Contrast with *source code*.

online. Stored in a computer and accessible from a terminal.

operator. A person who operates a machine. See *network operator*.

P

page. (1) The portion of a panel that is shown on a display surface at one time. (2) To move back and forth among the pages of a multiple-page panel. See also *scroll*. (3) (ISO) In a virtual storage system, a fixed-length block that has a virtual address and can be transferred between real storage and auxiliary storage. (4) To transfer instructions, data, or both between real storage and external page or auxiliary storage.

panel. (1) A formatted display of information that appears on a terminal screen. See also *help panel* and *task panel*. Contrast with *screen*. (2) In computer graphics, a display image that defines the locations and characteristics of display fields on a display surface.

parameter. A variable that is given a constant value for a specified application and that may denote the application.

part. A CMS file provided on a product tape or service tape as input to the build process. A part is the smallest serviceable unit of a component. See *build*.

peer system. From the perspective of a particular system node in a network, an equivalent system; one with which jobs and data can be mutually exchanged.

physical screen. Synonym for *screen*.

port table. An RSCS control block that describes the allocation status of switchable line ports available to RSCS.

primary link. In RSCS, one of a number of preferred links to use for transmitting data to a node. Contrast with *alternate link*.

problem determination. The process of identifying the source of a problem; for example, a program component, a machine failure, telecommunication facilities, user or contractor-installed programs or equipment, an environmental failure such as a power loss, or a user error.

process. A systematic sequence of operations to produce a specified result. A process is usually logical, not physical.

product. Any separately installable software program, whether supplied by IBM or otherwise, distinct from others and recognizable by a unique identification code. The product identification code is unique to a given product, but does not identify the release level of that product.

programmable operator facility. This facility enables automatic filtering and routing of messages from a specified virtual machine (for example, the operator) to a logical operator virtual machine in a local distributed or mixed environment. It also permits installation defined actions to be automatically performed.

protocol. (1) A specification for the format and relative timing of information exchanged between communicating parties. (2) The set of rules governing the operation of functional units of a communication system that must be followed if communication is to be achieved. (3) 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. (4) A set of rules for communication that are mutually understood and followed by two communicating stations or processes. The protocol specifies actions that can be taken by a station when it receives a transmission or detects an error condition.

R

receive. (1) Bringing into the specified buffer data sent to the user's virtual machine from another virtual machine or from the user's own virtual machine. (2) To load service files from a service tape.

recommended service upgrade (RSU) tape. A tape containing preventive service for upgrading the current release of a VM/ESA system once it has been installed.

remote. (1) Two entities (for example, a user and a server) are said to be remote to each other if they belong to different systems within a collection, or to different nodes within an SNA network. (2) In RSCS, a connection through one or more RSCS-defined links. Remote describes computers, I/O devices, and virtual machines, to and from which information must pass over an RSCS-defined link. From the perspective of a particular node in an RSCS network, remote refers to any other node in the network. Contrast with *local*.

remote job entry (RJE). Submitting a job and receiving output through an I/O device connected to a computer through communications equipment.

Remote Spooling Communications Subsystem Networking (RSCS). An IBM licensed program and special-purpose subsystem that supports the reception and transmission of messages, files, commands, and jobs over a computer network.

resource. (1) A program, a data file, a specific set of files, a device, or any other entity or a set of entities that the user can uniquely identify for application program processing in a VM system. (2) Any facility of the computing system or operating system required by a job or task, and including main storage, I/O devices, the processing unit, data sets, and control or processing programs.

Restructured Extended Executor (REXX) language. A general-purpose, high-level programming language, particularly suitable for EXEC procedures, XEDIT macros, or programs for personal computing. Procedures, XEDIT macros, and programs written in this language can be interpreted by the System Product Interpreter, also a component of VM.

return code. A code used by a program, routine, or subroutine that influences the execution of succeeding instructions.

RJE. Remote job entry.

RJE-type link. A type of RSCS link that connects a VM system and an RJE workstation using a BSC adapter.

route. A connection to another system by a logical link and one or more intermediate systems.

routing group. A collection of nodes that are grouped together for ease of routing. Groups can be owned by other groups, thus allowing for the creation of extensive tree structures.

RSCS. Remote Spooling Communications Subsystem Networking.

RSCS exit facility. An interface between RSCS and exit routines consisting of exit points established by EXITCALL macros and your own exit routines.

RSCS trace table. An internal trace table that is controlled through the use of ITRACE commands and statements. The ITRACE command and statement allow an installation to trace specific types of events for specific tasks running in the RSCS virtual machine.

S

save area. An area of main storage in which contents of registers are saved.

screen. An illuminated display surface; for example, the display surface of a CRT. Synonymous with *physical screen*. Contrast with *panel*.

scroll. To move all or part of the display image vertically to display data that cannot be observed within a single display image. See also *page*.

separator. Synonym for *delimiter*.

separator page. A printed page indicating the end of output for one job and the start of output for another job.

service. Changing a product after installation. See *corrective service, local service,* and *program update service*.

service machine. A virtual machine running a program that provides system-wide services.

session driver. An RSCS program that permits communication between RSCS and an SNA remote workstation or system. Contrast with *line driver*.

shared. Pertaining to the availability of a resource to more than one use at the same time.

simultaneous peripheral operations online (SPOOL). (1) (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (2) (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

SNA network. The part of a user-application network that conforms to the formats and protocols of Systems Network Architecture. It enables reliable transfer of data among end users and provides protocols for controlling the resources of various network configurations. The SNA network consists of network addressable units (NAUs), boundary function components, and the path control network.

SNANJE-type link. A type of RSCS link that connects a VM system and a peer NJE system through ACF/VTAM.

SNARJE-type link. A type of RSCS link that connects a VM system and an RJE workstation through ACF/VTAM.

SNA3270P-type link. A type of RSCS link that connects a VM system and an IBM 3270 Information Display System printer through ACF/VTAM.

source code. The input to a compiler or assembler, written in a source language. Contrast with *object code*.

spool. (1) (Noun) An area of auxiliary storage defined to temporarily hold data during its transfer between peripheral equipment and the processor. (2) (Verb) To use auxiliary storage as a buffer storage to reduce processing delays when transferring data between peripheral equipment and the processing storage of a computer.

SPOOL. Simultaneous peripheral operations online.

spool file. A file that contains output data saved for later processing.

spool file tag. (1) In VM/ESA, a 136-character data field associated with each output spool file generated. The use, content, and format of this field is a bilateral decision between the originator and receiver of the file. (2) In RSCS, the spool file tag contains the location identifier of the link on which the file is to be transmitted, the user ID of the virtual machine that is to receive the file, and a transmission priority value.

spool identifier (spool ID). A number automatically assigned by CP when the file is closed. The spool ID number can be from 0001 to 9999; it is unique for each spool file. To identify a given spool file, a user must

specify the owner's user ID, the virtual device type, and the spool ID.

spooling. The processing of files created by or intended for virtual readers, punches, and printers. The spool files can be sent from one virtual device to another, from one virtual machine to another, and to real devices. See *virtual console spooling*.

spool manager. An RSCS task that controls access to the CP spool system.

station. (1) One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. (2) One or more computers, terminals, or devices at a particular location.

store-and-forward. A manner of operating a data network in which packets or messages are stored before transmission to the ultimate destination.

subsystem. A secondary or subordinate system, usually capable of operating independent of, or asynchronously with, a controlling system.

supervisor. The part of a control program that coordinates the use of resources and maintains the flow of processing unit operations.

syntax. The rules for the construction of a command or program.

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

TAG. Tag queue element.

tag queue element (TAG). An RSCS control block describing the files enqueued for processing by RSCS. Also called *tag slot*.

tag shadow element. An RSCS data structure that represents a file on a particular link's queue. The tag shadow element contains queueing information and multi-streaming masks for files. Each file in RSCS's reader may be represented by one or more tag shadow elements.

tag slot. Synonym for *tag queue element*.

task. (1) The basic unit of work from the standpoint of a control program. (2) A basic unit of work to be accomplished by a computer. The task is usually specified to a control program in a multiprogramming or multiprocessing environment.

task name. A one- to four-character alphanumeric string the RSCS operator uses to identify a task when communicating with RSCS about that task.

task panel. Online display from which you communicate with the program in order to accomplish the program's function, either by selecting an option provided on the panel or by entering an explicit command. See *help panel*.

TCPASCII-type link. A type of RSCS link that establishes a connection to an ASCII printer that is attached to a remote terminal server in a TCP/IP network.

TCPNJE-type link. A type of RSCS link that connects a VM/ESA system to a remote NJE node in a TCP/IP network.

telecommunication line. Any physical medium such as a wire or microwave beam, that is used to transmit data. Synonymous with *transmission line*.

terminal. A device, usually equipped with a keyboard and a display, capable of sending and receiving information.

throughput. (1) A measure of the amount of work performed by a computer system over a period of time, for example, number of jobs per day. (2) In data communication, the total traffic between stations per unit of time.

TN3270E-type link. A type of RSCS link that connects a VM system and an IBM 3270 Information Display System printer through TCP/IP.

token. An eight-character symbol created by the CMS EXEC processor when it scans an EXEC procedure or EDIT macro statements. Symbols longer than eight characters are truncated to eight characters.

trace table. Synonym for *RSCS trace table*.

transmission algorithm (TA). An exit routine that assigns a file to a particular stream for transmission, concurrent with files in other streams, over a link.

Transmission Control Protocol/Internet Protocol (TCP/IP). A set of communication protocols that support peer-to-peer connectivity functions for local and wide area networks.

transmission line. Synonym for *telecommunication line*.

transparency mode. A BSC mode that permits transmission of any data, bypassing regular BSC control character scanning.

U

UFT. Unsolicited File Transfer

UFT exit. A type of RSCS exit that enables RSCS to adjust information for files that are sent over a UFT-type link to a Unsolicited File Transfer daemon in a TCP/IP network.

UFT-type link. A type of RSCS link that enables RSCS to send data streams to a Unsolicited File Transfer daemon in a TCP/IP network.

UFTD. Unsolicited File Transfer Daemon

UFTD exit. A type of RSCS exit that enables RSCS to adjust information for files that are received over a UFTD-type link from a UFT client in a TCP/IP network.

UFTD-type link. A type of RSCS link that enables RSCS to receive data streams from a UFT client in a TCP/IP network.

user. Anyone who requires the services of a computing system.

user ID. User identification.

user modification. Any change that a user originates for a product or component.

user-written CMS command. Any CMS file created by a user that has a file type of MODULE or EXEC. Such a file can be executed as if it were a CMS command by issuing its file name, followed by any operands or options expected by the program or EXEC procedure.

V

virtual console spooling. The writing of console I/O on disk as a printer spool file instead of, or in addition to, having it typed or displayed at the virtual machine console. The console data includes messages, responses, commands, and data from or to CP and the virtual machine operating system. The user can invoke or terminate console spooling at anytime. When the console spool file is closed, it becomes a printer spool file. Synonymous with *console spooling*.

virtual machine (VM). (1) In VM, a functional equivalent of a real machine. (2) In RSCS, a functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system (see, for example, *conversational monitor system*).

virtual machine group. The concept in GCS of two or more virtual machines associated with each other through the same named system (for example, IPL GCS1). Virtual machines in a group share common read/write storage and can communicate with one another through facilities provided by GCS. Synonymous with *group*.

Virtual Machine Serviceability Enhancements Staged/Extended (VMSES/E). A component of VM/ESA that provides the tools for installing and servicing the various components of the VM product after VM/ESA Version 1 Release 1.0. It is also the strategic installation and service tool for all of the other products that run on those VM/ESA platforms.

Virtual Machine/Enterprise Systems Architecture (VM/ESA). An IBM licensed program that manages the resources of a single computer so that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of a *real* machine.

virtual storage. Storage space that can be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme of the computing system and by the amount of auxiliary storage available, and not by the actual number of main storage locations.

virtual storage extended (VSE). The generalized term that indicates the combination of the DOS/VSE system control program and the VSE/Advanced Functions program product. Note that in certain cases, the term DOS is still used as a generic term; for example, disk packs initialized for use with VSE or any predecessor DOS or DOS/VS system are sometimes called DOS disks. Also note that the DOS-like simulation environment provided under the VM/ESA CMS component and CMS/DOS exists on VM/SP and VM/SP HPO program products and continues to be referred to as CMS/DOS.

virtual storage extended/priority output writers, execution processors, and input readers (VSE/POWER). An IBM licensed program that primarily spools input and output. The networking functions of the program enable a VSE/SP system to exchange files with or run jobs on another remote processor.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in a computer network. It provides single-domain, multiple-domain, and multiple-network capability. VTAM runs under MVS, OS/VS1, VM, and VSE.

VM. Virtual machine.

VM directory. A CP disk file that defines each virtual machine's typical configuration; the user ID, password, regular and maximum allowable virtual storage, CP command privilege class or classes allowed, dispatching priority, logical editing symbols to be used, account number, and CP options desired. Synonymous with *CP directory*.

VSE. Virtual storage extended.

VSE/POWER. Virtual Storage Extended/Priority Output Writers, Execution Processors, and Input Readers.

VTAM. Virtual telecommunications access method.

W

working set. The estimated number of pages of real storage that the virtual machine needs to execute.

workstation. An I/O device from which jobs can be submitted to a host system for processing, or to which output can be returned, or both.

wrap. The continuation of an operation from the maximum addressable location in storage to the first addressable location.

Z

3088. Refers to the IBM 3088 Multisystem Communications Unit, Models 1 and 2.

3270. Refers to a series of IBM display devices: for example, the IBM 3275, 3276 Controller Display Station, 3277, 3278, and 3279 Display Stations, the 3290 Information Panel, and the 3287 and 3286 printers. A specific device type is used only when a distinction is required between device types. Information about display terminal usage also refers to the IBM 3138, 3148, and 3158 Display Consoles when used in display mode, unless otherwise noted.

3350. Refers to the IBM 3350 Direct Access Storage device when used in native mode.

3800. Refers to the IBM 3800 Printing Subsystems. A specific device type is used only when a distinction is required between device types.

3812. Refers to the IBM 3812 Page Printer.

4224. Refers to the IBM 4224 Printer.

4245. Refers to the IBM 4245 Line Printer.

4248. Refers to the IBM 4248 Printer.

9370. Refers to a series of processors, namely the IBM 9373 Models 20 and 30, the IBM 9375 Models 40, 50, and 60, and the IBM 9377 Models 80 and 90.

Bibliography

This bibliography lists the books that provide additional information about your system.

RSCS Library

The following table lists the books in the RSCS library and their order numbers by task-oriented category.

Title	Order Number
Evaluation	
VM/RSCS: Licensed Program Specifications	GH24-5223
VM/RSCS: General Information	GH24-5218
Planning, Installation, and Service	
VM/RSCS: Planning and Installation	SH24-5219
VM/RSCS: Data Interchange Manager Installing, Managing, and Using	SH24-5248
End Use	
VM/RSCS: Operation and Use	SH24-5220
VM/RSCS: Reference Summary	SX24-5257
Customization	
VM/RSCS: Exit Customization	SH24-5222
Diagnosis and Reference	
VM/RSCS: Diagnosis Reference	SC24-5881
VM/RSCS: Messages and Codes	SH24-5221

VM/ESA Library

The following table lists books in the VM/ESA library that may be helpful when using RSCS.

Title	Order Number
VM/ESA Version 2 Release 4.0	
VM/ESA: General Information	GC24-5745
VM/ESA: VMSES/E Introduction and Reference	GC24-5837
VM/ESA: Service Guide	GC24-5838
VM/ESA: Planning and Administration	SC24-5750
VM/ESA: Performance	SC24-5782
VM/ESA: CP Command and Utility Reference	SC24-5773
VM/ESA: Group Control System	SC24-5757
VM/ESA: CMS User's Guide	SC24-5775
VM/ESA: CMS Command Reference	SC24-5776
VM/ESA: REXX/VM Reference	SC24-5770
VM/ESA: CMS Application Development Guide	SC24-5761
VM/ESA: CP Programming Services	SC24-5760
VM/ESA: Dump Viewing Facility	GC24-5853
VM/ESA: System Messages and Codes	GC24-5841
VM/ESA: CMS File Pool Planning, Administration, and Operation	SC24-5751
VM/ESA: Quick Reference	SX24-5290
CMS Utilities Feature	
VM/ESA: CMS Utilities Feature	SC24-5535

Other Related Books

The following table lists other books, outside the VM/Pass-Through Facility and VM/ESA libraries, that are helpful when using this book.

Title	Order Number
Transmission Control Protocol/Internet Protocol for VM	
TCP/IP for VM: Planning and Customization	SC24-5847
TCP/IP for VM: User's Guide	SC24-5848
TCP/IP for VM: Programmer's Reference	SC24-5849
TCP/IP for VM: Messages and Codes	GC24-5850
Network Job Entry	
Concepts and Protocols Overview	GG66-0224
Network Job Entry Formats and Protocols	SC23-0070
System Network Architecture	
Technical Overview	GC30-3073
Reference Summary	GA27-3136
Sessions Between Logical Units	GC20-1868
Other	
ACF/VTAM Programming	GC38-0286
GDDM: Installation and System Management	SC33-0152
IBM ESA/390 Principles of Operation	SA22-7201
IBM 3800 Printing Subsystem Programmer's Guide	SC26-3846
IBM 7171 ASCII Device Attachment Control Unit Reference Manual and Programming Guide	GA37-0021
4224 Printer Products and Program Directory	GC31-2550

Using the Master Index

The master index that is provided in this book it is created by merging the indexes of the individual books in the RSCS library.

An entry in a master index identifies a keyword or subject and is followed by one or more *master index book codes*. A master index book code is an abbreviation for one of the following RSCS book titles:

VM/RSCS: Diagnosis Reference	**DIAGNOSE**
VM/RSCS: Exit Customization	**EXIT**
VM/RSCS: Planning and Installation	**INSTALL**
RSCS Data Interchange Manager Installing, Managing, and Using	**INTER**
VM/RSCS: Messages and Codes	**MESSAGES**
VM/RSCS: Operation and Use	**OPERATE**

To use the master index, first think of a topic or subject you want to find and choose a keyword for that subject or topic. Next, find the keyword in the master index. The **last** group of letters in the entry is the master index book code. If a topic can be found in more than one book, the master index book codes will be separated by a comma. Then, go to that book's index and find the same entry; this entry will contain a page number to direct you to the information.

Master Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Readers' Comments

Virtual Machine
Remote Spooling Communications Subsystem
Networking
General Information
Version 3 Release 2

Publication No. GH24-5218-03

You may use this form to report errors, to suggest improvements, or to express your opinion on the appearance, organization, or completeness of this book.

Date: _____

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note

Report system problems to your IBM representative or the IBM branch office serving you.
U.S. customers can order publications by calling the IBM Software Manufacturing Solutions at **1-800-879-2755**.

In addition to using this postage-paid form, you may send your comments by:

FAX	1-607-752-2327	Internet	pubrcf@vnet.ibm.com
IBM Mail	USIB2L8Z@IBMMAIL	IBMLink	GDLVME(PUBRCF)

Would you like a reply? ___**YES** ___ **NO** If yes, please tell us the type of response you prefer.

___ Electronic address: _____

___ FAX number: _____

___ Mail: (Please fill in your name and address below.)

Name _____ Address _____

Company or Organization _____

Phone No. _____

Readers' Comments
GH24-5218-03

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60
International Business Machines Corporation
Information Development
1701 North Street
ENDICOTT NY 13760-5553

Fold and Tape | **Please do not staple** | Fold and Tape

GH24-5218-03

Cut or Fold
Along Line

File Number: S370/4300-39
Program Number: 5684-096

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GH24-5218-03

Spine information:

VM RSCS General Information *Version 3 Release 2*