

MERVA ESA Components



Workstation Based Functions

Version 4 Release 1

MERVA ESA Components



Workstation Based Functions

Version 4 Release 1

Note!

Before using this information and the product it supports, be sure to read the general information under “Appendix D. Notices” on page 205.

Second Edition, September 2001

This edition applies to Version 4 Release 1 of IBM MERVA ESA Components (5648-B30) and to all subsequent releases and modifications until otherwise indicated in new editions.

Changes to this edition are marked with a vertical bar.

This edition replaces SH12-6383-00.

© Copyright International Business Machines Corporation 1991, 2001. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	vii
Who Should Read This Book	vii

Summary of Changes	ix
-------------------------------------	-----------

Part 1. Overview of the MERVA ESA USE Feature **1**

Chapter 1. The Components of MERVA USE & Branch	3
Introduction	3
The Workstation Based USE Functions.	3
MERVA ESA Components	4
MERVA Base	4
MERVA Link or MERVA-MQI Attachment	4
SWIFT Link with USE	5
Telex Message Processing	6
MERVA USE & Branch Components	6
MERVA Base	6
MERVA Link	6
MERVA-MQI Attachment	6

Part 2. Connections via MERVA Link **7**

Chapter 2. Connecting MERVA ESA to the USE Functions via MERVA Link	9
Overview	9
Parameter Definitions	10
Customizing the Host Resources	14
Customizing VTAM Resources	15
Customizing CICS Resources	15
Customizing IMS Resources	17
Customizing APPC/MVS Resources	18
Customizing APPC/IMS Resources	18
Customizing MERVA ESA Resources for the USE Functions	19
Introduction and Troubleshooting	21
Step 1: Configuring the SNA Connection	22
Step 2: Configuring the LU 6.2 Session	23
Step 3: Configuring Outbound MERVA Link	24
Step 4: Configuring Inbound MERVA Link	25
Generating Traces	26
Customizing the Windows NT Based Components	28
Configuring Communications Server for Windows NT	28
Configuring Queues and Routing Conditions for USE Functions	38
Customizing the MERVA Link of MERVA USE & Branch	38
MERVA Link Conversation Security	39
Conversation Security Terms	40
Conversation Security for MERVA Link NT.	40
Parameter Relationships	43

Parameters for a MERVA Link Connection—CICS to Workstation	44
Parameters for a MERVA Link Connection—IMS to Workstation	49
Parameters for a MERVA Link Connection Independent of CICS or IMS.	52
Common Parameter for Telex and USE	57
Additional Parameters for USE	58
Additional Parameters for Telex	61

Chapter 3. Administering the USE Functions (MERVA Link) **63**

Verifying the USE Functions.	63
Verifying the MERVA Link Connection with MERVA USE & Branch	63
Verifying the SLS Service	64
Data Flow for the USE Functions	64
Data Flow of Pregenerated Session Keys for SLS	65
Data Flow of Session Keys Requests for SLS	65
Data Flow of USE Messages	67
Workstation Message Queues	70

Chapter 4. Telex with Workstation API Functions (MERVA Link) **71**

Installation and Customization of the Telex Functions	71
Customizing MERVA ESA Resources for the Telex Functions	71
Additional MERVA ESA Customizing Information	73
MERVA Link MFS User Exit EKAMU034	74
Data Flow for the Telex Functions	76
Functions for Outgoing Telexes.	76
Functions for Incoming Telexes.	78
Functions for Telex Transmission to MERVA USE & Branch	80
General Functions	80

Part 3. Connections via MERVA-MQI Attachment **83**

Chapter 5. Connecting MERVA ESA to the USE Functions via MERVA-MQI Attachment **85**

Overview	85
Parameter Definitions	86
Customizing the Host Resources	89
Customizing CICS Resources	90
Customizing IMS Resources	91
Customizing MQSeries Resources	92
Customizing MERVA ESA Resources for the USE Functions	103
Customizing the Windows NT Based Resources	106

	Configuring Windows NT	106
	Customizing MERVA USE & Branch Resources	106
	Customizing MQSeries Resources	107
	Configuring MERVA-MQI Attachment	110
	Message Security	111
	Parameter Relationships	112
	Parameters for a MERVA-MQI Attachment	
	Connection—MQSeries	112
	Parameters for a MERVA-MQI Attachment	
	Connection Independent of CICS or IMS	122
	Parameters for a MERVA-MQI Attachment	
	Connection—CICS to Workstation	128
	Parameters for a MERVA-MQI Attachment	
	Connection—IMS to Workstation	131

Chapter 6. Administering the USE Functions (MERVA-MQI Attachment) 133

	Verifying the USE Functions	133
	Verifying the MERVA-MQI Attachment	
	Connection and the SLS Services	133
	Data Flow for the USE Functions	134
	Data Flow of Pregenerated Session Keys for SLS	134
	Data Flow of Session Keys Requests for SLS	135
	Data Flow of USE Messages	137
	Workstation Message Queues	141

Part 4. Appendixes 143

Appendix A. Samples for Customizing Telex and USE Functions for MERVA Link on the Host System 145

	Samples for VTAM Definitions	145
	VTAM Application: CICS LU	145
	VTAM Application: APPC/MVS LU	145
	VTAM Application: APPC/IMS LU	146
	VTAM Logmode Table Entries	146
	VTAM PU and LU Definition for a PC Token	
	Ring Network	147
	VTAM PU and LU Definition for a PC with	
	SDLC Line	148
	Samples for Defining CICS Resources	148
	Defining CICS Resources using the CICS Macro	
	Level Interface	149
	Defining CICS Resources using the CSD Utility	150
	Sample for Defining IMS Resources	152
	Samples for Defining APPC/MVS Resources	152
	APPC/MVS LU in SYS1.PARMLIB	153
	APPC/MVS TP Profile	153
	Samples for Defining APPC/IMS Resources	154
	APPC/IMS LU in SYS1.PARMLIB	154
	APPC/IMS TP Profile	154
	Samples for Defining MERVA Link Resources	155
	EKAPT Header	155
	ASP Entry for Telex and USE	155
	MTP Entry for CICS	156
	MTP Entry for APPC/MVS	156
	MTP Entry for APPC/IMS	156
	Samples for Defining MERVA ESA Resources	157

	Message Format Service (MFS) Program Table	
	(DSLMPPT)	157
	Function Table for Telex and USE (DSLFNNTT)	158
	Logical Terminal Table for USE (DWLTT)	160
	Routing Table for USE (DWSRTSK)	160
	Routing Table for Telex (ENLRTLC)	160
	Routing Table EKARTTXU for Telex and USE	161
	Telex Only Scenario	161
	USE Only Scenario	163
	Sample Startup Jobs for MERVA Link	164
	Starting MERVA Link under CICS	164
	Starting MERVA Link under IMS	164

Appendix B. Samples for Customizing USE Functions for MERVA-MQI Attachment on the Host System 165

	Samples for Defining CICS Resources	165
	Defining CICS Resources using the CICS Macro	
	Level Interface	165
	Defining CICS Resources using the CSD Utility	166
	Sample for Defining IMS Resources	167
	Samples for Defining MQSeries Resources under	
	MVS	167
	DSLKCSQC (CICS only)	168
	DSLKCSQI (IMS only)	169
	DSLKCSQU	170
	Samples for Defining MERVA-MQI Attachment	
	Resources	173
	DSLKPUSE (MVS only)	173
	DSLKPUSV (VSE only)	174
	Samples for Defining MERVA ESA Resources	175
	Message Format Service (MFS) Program Table	
	(DSLMPPT)	175
	Function Table for USE (DSLFNNTT)	176
	Logical Terminal Table for USE (DWLTT)	178
	Routing Table for USE (DWSRTSK)	179
	Routing Table for Telex and USE (EKARTTXU)	179
	USE Only Scenario	188
	Sample Startup Jobs for MERVA-MQI Attachment	190
	Sample CICS Startup Job	190
	Sample IMS MPP Job	191

Appendix C. Samples for Customizing USE Functions for MERVA-MQI Attachment on the Workstation 193

	Samples for Defining MQSeries Resources under	
	Windows NT	193
	Samples for Defining MQSeries Resources for a	
	Connection from Windows NT to MVS	193
	Samples for Defining MQSeries Resources for a	
	Connection from Windows NT to VSE	197
	Samples for Defining MERVA-MQI Attachment	
	Resources	202
	Sample Configuration File for a Connection to	
	MVS	202
	Sample Configuration File for a Connection to	
	VSE	203

Appendix D. Notices 205

Trademarks	206	MERVA ESA Components Publications	221
Glossary of Terms and Abbreviations	209	Other IBM Publications	221
Bibliography.	221	S.W.I.F.T. Publications	222
MERVA ESA Publications	221	Index	223

About This Book

This book is written for users and administrators of the workstation based functions of the IBM licensed program Message Entry and Routing with Interfaces to Various Applications ESA Components Version 4 Release 1 (abbreviated to MERVA ESA in this book). The workstation based functions are installed on a workstation running under Microsoft Windows NT[®]. A connection must be established to allow the MERVA ESA system and the workstation to communicate with each other.

Note: Unless otherwise specified:

- The term *CICS* refers to CICS/VSE[®], CICS/MVS, and CICS/ESA[®].
- The term *IMS* refers to IMS/ESA[®].
- The term *MQSeries* refers to MQSeries[®] for OS/390[®], MQSeries for VSE/ESA[™], and MQSeries for Windows NT.
- The term *eNetwork Communications Server for Windows NT* refers to either Communications Server for Windows NT or eNetwork Personal Communications for Windows NT.
- The term MERVA USE & Branch refers to either MERVA USE & Branch for Windows NT or MERVA USE & Branch with SWIFT Link for Windows NT.

The workstation based functions are supplied with MERVA USE & Branch, and include:

- Workstation-based User Security Enhancements (USE)
- API functions to handle telex messages

Who Should Read This Book

You should read this book if you are involved in:

- Installing, customizing, and verifying any of the workstation based functions
- Administering the workstation based functions
- Tasks relating to the processing of telex messages
- Customizing the message processing and routing connections within your branch.

Summary of Changes

| This edition reflects the fact that the workstation based USE functions are now
| available via MERVA-MQI Attachment as well as via MERVA Link.

Part 1. Overview of the MERVA ESA USE Feature

This part provides a general overview of the USE functions for MERVA ESA.

It describes:

- The components that are required on the respective systems
- How to customize the systems to allow you to establish a connection between your MERVA ESA system and the components on the workstation.

Chapter 1. The Components of MERVA USE & Branch

This chapter provides an overview of the components necessary to establish the MERVA Link or MERVA-MQI Attachment connection between your MERVA ESA system and the USE functions on the workstation.

Introduction

The workstation based functions are made up of components of both MERVA ESA and MERVA USE & Branch.

The following MERVA ESA components are used:

- Base Functions of MERVA ESA, which includes the functions necessary to implement the Bilateral Key Exchange (BKE) and Secure Login and Select (SLS) services that make up the S.W.I.F.T. User Security Enhancements (USE)
- Telex Message Processing
- MERVA Link or MERVA-MQI Attachment
- SWIFT Link

The following components of MERVA USE & Branch are located on the workstation:

- MERVA Base, which includes:
 - The functions necessary to implement the BKE and SLS services of USE
 - An application programming interface (API) to allow you to develop your own applications and to handle telex messages
- MERVA Link or MERVA-MQI Attachment

The Workstation Based USE Functions

You use the workstation based USE functions to implement the BKE and SLS services. Figure 1 shows the components of MERVA ESA and MERVA USE & Branch that are required in this case.

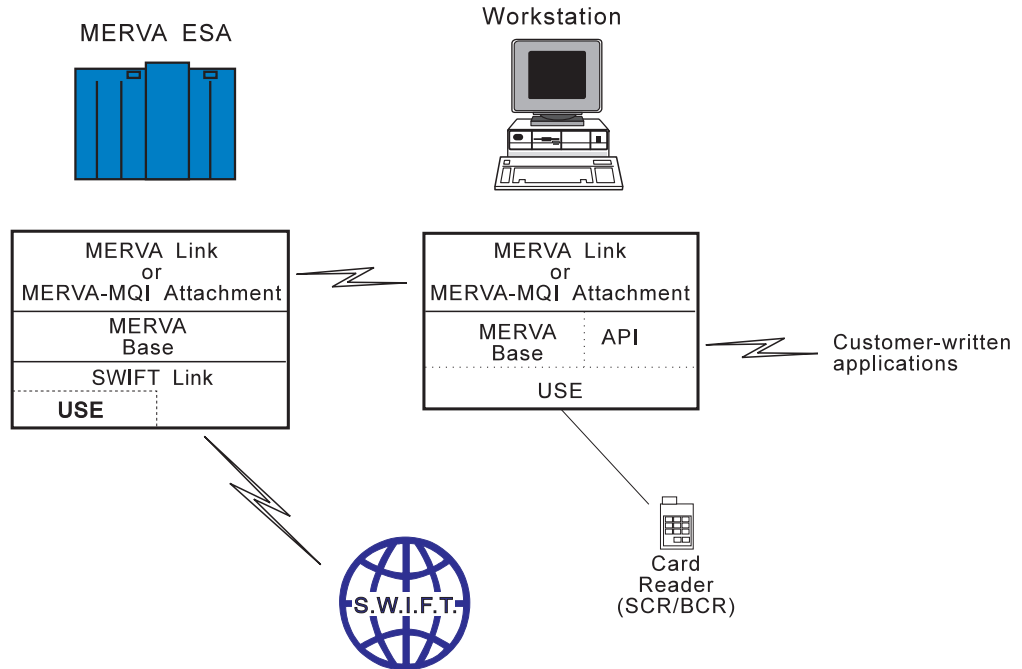


Figure 1. Components of MERVA ESA and MERVA USE & Branch Required for USE Functions

MERVA ESA Components

The following is a description of the components located in MERVA ESA that are required for the workstation based functions.

MERVA Base

MERVA ESA services are used by the workstation based functions, for example, the:

- Queue Management service for accessing messages in the MERVA ESA queues
- Routing facility to determine the message flow
- User File services for access control
- Journal service to put a record into, or to get a record from, the MERVA ESA journal.

MERVA Link or MERVA-MQI Attachment

Either MERVA Link or MERVA-MQI Attachment can be used to handle the communication between MERVA ESA and MERVA USE & Branch.

MERVA Link

The MERVA Link component of MERVA ESA can be used to communicate with the MERVA Link component of MERVA USE & Branch. The protocol existing between these two partners guarantees the integrity and security of messages.

In addition, the MERVA Link is used to process telex transmission acknowledgments.

The MERVA Link MFS user exit EKAMU133 is an essential element of the workstation based functions and must not be customized. It is responsible for:

- Instructing the MERVA Link that outgoing messages are to be handled as messages and not as status reports (by clearing the MERVA Link receipt return code)
- Creating a standard MERVA ESA MSGACK field from the receipt report returned by MERVA USE & Branch
- For telex messages, calling EKAMU034 to generate a unique message identifier

EKAMU133 allows the following formats to be specified in the partner table ASP definition for establishing the MERVA Link connection between MERVA ESA and MERVA USE & Branch:

Format Specification	Purpose
(MCB,TCOV,#)	MERVA Link connection for SWIFT and telex messages.
(,,W)	MERVA Link connection for SWIFT messages only.
(,ENLEXIT,P)	MERVA Link connection for telex messages only.

Note: It is recommended that you use format specification (MCB,TCOV,#). This enables you to handle a:

- Combined SWIFT and telex message exchange
- Dedicated SWIFT message exchange
- Dedicated telex message exchange

The routing table EKARTTXU included in MERVA ESA V4.1 supports the combined SWIFT and telex message exchange. When one of the other format specifications is used, the routing table also supports the dedicated SWIFT or dedicated telex message exchange respectively.

MERVA-MQI Attachment

The MERVA-MQI Attachment component of MERVA ESA can be used to communicate with the MERVA-MQI Attachment component of MERVA USE & Branch. Both partners communicate via MQSeries; this ensures message integrity and security.

The routing table EKARTTXU included in MERVA ESA V4.1 is used for message routing.

SWIFT Link with USE

SWIFT Link enables communication with the SWIFT network and performs all actions required by the SWIFT network. It can be adapted to the operational and functional requirements of the network. SWIFT Link supports the following SWIFT USE services:

SLS The SLS service replaces the paper-based Login and Select tables.

After entering a LOGIN or SELECT command on MERVA ESA, the command execution module generates a message requesting the session key from the card reader, and routes it via the appropriate MERVA Link or MERVA-MQI Attachment send queue to the workstation. The response from the workstation is routed via MERVA Link or MERVA-MQI Attachment back to a MERVA ESA queue. This invokes a program that processes the response. If the request is successfully answered, the LOGIN or SELECT is completed and sent to SWIFT.

BKE The BKE service enables the user to automatically generate the bilateral keys necessary for the authentication of financial messages.

The BKE exchange can be started from the workstation. It uses the following MERVA ESA components:

- MERVA Link or MERVA-MQI Attachment
- SWIFT Link

BKE messages, such as the MT 96x, are passed through MERVA ESA using appropriate routing tables. The generated keys are stored in the authenticator-key file on the workstation and a copy is held in the authenticator-key file of MERVA ESA. A database update is performed after completion of BKE.

Telex Message Processing

The Telex Message Processing Functions of MERVA ESA provide facilities for preparing messages in free format, in a standardized format agreed with correspondents, or in SWIFT format for telex transmission.

MERVA USE & Branch Components

The following is a description of the MERVA components located on the workstation that are involved in the workstation based functions.

MERVA Base

MERVA USE & Branch provides a set of base services that are used when running SWIFT Link, MERVA Link, or MERVA-MQI Attachment, for example message routing and audit logging. MERVA Base provides additional functions that let you work with messages in MERVA USE & Branch queues, for example:

- Display and print messages
- Display and print the diagnosis log
- Start or stop the automatic printing of messages
- Load messages from and unload messages to an external file
- Delete messages

MERVA Base also includes support for SWIFT USE, as well as functions to implement the SLS and BKE services.

MERVA Link

The services provided by the MERVA Link component of MERVA USE & Branch are a subset of those provided by the MERVA Link component of MERVA ESA.

The MERVA Link component of MERVA USE & Branch is the partner concerned with the exchange of messages with the MERVA Link component of MERVA ESA. In addition, MERVA Link supplies functions for monitoring and controlling the flow of messages to MERVA ESA.

MERVA-MQI Attachment

The MERVA-MQI Attachment component of MERVA USE & Branch exchanges messages, via MQSeries, with the MERVA-MQI Attachment component of MERVA ESA. The functions for monitoring and controlling the flow of messages are located in MQSeries.

How to configure the routing for MERVA-MQI Attachment is described in “Customizing MERVA USE & Branch Resources” on page 106.

Part 2. Connections via MERVA Link

Chapter 2. Connecting MERVA ESA to the USE Functions via MERVA Link

This chapter describes the steps you must take to establish a connection, using the MERVA Link component, between your MERVA ESA system and the components installed on the workstation.

Overview

To establish a MERVA Link connection between MERVA ESA and MERVA USE & Branch, you need to customize the following:

Transaction Monitor Subsystem	CICS or IMS TM
Communication Software	VTAM [®] and Communications Server, APPC/MVS, or APPC/IMS
MERVA-to-MERVA connection	The MERVA Link components of MERVA ESA and of MERVA USE & Branch

Figure 2 illustrates the components involved for CICS.

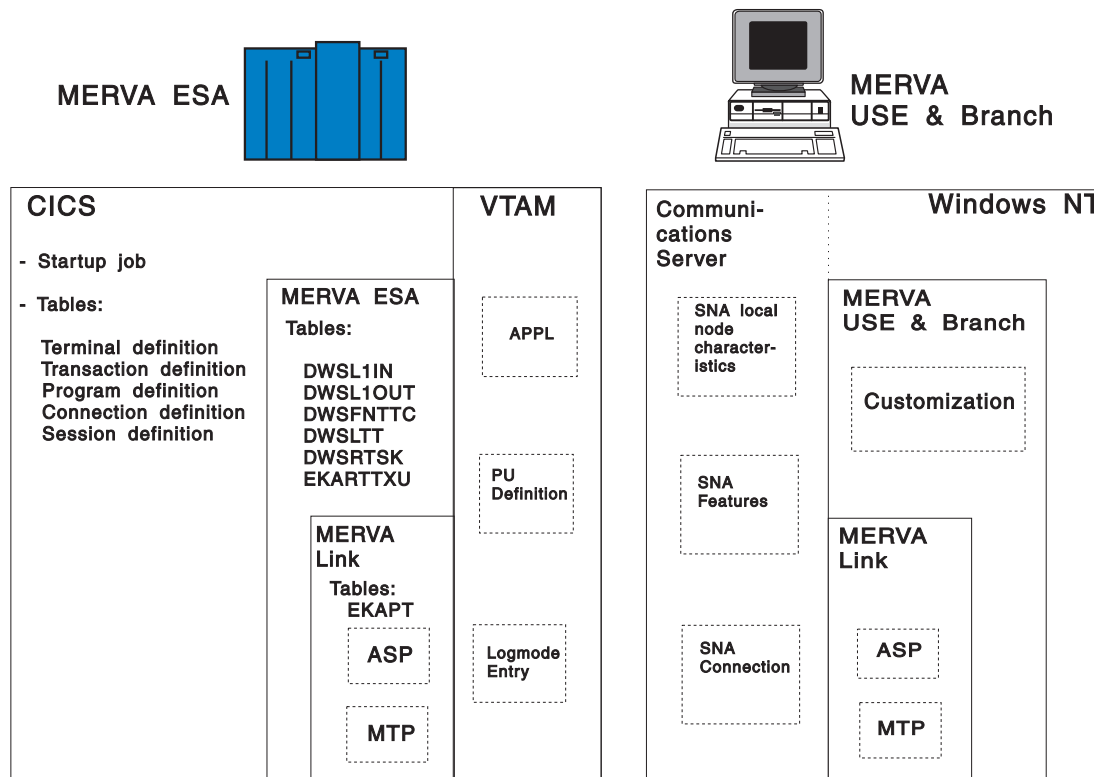


Figure 2. What You Need to Customize for CICS

Figure 3 illustrates the components involved for IMS.

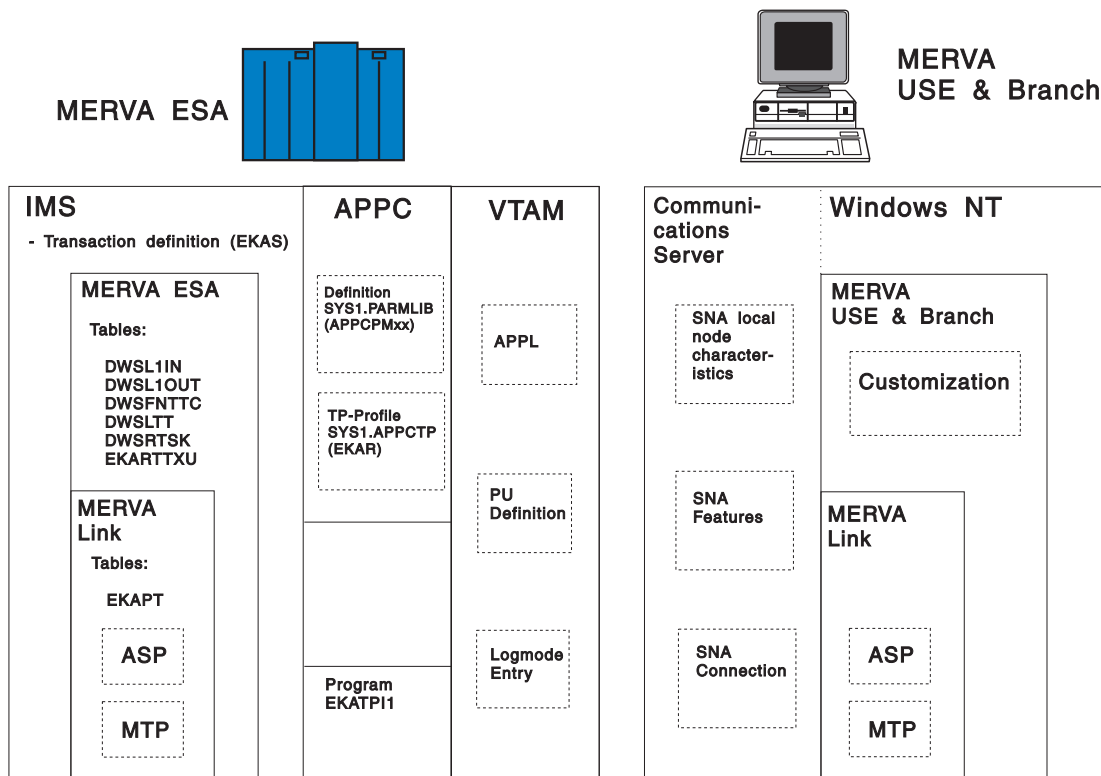


Figure 3. What You Need to Customize for IMS

Parameter Definitions

The workstation based telex and USE functions require a connection from MERVA ESA on the host to the MERVA USE & Branch on the workstation. This connection is established by the MERVA Link components of MERVA ESA and MERVA USE & Branch. The communication between the host system and the workstation is based on Advanced Program-to-Program Communication (APPC).

The parameters used to set up a connection between the host system and the workstation system identify resources or specify processing options. The identified resources are located either on the host or on the workstation. The terms *Host* and *PC* are used in the parameter names when the location of the resource is not evident.

Host Identifies a resource of MERVA ESA, MERVA Link ESA, CICS, IMS, APPC/MVS, or any other subsystem that executes in the host environment.

PC or Workstation

Identifies a resource on which Windows NT, eNetwork Communications Server for Windows NT, DB2 Universal Database®, MERVA USE & Branch, and MERVA Link run.

Refer to *MERVA for ESA Concepts and Components* for a description of the MERVA Link resources and its terminology.

The next tables list and describe all customization parameters that apply in a MERVA ESA installation including the workstation based functions. They also

contain page references to the tables in “Parameter Relationships” on page 43. Thus you can see in which subsystem customization a specific parameter is used.

The parameters are grouped as follows:

Parameter	Group
a..	VTAM and Communications Server customization (see Table 1)
b..	MERVA ESA and MERVA USE & Branch customization (see Table 2 on page 12)
c..	CICS customization (see Table 3 on page 13)
d..	IMS customization including APPC/MVS and APPC/IMS (see Table 4 on page 14)

Table 1. Customization Parameters for VTAM and Communications Server for Windows NT

Parameter	Name	Description	Page
a1	PC Network ID	SNA Network ID of the network in which the workstation resides.	52
a2	PC Node Name	Physical Unit (PU) name of the workstation in the SNA network. The name that other nodes in your network use to address the workstation node.	52
a3	PC Local Node ID	Node identifier of the workstation in the SNA network. The last 5 digits of the exchange identification (XID), which you need to activate your workstation as a PU 2.1 in the SNA network.	52
a4	LAN Destination Address	Token Ring address of the communications controller (for example, 37x5) that is the gateway to the SNA network. (This parameter is not necessary for connections with an SDLC line.)	52
a5	Host Network ID	SNA network ID of the network in which the VTAM system that is used by MERVA ESA is defined.	52
a7	CICS LU Name	VTAM application name of the CICS system (APPLID) under which MERVA ESA is running. If you are running MERVA ESA under IMS, see parameter d3.	44
a8	PC LU Name	Independent Logical Unit (LU 6.2) of the workstation system.	44
a9	PC Receiving TP Name	Name of the receiving APPC Transaction Program (TP) in MERVA Link.	52
a10	PC Receiving TP File Name	Full path and file name of the receiving TP in MERVA Link.	53
a12	SNA Mode Name	Name of a VTAM Logon Mode Entry for a connection from CICS or IMS to workstation. It is defined in VTAM and in the Communications Server.	45
a13	Symbolic destination name	The name of a side information profile that contains values for the following parameters: <ul style="list-style-type: none"> • Partner LU alias (a7 and d3) • Mode name (a12 and d4) • TP name (c3) <p>This profile is defined using the product that provides the APPC service, and offers a convenient way to uniquely identify a side.</p>	53

Table 2. Customization Parameters for MERVA ESA and MERVA USE & Branch

Parameter	Name	Description	Page
b1	Host MERVA Link Local Node Name	The name of the local MERVA Link Node defined in MERVA Link ESA.	53
b2	Host ASP Name	The name of the MERVA Link Application Support Process (ASP) that supports MERVA USE & Branch on the workstation. It is defined in MERVA Link ESA.	53
b3	Workstation ASP Name	The name of the MERVA Link Application Support Process (ASP) that supports MERVA ESA applications (USE and Telex) on the host. It is defined in MERVA Link on the workstation.	54
b4	Host Internal MTP Name	Internal name of the Message Transfer Process (MTP) in MERVA Link ESA. This MTP is associated with the Host ASP.	54
b5	Host External MTP Name	External name of the Message Transfer Process (MTP) in MERVA Link ESA.	54
b6	Workstation MTP Name	Name of the Message Transfer Process (MTP) in MERVA Link that is associated with the workstation ASP.	54
b7	Host ASP Send Queue	Name of the MERVA Link Send Queue in MERVA ESA that is owned by the Host ASP. This queue must be defined in the MERVA ESA Function Table.	58, 61
b8	Host ASP Control Queue	Name of the MERVA Link Control Queue in MERVA ESA that is owned by the Host ASP. This queue must be defined in the MERVA ESA Function Table.	55
b9	Workstation ASP Send Queue	Name of the MERVA Link send queue owned by the workstation ASP.	58, 61
b10	Workstation ASP Receive Queue	Name of the MERVA Link receive queue owned by the workstation ASP.	59
b11	Host ASP Security Options	Parameter in MERVA Link ESA that specifies whether or not the messages processed by the Host ASP must be authenticated and encrypted.	55
b12	USE Workstation Name	<p>Defines the name of the USE workstation when SLS is used. Specify 1 to 9 characters conforming to the rules for SWIFT addresses. If you specify less than 9 characters, the name is padded with Xs.</p> <p>This name is used when using SLS connected mode for the routing of a single session key request to a MERVA Link send queue for sending to the USE workstation. A single session key request is a SWIFT message type 999 with a special format handled by the SWIFT Link programs. The name of the USE workstation is used in the application header as destination address and can be evaluated during the routing of this message.</p>	59
b13	Host Pregenerated Session Key Queue for GPA	Name of the MERVA ESA queue that is used for storing of the pregenerated session keys for the LOGIN function of the master logical terminal (GPA).	59
b14	Host Pregenerated Session Key Queue for FIN	Name of the MERVA ESA queue that is used for storing of the pregenerated session keys for the SELECT function of the financial application (FIN).	59
b15	Host ASP User Exit	Number or name of the MERVA ESA MFS user-exit program EKAMU133 defined in the MERVA ESA MFS Program Table (DSLMPPT). EKAMU133 supports the MERVA ESA V4 USE feature in MERVA Link ESA.	55

Table 2. Customization Parameters for MERVA ESA and MERVA USE & Branch (continued)

Parameter	Name	Description	Page
b16	Host Message Format	ASP parameter in MERVA Link ESA that specifies the message transmission format to be used between MERVA Link ESA and MERVA Link on the workstation.	55
b17	Host Session Key Request Routing Table	Name of the MERVA ESA routing table which controls the routing for a session key request. The session key requests are routed to the MERVA Link send queue (parameter b7).	60
b18	Host ASP Routing Table	Name of the MERVA ESA routing table which controls the routing of all messages handled by MERVA Link ESA. This includes the MERVA Link internal messages, the SLS and BKE messages received from MERVA USE & Branch, and the telex messages sent to MERVA USE & Branch.	57
b19	Host Telex Station Command Routing Table	Name of the MERVA ESA routing table which controls the routing of the telex station commands and responses. The commands are routed to the MERVA Link send queue (parameter b7).	61
b20	Host ASP Send Queue Cluster	Names of the send queues in MERVA Link ESA. The queues must be defined in the MERVA ESA Function Table DSLFNNT.	62
b21	Host ASP Incoming Report Routing Options	Parameter in MERVA Link ESA that specifies local routing options for incoming status reports. The second subparameter specifies the name of the MERVA Link ACK Wait Queue. This queue must be defined in the MERVA ESA Function Table.	62
b22	Host ASP Journaling Options	Parameter in MERVA Link ESA that specifies MERVA Link journaling options.	56
b23	Host Master Logical Terminal Name	Name of the master logical terminal used for the LOGIN function (GPA).	60
b24	MERVA Link Local Node Name	The name of the local MERVA Link Node defined in MERVA Link on the workstation.	56

Table 3. CICS Customization Parameters

Parameter	Name	Description	Page
c1	CICS Partner System ID	Identification of the connection to the workstation in CICS.	46
c2	CICS ISC Support	The Inter-System Communication support (ISC) must be included in CICS by specifying ISC=YES to enable APPC with MERVA Link on the MERVA USE & Branch.	46
c3	Host Receiving Transaction ID	CICS transaction ID of the MERVA Link receiving transaction (for example, EKAR). This transaction ID is associated with the Host Receiving TP Name.	46
c4	Host Receiving TP Name	The name of the MERVA Link program that handles inbound APPC conversations in the CICS environment (EKATR10).	47
c5	Host Sending Transaction ID	CICS transaction ID of the MERVA Link sending transaction (for example, EKAS). This transaction ID is associated with the Host Sending Program Name.	47
c6	Host Sending Program Name	The name of the MERVA Link program that handles outbound messages in the MERVA ESA CICS and in the MERVA ESA IMS environment (EKAAS10).	48

Table 4. IMS Customization Parameters Including APPC/MVS and APPC/IMS

Parameter	Name	Description	Page
d1	Host Sending TP Name	Name of the MERVA Link APPC TP that supports APPC/MVS and APPC/IMS in the MERVA ESA IMS environment for outbound conversations. This parameter must be specified when the default MERVA Link APPC TP (EKATS10) must not be used. The latter program can be used only in the CICS environment for APPC.	49
d2	Workstation LU Name	APPC Logical Unit (LU) of the workstation (the same as parameter a8).	49
d3	APPC/MVS LU Name	Name of an APPC/MVS Logical Unit (LU). APPC/MVS LUs are defined in VTAM as VTAM applications. This parameter is applicable only in the MERVA ESA IMS environment. The APPC/MVS LU that must be selected depends on the version of your IMS system. For IMS/VS Version 2 or IMS/ESA Version 3 you must select the APPC/MVS System Base LU that is associated with the APPC/MVS transaction scheduler ASCH. For IMS/ESA Version 4 you must select the APPC/IMS Base LU (also one of the APPC/MVS LUs) that is associated with the APPC/IMS transaction scheduler.	49
d4	SNA Mode Name	Name of a VTAM Logon Mode Entry for a connection from APPC/MVS or APPC/IMS to the workstation (the equivalent of a12). It is defined in VTAM and in the Communications Server.	50
d5	Host Sending Transaction ID	IMS transaction ID for the MERVA Link sending transaction (for example, EKAS). This transaction ID is associated with the Host Sending Program Name.	50
d6	Host Sending Program Name	The name of the MERVA Link program that handles outbound messages in the MERVA ESA CICS and in the MERVA ESA IMS environment (EKAAS10).	50
d7	Conversation Security	This parameter specifies the required security level of a conversation in the APPC/MVS environment.	50
d8	APPC/MVS TP Profile Name	The name of the APPC/MVS Transaction Program (TP) Profile that describes the execution environment for the MERVA Link receiving APPC TP EKATPI1 in the APPC/MVS or in the APPC/IMS environment. The TP profiles for APPC/MVS and APPC/IMS are different because the TP execution environments are different. The sample TP profile name for APPC/MVS is EKAR. The sample TP profile name for APPC/IMS is EKARI4.	51
d9	Host Receiving TP Name	The name of the MERVA Link program that handles inbound APPC conversations in the APPC/MVS and APPC/IMS environments (EKATPI1).	51

Customizing the Host Resources

This section describes how to adapt your host system's parameters for use with the MERVA USE & Branch feature. The host parameters cover the following areas:

- VTAM
- CICS
- IMS
- APPC/MVS
- APPC/IMS

- MERVA ESA

Customizing VTAM Resources

When customizing VTAM resources, you must specify the following parameters:

- PC Node Name (**a2**)
- PC Local Node ID (**a3**)
- LAN Destination Address or Remote Link Address (**a4**)
- Host Network ID (**a5**)
- CICS LU Name (**a7**)
- PC LU Name (**a8** and **d2**)
- SNA Mode Name (**a12** and **d4**)
- APPC/MVS LU Name (**d3**)

For the APPC conversation between MERVA ESA and MERVA USE & Branch, an active LU 6.2 session must be available. Following definitions are necessary for that session:

- Define the workstation as a physical unit (PU macro). This enables the workstation to be activated as a PU 2.1 node in the SNA network. Specify the name of the PU in parameter **a2**.
- If your workstation resides on a Token Ring network and you have defined the IDNUM parameter in the PU macro, specify its value as your local node ID in parameter **a3**.
- The PC LU Name is specified in parameter **a8** (MERVA ESA CICS) and in parameter **d2** (MERVA ESA IMS).

If MERVA ESA executes in the CICS environment, the LU 6.2 session is established between the CICS LU (**a7**) and the workstation local LU defined as an independent LU in the PC Communications Server. Specify the PC LU Name in parameter **a8**.

If MERVA ESA executes in the IMS environment, the LU 6.2 session is established between an APPC/MVS LU (**d3**) and the workstation local LU. The applicable APPC/MVS LU Name is either the name of the APPC/MVS System Base LU or the name of the APPC/IMS Base LU. The latter LU must be used in the APPC/IMS environment.

Specify the Workstation LU Name in parameter **d2**.

- Specify the appropriate logmode entries (**a12** and **d4**) for the APPC conversation. Figure 28 on page 147 shows sample logmode entries for an APPC connection in the CICS and IMS environments.
- The network ID (**a5**) of MERVA ESA must be specified in the start option "NETID=..." when VTAM is started.
- The LAN Destination Address respectively Remote Link Address (**a4**) must be defined in the network control program (NCP) of the control unit used. This parameter is only necessary when the connection to the host system is established using a Token Ring network.

Customizing CICS Resources

When customizing CICS resources, you work with the following parameters:

- CICS LU Name (**a7**)
- PC LU Name (**a8**)
- SNA Mode Name (**a12**)
- Host ASP User Exit (**b15**)
- Host Session Key Request Routing Table (**b17**)

- Host ASP Routing Table (**b18**)
- Host Telex Station Command Routing Table (**b19**)
- CICS Partner System ID (**c1**)
- CICS ISC Support (**c2**)
- Host Receiving Transaction ID (**c3**)
- Host Receiving TP Name (**c4**)
- Host Sending Transaction ID (**c5**)
- Host Sending Program Name (**c6**)

Note: In the following description, *xx* identifies the CICS release level. For example, EKACSD41 refers to CICS Version 4 Release 1. The identifier *xx=21* is an exception. It is used only for macro level CICS resource definitions which are also valid in a higher release level of CICS.

Do the following:

1. Ensure that your CICS system includes the ISC support (parameter **c2**) by the appropriate parameter in the CICS startup job. The CICS LU Name (**a7**) must also be specified in the CICS startup job. A sample is shown in “Starting MERVA Link under CICS” on page 164.
2. Modify the following tables in your CICS system:

CICS Transaction Definition

The following local MERVA Link transactions must be defined for CICS:

EKAS	The MERVA Link Sending ASP transaction is assigned to program EKAAS10 (c5 and c6).
EKAR	The MERVA Link Receiving MTP transaction is assigned to program EKATR10 (c3 and c4).

The copy book EKAPCT21, which is shipped with your MERVA ESA system, contains sample values for these parameters. You can include the copy book in your program control table (PCT). Figure 31 on page 149 shows a sample CICS program control table for MERVA Link ESA.

Alternatively, you can use the copy book EKACSD xx and the offline utility program DFHCSDUP to generate the necessary table entries (see Figure 34 on page 151).

In general, these entries are already included in the customization of the MERVA ESA base product.

CICS Program Definition

The following programs must be defined for CICS:

EKAPT	MERVA Link partner table.
EKAAS10	Sending ASP program (c6).
EKATR10	Receiving MTP program (c4).
EKAMU034	User-exit program to generate unique message identifiers for telex messages.

EKAMU133	User-exit program to handle SWIFT and MERVA for Workstation telex acknowledgments and negative acknowledgments (b15).
EKARTTXU	Routing table for the Telex and USE functions to exchange messages with the workstation (b18).
ENLR2TLC	Routing table for the Telex functions to send commands to and to receive responses from the workstation (b19).
DWSRTSK	Routing table for the USE functions to request session keys from the workstation (b17).
DWSLTT	Logical terminal table.

Note: This is not a complete list of all mandatory MERVA Link programs. The complete list can be found in copy book EKAPPT21.

The copy books DWSPPT21, EKAPPT21, and ENLPPT21, which are shipped with your MERVA ESA system, contain sample values for these parameters. You can include the copy books in your processing program table (PPT).

Figure 32 on page 149 shows a sample CICS processing program table.

Alternatively, you can use the copy books DWSCSDxx, EKACSDxx, and ENLCSDxx and the offline utility program DFHCSDUP to generate the necessary table entries (see Figure 34 on page 151).

In general, these entries are already included in the customization of the MERVA ESA base product.

CICS Connection Definition

An APPC connection between CICS and the workstation must be defined in CICS. This is either done in the CICS terminal control table (TCT), or using Resource Definition Online (RDO) transactions, or with the offline utility DFHCSDUP.

For the LU name (**a8**) of the workstation and the mode name (**a12**), use the names specified in “Customizing VTAM Resources” on page 15.

Choose CICS Partner System ID (**c1**) and create the corresponding CICS resources.

For details, refer to either Figure 33 on page 150 or Figure 35 on page 152 and the *CICS/ESA V4.1 Resource Definition Guide*.

Customizing IMS Resources

When customizing IMS resources, you work with the following parameters:

- Host Sending Transaction ID (**d5**)
- Host Sending Program Name (**d6**)

Define the Host Sending Transaction ID (parameter **d5**) as an IMS transaction (for example, EKAS). The Host Sending Program Name (parameter **d6**) that must be assigned to the transaction **d5** is EKAAS10. This program supports the message transfer to a partner system using the MERVA Link message integrity protocol.

Figure 36 on page 152 shows a sample IMS transaction definition.

Customizing APPC/MVS Resources

When customizing APPC/MVS resources, you work with the following parameters:

- APPC/MVS LU Name (**d3**)
- APPC/MVS TP Profile Name (**d8**)
- Host Receiving TP Name (**d9**)

Do the following:

1. Define an APPC/MVS LU in VTAM (VTAM APPC application). Specify the characteristics of that LU in an LUADD statement in an APPCPMxx member of the SYS1.PARMLIB. Define this LU as the APPC/MVS System Base LU.

Parameter **d3** is the ACBNAME parameter defined in the VTAM application entry of the APPC/MVS LU (see Figure 26 on page 146).

Figure 37 on page 153 shows a sample APPC/MVS LU definition.

2. Define a TP profile for the receiving transaction (parameter **d8**). The sample APPC/MVS TP Profile Name in the APPC/MVS environment is EKAR. The APPC/MVS utility program ATBSDFMU or an online dialog can be used to generate an APPC/MVS TP profile.

A message from a workstation causes the start of a job in an MVS address space that is controlled by APPC/MVS. This job executes the program EKATPI1 (parameter **d9**). EKATPI1 is the Host Receiving TP Name.

Figure 38 on page 153 shows a sample APPC/MVS TP profile definition.

Customizing APPC/IMS Resources

APPC/IMS is supported by IMS/ESA and MVS/ESA™. APPC/IMS is based on the APPC/MVS subsystem of MVS/ESA. In addition to the functions provided by APPC/MVS, APPC/IMS allows to schedule inbound conversations (receiving transactions) in an IMS Message Processing Region (MPR). APPC/IMS uses a proprietary APPC/MVS task scheduler for that purpose.

When customizing APPC/IMS resources, you work with the following parameters:

- APPC/MVS LU Name (**d3**)
- APPC/MVS TP Profile Name (**d8**)
- Host Receiving TP Name (**d9**)

Do the following:

1. Include the APPC/IMS support in your IMS system. Start IMS with the parameter APPC=Y.
2. Define an APPC/MVS LU in VTAM (VTAM APPC application). Specify the characteristics of that LU in an LUADD statement in an APPCPMxx member of the SYS1.PARMLIB. Define this LU as the Base LU that is associated with the APPC/IMS transaction scheduler. This LU is called the APPC/IMS Base LU.

Parameter **d3** is the ACBNAME parameter defined in the VTAM application entry of the APPC/IMS LU (see Figure 26 on page 146).

Figure 39 on page 154 shows a sample APPC/IMS LU definition.

3. Define a TP profile for the receiving transaction (parameter **d8**). The sample APPC/MVS TP Profile Name in the APPC/IMS environment is EKARI4. The APPC/MVS utility program ATBSDFMU or an online dialog can be used to generate an APPC/MVS TP profile.

A message from a workstation causes the start of a transaction in an IMS MPR. This transaction executes the program EKATPI1 (parameter **d9**). EKATPI1 is the Host Receiving TP Name.

Figure 40 on page 154 shows a sample APPC/IMS TP profile definition.

Customizing MERVA ESA Resources for the USE Functions

When customizing MERVA ESA resources, you work with the following parameters:

- PC LU Name (**a8** and **d2**)
- PC Receiving TP Name (**a9**)
- Host MERVA Link Local Node Name (**b1**)
- Host ASP Name (**b2**)
- Workstation ASP Name (**b3**)
- Host Internal MTP Name (**b4**)
- Host External MTP Name (**b5**)
- Workstation MTP Name (**b6**)
- Host ASP Send Queue (**b7**)
- Host ASP Control Queue (**b8**)
- Host ASP Security Options (**b11**)
- USE Workstation Name (**b12**)
- Host Pregenerated Session Key Queue for GPA (**b13**)
- Host Pregenerated Session Key Queue for FIN (**b14**)
- Host ASP User Exit (**b15**)
- Host Message Format (**b16**)
- Host Session Key Request Routing Table (**b17**)
- Host ASP Routing Table (**b18**)
- Host ASP Journaling Options (**b22**)
- Host Master Logical Terminal Name (**b23**)
- CICS Partner System ID (**c1**)
- Host Sending Transaction ID (**c5** and **d5**)
- Host Sending TP Name (**d1**)
- APPC/MVS LU Name (**d3**)
- SNA Mode Name (**d4**)

The following MERVA ESA tables are affected:

- DSLFNNT
- DSLMPTT
- DWSLTT
- DWSRTSK
- EKAPT
- EKARTTXU

Do the following:

1. Define the parameters of an ASP and an MTP entry in your MERVA Link partner table EKAPT. These PT entries describe the USE application and the

connection to the workstation in MERVA Link ESA. The source member EKAPTTXU, which is supplied with the MERVA ESA base product, contains sample PT generation statements.

For the LU name (**a8** and **d2**) of the workstation and the mode name (**d4**) use the values specified in “Customizing VTAM Resources” on page 15.

EKAOSVR is the PC Receiving TP Name that must be specified in the MTP entry (parameter **a9**).

There are default values for the Host MERVA Link Local Node Name (**b1**), Host ASP Name (**b2**), Workstation ASP Name (**b3**), Host Internal MTP Name (**b4**), Host External MTP Name (**b5**), and Workstation MTP Name (**b6**). The values are the same as those in the default configuration in MERVA ESA V4 USE feature. You can also choose your own values.

Defaults for the MERVA Link Send Queue (**b7**) and MERVA Link Control Queue (**b8**) are supplied. If you change these values, change the function table too.

If you want your generated session keys to be encrypted during the transfer from the workstation to the host, specify ENCRYPT in the parameter **b11** (Host ASP Security Options).

Parameter **b15** includes the exit number 7133 of the user-exit program EKAMU133, which passes SWIFT acknowledgment-related data from MERVA ESA to the MERVA ESA V4 USE feature. This program and the related entries in the MFS program table DSLMPTT are available when MERVA ESA is installed.

The format specification FORMAT=(MCB,TCOV,#) (**b16**) supports both the telex and the USE message exchange.

The defaults specified for the Host ASP Journaling Options (**b22**) result in the journaling of incoming and outgoing messages. There are no incoming reports that could be journaled. You can choose your own values.

For the sending transaction ID of MERVA ESA (**c5** and **d5**), use the names specified in “Customizing CICS Resources” on page 15 and “Customizing IMS Resources” on page 17.

If your MERVA system is running under:

- CICS, the system identifier (**c1**) appears in the statement LINK=(APPC,**c1**) in the MERVA Link partner table. For **c1**, use the value specified in “Customizing CICS Resources” on page 15.
- IMS, the LINK statement is LINK=(APPC,**d2,d4**). For **d2** and **d4**, use the values specified in “Customizing VTAM Resources” on page 15.

For MERVA ESA IMS, you must add the parameter MTPCI=(EKATPO1) to the MTP generation macro (parameter **d1**) of the partner table.

For APPC/IMS, you must add the parameter LOCLU=**d3** to the MTP generation macro and specify the name of the APPC/IMS Base LU as parameter **d3**. In the APPC/MVS environment the APPC/MVS System Base LU is used by default as the local LU for outbound conversations.

Modify the PC LU Name **a8** (DEST=(PCLU00,...)).

Assemble and link-edit your MERVA Link partner table. The load module name must be EKAPT.

Examples are shown in:

- Figure 42 on page 156

- Figure 43 on page 156
- Figure 44 on page 156
- Figure 45 on page 157

For more information, refer to the *MERVA for ESA Macro Reference*.

2. Add the new functions (queues) in your function table (DSLFNNTT).
An example of a set of functions that is sufficient for USE is included in the copy book DWSFNNTC, which is shipped with your MERVA ESA system.
3. Check that you have installed the new routing tables DWSRTSK (**b17**) and EKARTTXU (**b18**) in your load library:
 - A reference to parameter **b17** must be made in the logical terminal table (DWSLTT).
 - A reference to parameter **b18** must be made in the function table copy book DWSFNNTC.
4. Change the logical terminal table. First modify the name of the master logical terminals (**b23**) according to the names used in your installation. The modified master logical terminal names must also be used in the routing table EKARTTXU (**b18**). You should check, in addition, whether these names are used in the routing tables specified by the parameters ROUTIN and ROUTOUT. There are new parameters for the USE functions, for example, TFLAG, ROUTSK (**b17**), USENAME (**b12**), and SKEYQ (**b13** and **b14**):
 - Set TFLAG equal to SLS.
 - Set parameter **b17** equal to DWSRTSK, for example, to determine the routing table for the SLS request.
 - Specify a name for USENAME **b12** (for example, USEMERVA2). Parameter **b12** is a routing condition (see “Routing Table for USE (DWSRTSK)” on page 179) to determine the send queue to route the LOGIN or SELECT request to. The send queue identifies the MERVA Link connection. This connection determines the workstation that provides the session keys from the assigned card reader.
 - Assign your session key queue name to the parameter SKEYQ (for example, **b13** = SLSGPA and **b14** = SLSFIN). This is also defined in the function table. The session key queue is filled with pregenerated session keys when a session key preload request is started from the workstation.

Assemble and link-edit the logical terminal table. The program name must be DWSLTT. If you specify a different name, it must be specified in the SWIFT Link customization parameter module DWSPRM using the DWSPARM macro. See the examples in Table 9 on page 58, “Logical Terminal Table for USE (DWSLTT)” on page 178, and the *MERVA for ESA Macro Reference*.

Introduction and Troubleshooting

This section contains a brief overview of SNA APPC, and explains the basics of how to use the SNA parameters described in “Parameter Definitions” on page 10.

The following picture shows the layers involved when using MERVA Link with SNA. It does not show the lower levels such as Data Link Control (DLC) and Path Control, because you are not interested in these layers when customizing a MERVA Link connection.

As shown in Figure 4, there are four layers. Two of the layers, ASP/MTP and the MERVA Link Node, reside in MERVA; the other two are part of SNA.

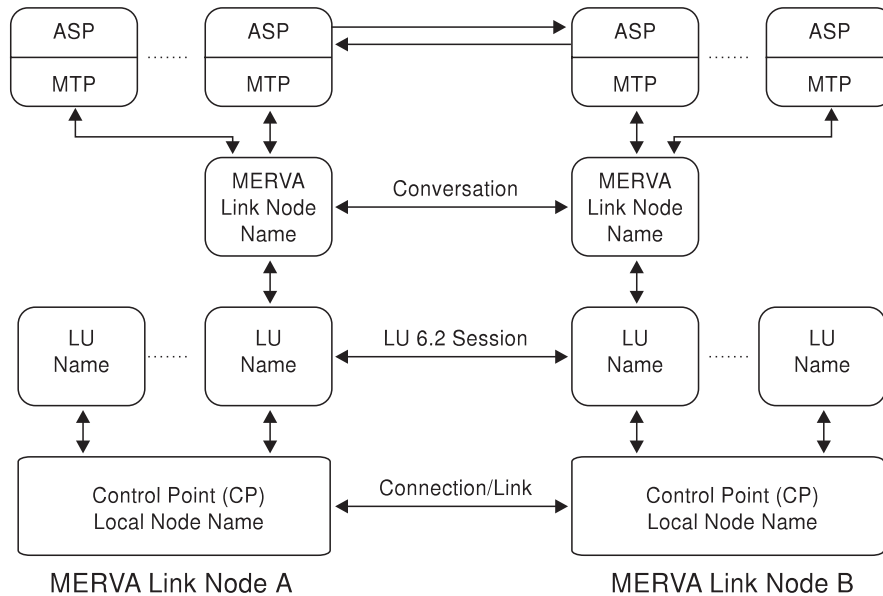


Figure 4. Flow of Session Key Request Messages for SLS and Pregenerated Session Keys

If you start to configure a MERVA Link connection, it is a good idea to proceed from the bottom to the top. That means building up the SNA connection (the SNA link) first, continuing with the SNA session, and finally building up the MERVA Link conversation. If a lower layer does not work correctly, all layers above it will also fail.

Before you start with the configuration, collect all necessary parameters for the customization process. It's very unlikely that the CICS or VTAM parameters will be changed for your NT workstation. Usually, you have to gather the parameters for the NT workstation from the host side and adapt to them. Each of the following chapters starts with a list of necessary parameters that you need to know about before you configure the appropriate part. It's a good idea to write them down on a configuration sheet, as this lowers the possibility of errors and makes things easier to sort out.

Note: This section describes a peer-to-peer connection to the host. If your network has an APPN Network Node Server, you can use the APPN features, in fact, this is highly recommended. However, the configuration of an APPN connection is different from the configuration steps described below.

Step 1: Configuring the SNA Connection

Necessary parameters:

- PC Network ID (a1)
- PC Node Name or Control Point name (a2)
- PC Local Node ID (or XID) (a3)
- LAN Destination Address (a4)
- Host Network ID (a5)
- Host Node Name or Adjacent CP Name (a6)

The above parameters are the ones you need to define a SNA link (or connection) to the communications controller of the host system. Get these parameters from the network administrator of the host system.

Follow the steps described in the sections:

- “Node Configuration” on page 28
- “Device Configuration” on page 29
- “Connection Configuration” on page 30

All these parameters must be set in the SNA Node Configuration program of Communications Server or Personal Communications. You don’t have to customize anything in MERVA for this first step.

You can check this configuration immediately:

1. In the SNA Node Configuration program, save the configuration (use ‘Save as...’ if this is a new configuration), then exit the configuration program.
2. In the SNA Node Operations program, start SNA with the configuration you just created.
3. Select the display of the links. You should see one link, and it should become active after a few seconds.

If the link does not become active, you made an error during the configuration of one of the necessary parameters described above. The host communications controller knows the Network ID, Node Name and Node ID of the NT workstation and rejects any connection attempt with wrong parameters defined in these fields. If you have difficulty determining the cause of the error, you can look into the SNA logs. These are described in “SNA Traces and Logs” on page 26.

Common error causes:

- The destination address is wrong.
- The network ID or node IDs do not match the definitions on the host.
- The adjacent CP name is wrong. This must be the CP name of the communications controller (the gateway to the SNA network), not the CP of the CICS.
- The network adapter was not successfully installed.

Step 2: Configuring the LU 6.2 Session

Necessary parameters:

- Host Network ID (a5)
- CICS LU Name (a7)
- PC LU Name (a8)
- SNA Mode Name (a12)

An active link (as described in the previous section) is also a prerequisite.

Follow the steps described in:

- “Local LU 6.2 Configuration” on page 32
- “Partner LU 6.2 Configuration” on page 33
- “LU 6.2 Mode Configuration” on page 34

There is still no need to configure MERVA Link. Everything is done in the SNA Node Configuration program of either Communications Server or Personal Communications.

To check this configuration step, do the following:

1. Save your configuration and close the 'SNA Node Configuration' program.
2. In the SNA Node Operations program, go to the list of Local LUs 6.2. Select your Local LU. In the menu, select **Operations > CNOS Initialize...** (CNOS = change number of sessions).
3. In the following panel, select your Local LU name, type in the full (that means with preceding network ID) host LU 6.2 name, and change the mode name to the one you have defined. Press the OK button.
4. In SNA Node Operations, change to the LU 6.2 Sessions list. You should see two or more session entries in the list.

If you don't see any sessions entries in the LU 6.2 Sessions list, generate SNA traces. These are described in "SNA Traces and Logs" on page 26.

Common error causes:

- The LU names do not match the definitions on the host.
- The workstation LU must be independent. Check the definitions on the workstation and the host.
- If the LU 6.2 definitions need the workstation LU name to be identical to the CP name, don't define any local LU on the workstation. Personal Communications will then take the CP name as the default local LU name.

Step 3: Configuring Outbound MERVA Link

Necessary parameters:

- Symbolic destination name (a13)
- Partner TP name (host receiving transaction ID) (c3)
- Host MERVA Link node name (b1)
- Host ASP name (b2)
- Host MTP name (b4)
- Workstation MERVA Link node name (b24)
- Workstation ASP name (b3)
- Workstation MTP name (b6)
- Workstation ASP queues (b9, b10)
- Host ASP security options, for example encryption or authentication (b11)

If you are using security exits, they must be also defined. If you are using conversation security, you also need a user ID that has a password.

It is assumed that you have already done the two steps described above, and that a LU 6.2 session is now running.

First, update your SNA configuration in the SNA Node Configuration program. Refer to "CPI-C Side Information Configuration" on page 35 for information on how to setup the symbolic destination. Don't forget to save your configuration afterwards.

Next, define your MERVA Link local node name, the MERVA Link partner nodes, and an ASP for the outbound process. For more information on how to do this, refer to “Customizing the MERVA Link of MERVA USE & Branch” on page 38, and to the *MERVA USE & Branch for Windows NT Installation and Customization Guide*.

To check your configuration, do the following:

1. Start the new SNA configuration in the SNA Node Operations program. The link should be active, and an LU 6.2 session should not be opened so far.
2. Start MERVA in multi-user mode. Go to the MERVA Link Operating panel.
3. Select the ASP you created. In the menu, select **Selected > Kickoff**.
4. After a few seconds, the MERVA Link Operating panel will display one of the following:
 - QEMPTY, which means the sending queue is empty
 - CNFRMD, which means that a message was already sent and has been confirmed
 - An error code

If the MERVA Link Operating panel displays QEMPTY or CNFRMD, the configuration was a success. In this case, you will also see some opened sessions in the SNA Node Operations program.

If you get an error code, double-click on the ASP in the list. This opens a window containing a button labeled with the error code. Click this button for an explanation of the error.

In addition, you can also switch the outbound traces on. How to do this is described in “MERVA Link Inbound and Outbound Traces” on page 27.

The trace files will be written in the defined directory whenever you select **Kickoff** or **Start ASP**.

Common error causes:

- Node names, ASP names, or MTP names do not match the definitions on the host.
- Symbolic destination in MERVA Link customizing do not match the CPI-C side information in the SNA Node Configuration.
- An incorrect TP name was provided during MERVA Link customization or SNA node configuration. If a TP name was specified for both, the one specified during MERVA Link customization has precedence. This TP name must match the receiving TP name on the host.
- The ASP was not defined with **Line EBCDIC**.

Step 4: Configuring Inbound MERVA Link

Necessary parameters:

- PC receiving TP name (a10)
- PC receiving TP file name (a11)
- Conversation security definitions

To allow inbound MERVA Link connections, you have to define a transaction program (TP) during SNA node configuration, which is described in “Transaction Program Configuration” on page 36. To define the conversation security for this TP, see “LU 6.2 Security Profiles” on page 37.

To check the new configuration, do the following:

1. Save your new SNA configuration and restart SNA in the SNA Node Operations program.
2. On the partner node (the host node), start the corresponding ASP to establish a MERVA Link connection.
3. If all goes well, the outbound status of the ASP on the host should change either to QEMPTY or CNFRMD.

If the test fails, first check if there are any sessions established. You can check this in the LU 6.2 Session list in SNA Node Operations. If the sessions are established, check if the TP is actually started. If you get an inbound error report on the host, then the TP was already started and generated this error report. If you don't get an error report, you have two possibilities:

- Turn off the **Background Process** switch in the TP definition. A window will appear if the TP is started.
- Activate an inbound trace in MERVA Link Operating. If the trace is written, the TP was definitely started. This approach has the added benefit that the inbound trace might contain information about possible errors.

Common error causes:

- If the TP is not started, you have either defined a wrong path in the SNA TP definitions, or Windows NT is unable to execute the transaction program. The latter can be checked by starting **ekatpi.exe** directly from a MSDOS command shell.
- If the TP is started, it may be having trouble reaching the ACT in the shared memory region. Check that both MERVA and the MERVA Link Daemon are running.
- Check whether the databases are bound correctly, and if the instance name given as parameter to the TP is correct.

Generating Traces

You can activate traces on three different levels:

- SNA Traces and Logs (lowest level)
- MERVA Link inbound and outbound traces (one trace per connection)
- MERVA diagnose and programmers trace (highest level)

SNA Traces and Logs

The SNA Log is always written. You can access it through the SNA Node Operations program. In the menu select 'Launch' -> 'Log Viewer'. In the SNA Log you find error and success messages along with error codes and sense data. To view SNA sense data information you may use the 'Display SNA Sense Data' program which is installed with Personal Communications and Communications Server.

The SNA Trace delivers additional information about the API calls used for SNA. In most times you will generate this trace and sent it to an IBM representative. To generate an SNA trace, do the following:

1. Start the SNA Node Operations program.
2. In the menu select **Launch > Trace Facility**.
3. In the Trace window, select the following:

Function Name: APPN and APPC
Component Name: CPI-C API, APPC API
Trace Options: Verb Trace (up to 128 bytes of data)

4. Press the Start button (or, if the trace facility is already running, the Apply button).
5. Run the program with the error you want to trace.
6. Go back to the Trace window and push the **Save** button. Specify a destination directory and a name for the trace file, or use the default.
7. Push the **Format** button. The trace file, which has the extension *.tlg, is written to the directory specified in the previous step.

You can clear all trace settings by pushing the **Clear** button.

MERVA Link Inbound and Outbound Traces

MERVA Link inbound and outbound traces are written in several files: one inbound and outbound trace file for each connection attempt. The traces contain information about MERVA Link actions and the PDUs sent and received (trace level 3).

To activate the MERVA Link traces:

1. Start MERVA Link Operating.
2. In the menu **ASP_List > Set inbound trace**, you can set the trace levels for TCP/IP and SNA separately. Set the APPC inbound trace to **3** and press **Save**.
3. In the menu **ASP_List > Set trace directory**, provide an existing trace path for all MERVA Link trace files.
4. Select the ASP in MERVA Link Operating.
5. Open in the menu **Selected > Set outbound trace**. Set the trace level to **3** and press **Save**.

These settings will be saved and will still be active the next time you start MERVA Link.

Outbound trace file names have the format *ASPname.timestamp*; inbound trace file names have the format *PartnerNodeName.n*, where *n* is a number from 0 to 9). Outbound traces are generated whenever either an ASP is started or a Kickoff is made on the NT side. Inbound traces are generated whenever an incoming connection starts the transaction program.

MERVA Diagnosis Log and Programmer's Trace

The MERVA diagnosis log and programmer's trace contain information, success messages, and error reports from MERVA and the MERVA Link interfaces to MERVA. The diagnosis log contains information for the customer, while the programmer's trace is written for an IBM representative who might need to solve more serious problems.

Set the tracing level in the MERVA main menu under **Setup > Logging Level**. For most error situations, a logging level of 3 is sufficient.

All MERVA Link diagnosis log entries are written in the `\Logs\Base\` directory of your MERVA instance. The MERVA Link programmers trace entries can be found in `\Traces\Base\` and `\Traces\MRVLink\` directories of the MERVA instance.

Customizing the Windows NT Based Components

This section describes how to set up the workstation to run the MERVA USE & Branch, including how to define an Advanced Program-to-Program Communications (APPC) Logical Unit 6.2 (LU 6.2) connection to a host system.

Because a description of the security facilities of APPC/MVS, CICS, and Windows NT is beyond the scope of this book, only information describing which security levels are supported and how to manage the workstation side is supplied.

Configuring Communications Server for Windows NT

A sample set of Windows NT SNA definitions is shown in the following. The samples apply to Version 5.0.1 of the IBM Communications Server for Windows NT. For more detailed information, refer to the Communications Server documentation.

The following steps are necessary to define the APPC connection:

- Node configuration
- Device configuration
- Connection configuration
- LU 6.2 session configuration
- CPI-C side information configuration
- Transaction program configuration

Note: For all parameters that are not explicitly mentioned here, use the default value of the parameter.

Node Configuration

In this part of the configuration, you work with the following parameters:

- Network ID (**a1**) (also known as the network name)
- Local node name (**a2**) (also known as the control point name)
- Local node ID (**a3**)

These parameters are described in Table 1 on page 11.

Do the following:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **New** and **CPI-C, APPC, or 5250 Emulation**.

The Define the Node window is displayed.

The following figure shows an example of this window:

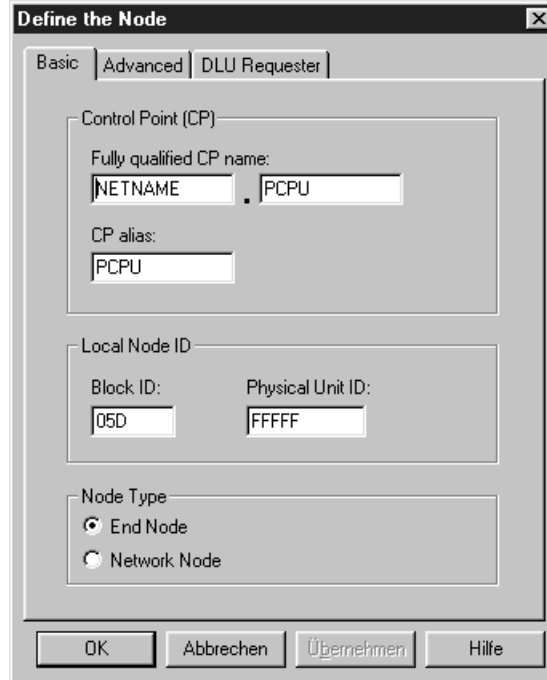


Figure 5. The Define the Node Window

In the Define the Node window:

- The fully-qualified unique **Control Point (CP)** name identifies the node within the network environment. The first part is the name of the network (a1), and the second part is the CP name (a2).
- The **Local Node ID** is the concatenation of the values specified for the IDBLK and IDNUM parameters (a3) in the VTAM PU definition.
- The **Node Type** must be the **end_node** because APPN services are not used in this example.
- The defaults are used for the remaining parameters in the Define the Node window.

Device Configuration

To define a supported communication device:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **LAN** as **DLC** type, then select **New**.
The Define a LAN Device window is displayed.

Figure 6 on page 30 shows an example of this window:



Figure 6. The Define a LAN Device Window

In the **Adapter number** field, specify the adapter number. If an adapter number is shown by default, the IBM LLC2 LDC interface was not installed during the Communications Server installation.

Connection Configuration

In this part of the configuration, you work with the following parameters:

- Local node ID (**a3**)
- LAN destination address (**a4**) (token ring only)

These parameters are described in Table 1 on page 11.

To define links to other nodes in the SNA network:

1. Click **Start -> Programs -> IBM Communications Server -> Configure Connections**.
2. Select **LAN** as **DLC** type, then select **New**.
The Define a LAN Device window is displayed.

In the **Destination address** field, specify the address of the remote node to which the connection is to be established (**a4**), as shown in the following figure.

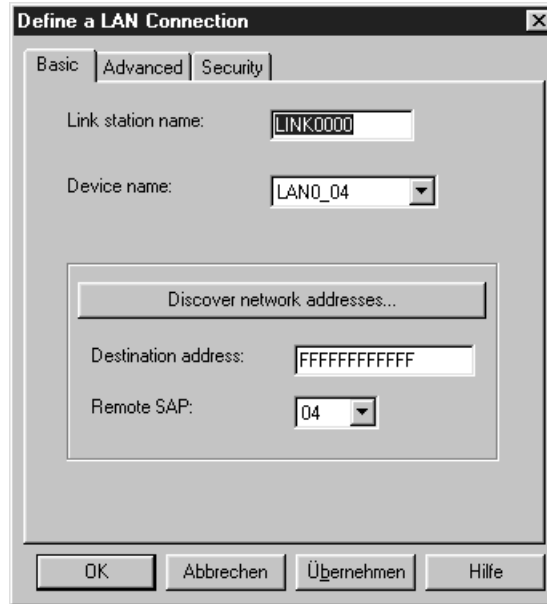


Figure 7. The Define a LAN Connection Window—Basic Page

To activate the link:

1. Select the **Advanced** page.
2. Select **Activate link at start**, as shown in the following figure.

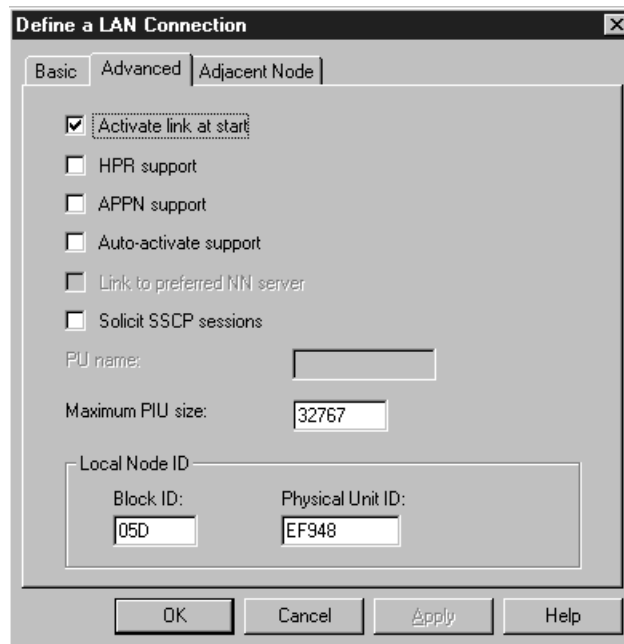


Figure 8. The Define a LAN Connection Window—Advanced Page

Select the **Adjacent Node** page and fill in the fields as shown in the following figure.

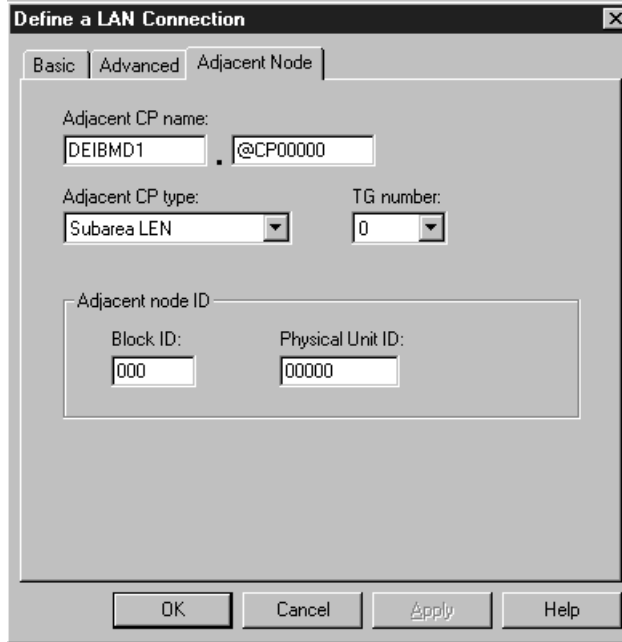


Figure 9. The Define a LAN Connection Window—Adjacent Node Page

LU 6.2 Session Configuration

The setup of LU 6.2 sessions is configured in a Local LU profile, in Side Information profiles, in Partner LU profiles, in Mode profiles, in a Transaction Program Name profile, and in a Partner LU 6.2 Location profile. Each profile must be contained in the sample configuration. You can also use Conversation Security profiles (Conversation Security Access List and Resource Security Access List).

The Session Security, Session Timeout, and the Class of Service (COS) profiles provided by the Communications Server are not modified for MERVA Link purposes.

Local LU 6.2 Configuration: In this part of the configuration, you work with the following parameter:

- Local LU name (**a8** and **d2**)

This parameter is described in Table 1 on page 11.

To define a local LU:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **Configure Local LU 6.2**, then select **New**.
The Define a Local LU 6.2 window is displayed.

The following figure shows an example of this window:



Figure 10. The Define a Local LU 6.2 Window

In the Define a Local LU 6.2 window:

- The **Local LU name** is the name of the APPC LU as defined in the label of the LU macro that describes the LU in VTAM (a8 and d2).
- Define the local LU as an independent LU. To do this, select **No** in the field **Local LU is dependent?**
- You can choose any name for the **Local LU alias**. The default LU alias is the LU name.

Partner LU 6.2 Configuration: The name of the partner LU is the name of the LU in which the partner program is located. The local LU recognizes the name of the remote LU to allocate a conversation.

In this part of the configuration, you work with the following parameters:

- Partner network ID (**a5**)
- Partner node name (**a6**)
- Partner LU name (**a7** and **d3**)

These parameters are described in Table 1 on page 11 and Table 4 on page 14.

To define a partner LU:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **New** and **Configure Partner LU 6.2**.

The Define a Partner LU 6.2 window is displayed.

The following figure shows an example of this window:

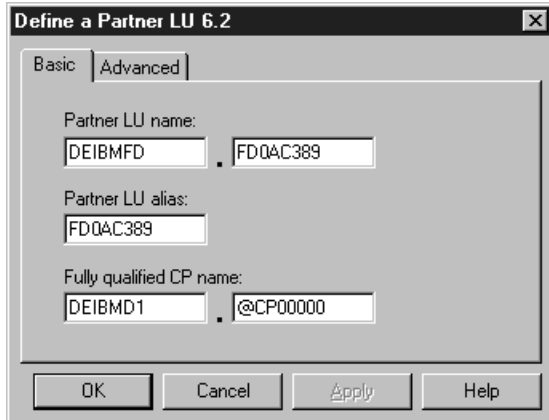


Figure 11. The Define a Partner LU 6.2 Window

In the Define a Partner LU 6.2 window:

- Specify the fully-qualified partner LU name to which you want to connect. The fully-qualified name is the network name (a5) and the LU name (a7 and d3).
- Specify the fully-qualified adjacent control point name (a5.a6).
- You can choose any name for the partner LU alias. The default partner LU alias is the partner LU name.

LU 6.2 Mode Configuration: In this part of the configuration, you work with the following parameters:

- Mode name (a12 and d4)

To define an LU 6.2 mode:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration.**
2. Select **New** and **Configure Modes.**
The Define a Mode window is displayed.

The following figure shows an example of this window:

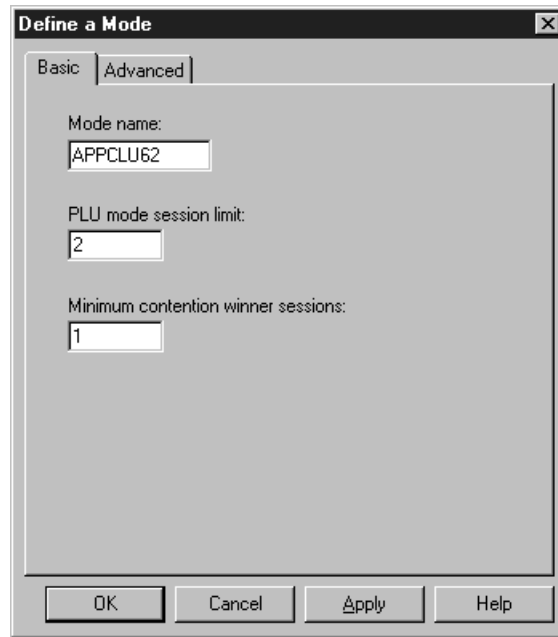


Figure 12. The Define a Mode Window

Two sessions are sufficient for MERVA Link operations. Each system, the local system and the partner system, gets a session of its own as a contention-winner session. Each system can start a conversation without asking for permission from the partner system.

MERVA Link does not need a specific RU size. You can specify any valid number. Note that this number affects the resources of the local and the partner system. The sample maximum RU size of 16384 ensures the highest possible message transfer rate. Depending on your system resources, you might specify a lower number.

The RU size that actually applies to an LU 6.2 session is negotiated between the partner systems at the time when the session is bound. It depends on the RU sizes that are specified in the log mode table of the partner SNA node. It also depends on the mechanism that binds the LU 6.2 session.

CPI-C Side Information Configuration: You must define the receiving processes in the partner LUs of MERVA Link in CPI-C Side Information. The definition of the side information is called the *symbolic destination name*.

In this part of the configuration, you work with the following parameters:

- Partner LU alias (**a7** and **d3**)
- Mode name (**a12** and **d4**)
- TP name (**c3**)

These parameters are described in Table 1 on page 11, Table 3 on page 13, and Table 4 on page 14.

To define CPI-C side information:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **New** and **Configure CPI-C Side Information**.

The Define CPI-C Side Information window is displayed.

The following figure shows an example of this window:

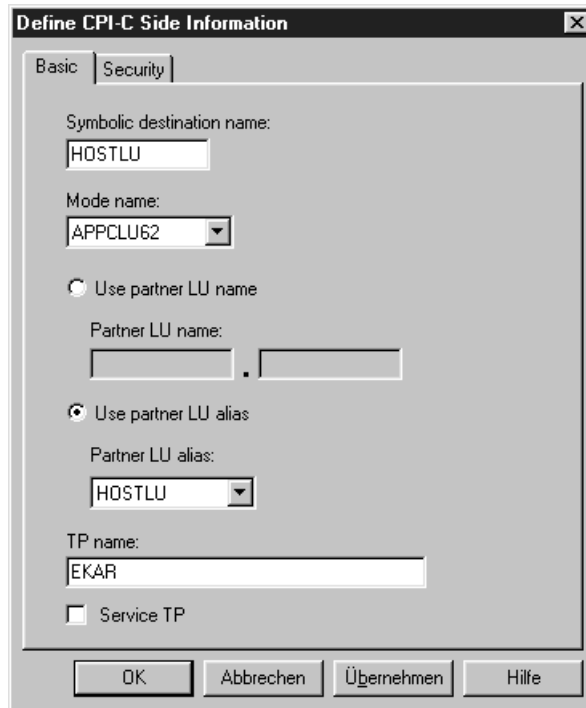


Figure 13. The Define CPI-C Side Information Window

The sample Side Information profile shown in Figure 13 defines a symbolic destination for the partner MERVA Link system. The sample symbolic destination name is the same as the partner LU alias name (a7).

The TP name is the transaction identifier of the MERVA Link ESA receiving transaction (c3 or d8). The sample MERVA Link transaction identifiers are EKAR and EKARI4.

Transaction Program Configuration: In this part of the configuration, you work with the following parameter:

- TP name (a9)

To define an LU 6.2 transaction program:

1. Click **Start -> Programs -> IBM Communications Server -> SNA Node Configuration**.
2. Select **New** and **Configure Transaction Programs**.

The Define a Transaction Program window is displayed.

The following figure shows an example of this window:

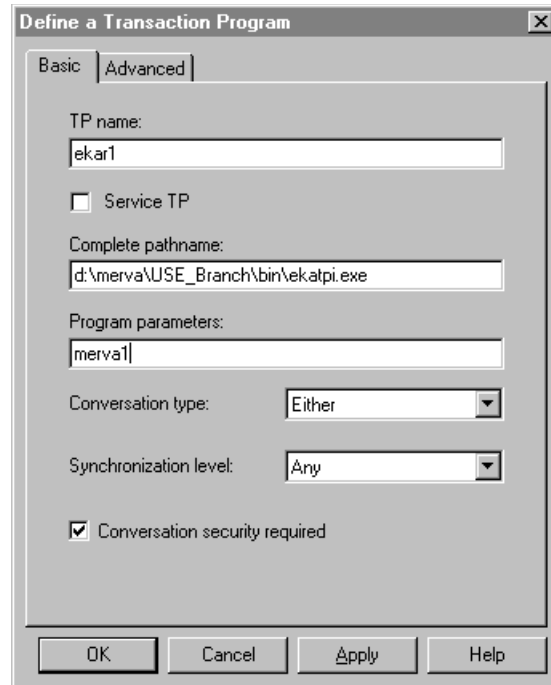


Figure 14. The Define a Transaction Program Window

The sample LU 6.2 definition defines the MERVA Link receiving TP that is associated with the MERVA instance **merva1**. For each MERVA instance in a MERVA host, you must generate a unique TP name profile for the MERVA Link receiving TP. Only the TP name and the program parameter name must be different because the MERVA instance is specified as a program parameter.

In the field **Complete pathname**, specify the Windows NT path of the MERVA Link receiving program **ekatpi.exe**. The program is located in the subdirectory **bin** of the MERVA installation directory, for example, **D:\merva\USE_Branch\bin**.

Note the MERVA Link receiving TP supports only mapped conversations for which the synchronization level is set to **Confirm** or **Any**.

By default, the flag **Conversation security required** is selected. The following prerequisites apply:

- To start the TP, conversation security information is required. Incoming allocation requests for this TP without this information are rejected.
- The supplied user must be defined in the Windows NT system and in the LU 6.2 conversation security. To define user IDs and passwords, use **Configure User ID Passwords**.

In the Advanced window, ensure that the fields **Dynamically loaded**, **Full duplex support**, and **Background process** are checked.

LU 6.2 Security Profiles: A MERVA Link inbound conversation and a MERVA Link receiving TP can be protected by the Communications Server conversation security function. An inbound conversation is protected by selecting **Conversation security required** in the Define a Program Transaction window. Select this option to check the conversation security when you create a transaction program definition.

To define conversation security for a TP, do the following in the SNA Node Configuration program:

On the server side:

1. Select **Conversation security required** in the Define a Transaction Program window.
2. In the Configure User ID Passwords window, type the user ID and the corresponding password for each user.

On the client side:

1. In the Partner LU 6.2 window, select **Advanced -> Conversation security support**.
2. Provide the user ID and the password in **Security** of the CPI-C side information or in the MERVA Link Partner Node Customization.
To change the user ID and password during a session, select **ASP -> Detailed Information -> ISC Info** in the MERVA Link Operating window.

Automatically Starting Communications Server Resources

You can specify that the SNA Communications Server starts automatically when the Windows NT system is started. Use the **CSSTART** command to start the Communications Server with a specific configuration. This command is part of the Command Line Utilities for command line programs.

For more information about the parameters of the command line programs, refer to the corresponding documentation.

Configuring Queues and Routing Conditions for USE Functions

With MERVA USE & Branch, you receive sample routing that provides complete USE functionality (see the routing appendix of the *MERVA USE & Branch for Windows NT Installation and Customization Guide*).

Use the Import utility to import the supplied standard routing (see “Installing MERVA” in the *MERVA USE & Branch for Windows NT Installation and Customization Guide*).

Customizing the MERVA Link of MERVA USE & Branch

In this part of the configuration, you work with the following parameters:

- Local Node Name (**b24**)
- Partner Node Name (**a7** and **d3**)
- Symbolic Destination (**a7**)
- Transaction Program Name (**c3** and **d8**)
- Local ASP Name (**b3**)
- Local MTP Name (**b6**)
- Partner ASP Name (**b2**)
- Partner MTP Name (**b5**).

These parameters are described in Table 1 on page 11, Table 2 on page 12, Table 3 on page 13, and Table 4 on page 14.

All the parameters of MERVA Link of MERVA USE & Branch must be specified using either the Customization program or the Import utility.

To customize MERVA Link:

1. Specify your **Local Node Name**.

To specify the local node name, select *Component data* and *MERVA Link* from the Customization program. The MERVA Link entry window is shown. Enter the **Local Node Name (b24)**.

2. Define the **Partner Node Name**.

To define the partner node details, select *Partner Nodes* from the *MERVA Link* menu item. The *Partner Nodes* window is shown. To define partner node details, click *New*, or select a partner node and click *Edit*. The Intersystem Communication (ISC) Information window is shown.

Enter the Symbolic Destination (**a7**) and the Transaction Program Name (**c3** and **d8**). The Symbolic Destination is the name of the Side Information Profile defined in the Communications Server for an SNA APPC connection to the partner system. The Side Information Profile is the reference to the partner LU Name (**a7** and **d3**).

3. Depending on whether MERVA ESA is running under CICS or under IMS, the Transaction Program Name must correspond to the:

- CICS transaction definition
- APPC/MVS TP profile name (for IMS/VS V2 or IMS/ESA V3)
- APPC/IMS TP profile name (for APPC/IMS of IMS/ESA V4 or higher)

4. The user ID must be an existing user ID on MERVA ESA.

5. Define **ASP Information**.

To define your ASP details, click *New* on the MERVA Link window, or select an ASP name in the MERVA Link window and click *Edit*. The ASP Information window is shown.

Enter the ASP Name (**b3**), the partner ASP Name (**b2**), the MTP Name (**b6**), and the partner Node Name (**b5**). The Transfer Format must be Line EBCDIC. The Preferred Connection Type must be APPC. See the *MERVA USE & Branch for Windows NT Installation and Customization Guide* for a description of the individual fields.

MERVA Link Conversation Security

Cooperative processing allows application programs to establish communications with partner programs on other systems, and to share work, data, and services between systems and across networks. This ability to access other programs and all the resources at their disposal poses special security considerations for installations that use cooperative processing.

In a client-server environment it is the server that sets the conversation security requirements for a conversation with a specific client (or for all clients). A client must be aware of the server's security requirements, and must provide the applicable access security information.

MERVA Link can be considered as a client when it allocates a conversation to a partner system to send messages to a partner application, the server. MERVA Link can be considered as a server when it accepts an inbound conversation from a partner system to receive messages, and to pass the messages to a receiving application.

Conversation security terms are explained in the following section. Conversation security is discussed separately for MERVA Link for Windows NT in the follow-on sections.

Conversation Security Terms

A number of terms are used when discussing conversation security.

APPC Conversation Security Information

APPC conversation security information is a Conversation Security User ID (CSU) and an optional Conversation Security User Password (CSP). The CSU identifies a user in a server system. The CSU and the optional CSP must be provided by a client application when it requests the services of a server application.

User profile management facilities apply in the Windows NT environment. Communications Server for Windows NT makes use of these facilities for APPC conversation security purposes.

Outbound Conversation Security Types

The security type of an outbound APPC allocate request can be NONE, SAME, or PROGRAM.

Security type **NONE** means that no conversation security information is provided for an outbound conversation. Depending on the security mechanisms implemented in the partner system, a conversation with the partner process can be established, or the allocate request is rejected because of insufficient access authority.

Security type **SAME** means that a conversation security user ID is provided for an outbound conversation. Depending on the requirements of the partner system, the requested conversation security level **already_verified** can be downgraded to **none**.

Security type **PROGRAM** means that a conversation security user ID and a conversation security user password is provided for an outbound conversation. Depending on the requirements of the partner system, the requested conversation security level **conversation** can be downgraded to **none**.

Inbound Conversation Security Levels

An inbound conversation is accepted and the server TP is started if the conversation security level meets the requirements of the server.

Security level **none** means that no conversation security information is available from the inbound conversation. The server TP can access only unprotected resources.

Security level **already_verified** means that a user ID is available from the inbound conversation, and the user ID has been verified by the client system.

Security level **conversation** means that a user ID and password are available from the inbound conversation. The user ID and password are verified in the server system.

Conversation Security for MERVA Link NT

The following considerations apply in the MERVA USE & Branch environment.

MERVA Link NT Outbound Conversation Security

A MERVA Link sending process (client) must match the conversation security settings that have been set up by the partner system (server). The security type of a MERVA Link outbound allocate request can be NONE, SAME, or PROGRAM. MERVA Link chooses the security type based on the security information provided

in MERVA Link customization. The security information can be a user ID, a password, or both. Conversation security settings defined in the Communications Server configuration are overwritten by the settings defined in the MERVA Link customization.

Security Type NONE: If a user ID is not part of the ISC parameters associated with a specific MERVA Link partner node, MERVA Link allocates a conversation to that partner with security type NONE. The partner system rejects the allocation request if a security type other than NONE is defined on the partner inbound system.

Security Type SAME: If a user ID is part of the ISC parameters associated with a specific MERVA Link partner node but a password is not specified, MERVA Link allocates a conversation to that partner with security type SAME. The requested security type can be downgraded to NONE if the partner system inbound security requirements are also set to NONE.

If the requested conversation security type (SAME) is accepted by the partner system, the conversation access security information in the SNA Attach Header (FMH 5) that is sent to the partner system will contain the user ID and the AV indicator (user ID already verified). The user ID is checked in the partner system. A conversation with the partner process is established only if the user is defined in the receiving system.

Any partner system of MERVA Link NT is advised not to accept conversation security type SAME from the Communications Server LU. A Windows NT system is not considered as a trusted system because a single user can have the full control of all user and security administrator functions.

Security Type PROGRAM: If a user ID and a password are part of the ISC parameters associated with a specific MERVA Link partner node, MERVA Link allocates a conversation to that partner with security type PROGRAM. Depending on the security requirements of the partner inbound system, the requested security type (PROGRAM) can be downgraded to NONE.

The user ID and the password are checked on the partner system. A conversation with the partner process is established only if the access security information is acceptable.

MERVA Link NT Inbound Conversation Security

The MERVA Link NT Inbound Conversation Security is specified and handled by the Communications Server outside the scope of MERVA Link. When the MERVA Link SNA APPC TP executes, all security checks have been successfully passed.

The Communications Server checks the user ID and password—as far as applicable—before an inbound transaction program is started. User IDs and passwords in the NT environment are case-insensitive and written in mixed case while user IDs and passwords in the MVS™ environment are usually all uppercase characters. This is why Communications Server not only checks the user ID and password as it is received, but also checks for a match of the user ID and password as if they had all lowercase characters.

The required conversation security level for an inbound conversation is specified in the Communications Server's Configuring Partner LU 6.2 and in the Configuring

Transaction Programs (TP) windows within the SNA node configuration program. The related security information can be defined in the Communications Server User IDs and Passwords configuration.

Configure Transaction Programs: Activate the **Conversation security required** checkbox in the transaction program definition to set the security requirement to either PROGRAM or SAME. The distinction between the levels SAME and PROGRAM is made in the Partner LU 6.2 configuration. If the security requirement is set to SAME, the user ID and the AV indicator is expected from the partner system. The client's user ID must be defined as an NT user in the Windows NT User Management. If the security requirement is set to PROGRAM, the user ID and the password must be provided by the partner system and the user ID and the password must be defined in the Communications Server user ID and password list.

If the security requirements are not met by the client system, the conversation security is automatically downgraded to NONE and the conversation is rejected.

Configure Partner LU 6.2: Activate the checkbox "Conversation security support" in the "Define a Partner LU 6.2" advanced panel of the SNA node configuration program to indicate that this partner LU is a trusted system. A trusted system can send "Already Verified" (AV) indicators during the conversation security check. This means that this partner LU is allowed to use the conversation security level SAME and the server side won't reject any conversation allocation attempts with this security level.

Configure User IDs and Passwords: Use this panel in the SNA node configuration to define a list of user IDs and passwords to check the provided user IDs and passwords if conversation security level PROGRAM is used.

How to Recognize a Security Violation

A conversation security violation can be reported in MERVA Link NT as a resource failure (access denied) or as an APPC security failure.

Resource Failure: A mismatch between the security information provided by the client and the security requirements of the server can result in a downgrade of the conversation security level to **NONE**. As a consequence, the server can access only unprotected resources. If it requires access to a protected resource, an APPC resource failure is indicated by the SNA APPC service.

An APPC resource failure is reported by MERVA Link NT as a sending ASP error with diagnostic code **TS2E1A**. Hexadecimal 1A (26) is the APPC service return code for this error (CM_RESOURCE_FAILURE_NO_RETRY).

Security Not Valid: The conversation security information provided by the client is verified in the server system. If this verification fails (user ID or password incorrect), an APPC security violation is indicated by the SNA APPC service.

An APPC security violation is reported by MERVA Link NT as a sending ASP error with diagnostic code **TS2E06**. Hexadecimal 06 (6) is the APPC service return code for this error (CM_SECURITY_NOT_VALID).

Parameter Relationships

The following parameter relationship tables show where parameters occur across all the involved systems: VTAM, CICS, IMS, APPC/MVS, MERVA ESA, MERVA USE & Branch, and Communications Server for Windows NT. They describe in which subsystem each parameter is defined, where within the subsystem each parameter must be specified, and the default values that are supplied with the sample configuration. In this way, the tables provide you with an overview of the entire system. For example, if you have completed the initial configuration and the connection does not work, you can use the tables to quickly find and investigate the value of a particular parameter that you suspect is causing the problem.

The tables have the following columns:

Name	The parameter name, for example, a1 . Each parameter is described in "Parameter Definitions" on page 10.
Host (MVS)	The host application in which the parameter occurs, for example, CICS.
Workstation	The workstation fields in which the parameter must be adapted.
Value in Sample	The initial value set in the sample code.
Your value	If you change a value for your system, write the new value in this column. Do not change the value of a parameter that has a value already printed in this column.

Parameters for a MERVA Link Connection—CICS to Workstation

Table 5 shows the necessary parameters for the connection from MERVA ESA under CICS to MERVA USE & Branch.

Table 5. Parameters for a MERVA Link Connection—CICS to Workstation

Name	Host (MVS)	Workstation	Value in Sample	Your Value
a7	<p>VTAM</p> <p>In the VTAM definition:</p> <pre> VBUILD TYPE=APPL a7 APPL : : ,ACBNAME=a7 </pre> <p>CICS</p> <p>One of the following:</p> <ul style="list-style-type: none"> In the system initialization table DFHSITxx: <pre> DFHSIT ... ,APPLID=a7 </pre> In the CICS startup job: <pre> //SYSIN DD * : : ,APPLID=a7 </pre> 	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> Window: Partner LU <ul style="list-style-type: none"> Field: Partner LU name: a5.a7 Field: Partner LU alias: a7 Window: Side Information <ul style="list-style-type: none"> Field: Partner LU alias: a7 	HOSTLU	
a8	<p>VTAM</p> <p>In the VTAM definition:</p> <pre> a8 LU LOCADDR=0, : : </pre> <p>CICS</p> <p>One of the following:</p> <ul style="list-style-type: none"> In the terminal definition DFHTCTxx: <pre> DFHTCT TYPE=SYSTEM, NETNAME=a8, </pre> In the offline utility: <pre> //SYSIN DD * DEFINE CONNECTION(c1) NETNAME(a8) : : MERVA ESA MERVA Link Partner Table EKAPT TYPE=ASP, : : ,DEST=(a8,...) </pre> 	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> Window: LU 6.2 Local LU <ul style="list-style-type: none"> Field: Local LU name a8 Field: Local LU alias a8 	PS2LU00	

Table 5. Parameters for a MERVA Link Connection—CICS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
a12	<p><u>VTAM</u></p> <p>In the VTAM definition:</p> <pre> VBUILD TYPE=APPL a7 APPL DLOGMOD=a12, MODETAB=PPC3270, : : PPC3270 MODETAB a12 MODEENT LOGMODE=a12, : : a8 LU LOCADDR=0, DLOGMOD=a12, : : </pre> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> In the terminal definition DFHTCTxx: DFHTCT TYPE=MODESET, MODENAM=a12, : : In the offline utility: //SYSIN DD * DEFINE SESSIONS(c1AAC) MODENAME(a12) : : 	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> Window: Mode <ul style="list-style-type: none"> – Field: Mode name a12 Window: Side Information <ul style="list-style-type: none"> – Field: Mode name a12 	APPCLU62	

Table 5. Parameters for a MERVA Link Connection—CICS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
c1	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> In the terminal definition DFHTCTxx: <pre>DFHTCT TYPE=SYSTEM, : : ,TRMTYPE=LUTYPE62 ,SYSIDNT=c1 DFHTCT TYPE=MODESET, SYSIDNT=c1, :</pre> In the offline utility: <pre>//SYSIN DD * DEFINE CONNECTION(c1) :</pre> <p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: <pre>EKAPT TYPE=MTP, : ,LINK=(APPC,c1)</pre> </p>		PS22	
c2	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> In the system initialization table DFHSITxx: <pre>DFHSIT ... ,ISC=c2</pre> In the CICS startup job: <pre>//SYSIN DD * : : ,ISC=c2</pre> 		YES	YES
c3	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> In the transaction definition DFHPCTxx: <pre>DFHPCT TYPE=ENTRY, : : ,TRANSID=c3</pre> In the offline utility: <pre>//SYSIN DD * DEFINE TRANSACTION(c3) :</pre> 	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> Window: Side Information <ul style="list-style-type: none"> Field: TP name c3 <i>or</i> In the MERVA Link definitions that are part of the ISC customization program in MERVA Link NT: <ul style="list-style-type: none"> Transaction program name c3 	EKAR	

Table 5. Parameters for a MERVA Link Connection—CICS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
c4	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=c3, PROGRAM=c4, : : • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=c4, : : • In the offline utility: //SYSIN DD * DEFINE TRANSACTION(c3) PROGRAM(c4) : : DEFINE PROGRAM(c4) : : 		EKATR10	EKATR10
c5	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=c5, : : • In the offline utility: //SYSIN DD * DEFINE TRANSACTION(c5) : : <p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, TRAN=c5, : :</p> <p>In the function table DWSFNTPC: DSLFT NAME=b7, TRAN=c5, : :</p>		EKAS	

Table 5. Parameters for a MERVA Link Connection—CICS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
c6	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=c5, PROGRAM=c6, : : • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=c6, : : • In the offline utility: //SYSIN DD * DEFINE TRANSACTION(c5) PROGRAM(c6) : : DEFINE PROGRAM(c6) 		EKAAS10	EKAAS10

Parameters for a MERVA Link Connection—IMS to Workstation

Table 6 shows the necessary parameters for the connection from MERVA ESA under IMS to MERVA USE & Branch.

Table 6. Parameters for a MERVA Link Connection—IMS to Workstation

Name	Host (MVS)	Workstation	Value in Sample	Your Value
d1	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=MTP, ⋮</p> <p>MTPCI=(,d1), ⋮</p>		EKATPO1	EKATPO1
d2	<p><u>VTAM</u></p> <p>In the VTAM definition:</p> <p>d2 LU LOCADDR=0, ⋮</p> <p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=ASP, DEST=(d2,...), ⋮</p> <p>EKAPT TYPE=MTP, ⋮</p> <p>,LINK=(APPC,d2,d4)</p>	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> • Window: LU 6.2 Local LU <ul style="list-style-type: none"> – Field: Local LU name d2 – Field: Local LU alias d2 	PS2LU00	
d3	<p><u>VTAM</u></p> <p>In the VTAM definition:</p> <p>VBUILD TYPE=APPL d3 APPL ACBNAME=d3, ⋮</p> <p><u>APPC/MVS</u></p> <p>In APPCPMxx:</p> <p>LUADD ACBNAME(d3) ⋮</p> <p><u>MERVA ESA APPC/IMS</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=MTP, ⋮</p> <p>,LOCLU=d3</p>	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> • Window: Partner LU <ul style="list-style-type: none"> – Field: Partner LU name a5.d3 – Field: Partner LU alias d3 • Window: Side information <ul style="list-style-type: none"> – Field: Partner LU alias d3 	HOSTLU APPC/MVS and APPC/IMS	

Table 6. Parameters for a MERVA Link Connection—IMS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
d4	<p>VTAM</p> <p>In the VTAM definition:</p> <pre>d4 MODEENT LOGMODE=d4, : : d2 LU LOCADDR=0, DLOGMOD=d4, : :</pre> <p>MERVA ESA</p> <p>In the MERVA Link partner table:</p> <pre>EKAPT TYPE=MTP, : : LINK=(APPC,d2,d4), : :</pre>	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> • Window: Mode <ul style="list-style-type: none"> – Field: Mode name d4 • Window: Side information <ul style="list-style-type: none"> – Field: Mode name d4 	APPCLU62	
d5	<p>IMS</p> <p>In the transaction definition:</p> <pre>APPLCTN PSB=d6, : : TRANSACT CODE=d5,</pre> <p>MERVA ESA</p> <p>In the MERVA Link partner table:</p> <pre>EKAPT TYPE=ASP, TRAN=d5, : :</pre> <p>In the MERVA Link function table DWSFN TTC:</p> <pre>DSL FNT NAME=b7, : : ,TRAN=d5</pre>		EKAS	
d6	<p>IMS</p> <p>In the transaction definition:</p> <pre>APPLCTN PSB=d6, : :</pre>		EKAAS10	EKAAS10
d7	<p>VTAM</p> <p>In the VTAM definition:</p> <pre>VBUILD TYPE=APPL d3 APPL ACBNAME=d3, SECACPT=d7, : :</pre>		ALREADYV	

Table 6. Parameters for a MERVA Link Connection—IMS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
d8	<p><u>APPC/MVS TP Profile</u> (SYS1.APPCTP) ⋮ TPADD TPNAME(d8) ⋮ //EKA EXEC PGM=d9 ⋮ //EKASNAP DD DSN=...</p> <p><u>APPC/IMS TP Profile</u> (SYS1.APPCTP) ⋮ TPADD TPNAME(d8) ⋮ TRANCODE=d9 ⋮</p>	<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> Window: Side information <ul style="list-style-type: none"> – Field: TP name d8 <p><i>or</i></p> <p>In the MERVA Link definitions that are part of the ISC customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Transaction program name d8 	<p>EKAR (APPC/MVS) or EKARI4 (APPC/IMS)</p>	
d9	<p><u>APPC/MVS TP Profile</u> (SYS1.APPCTP) ⋮ //EKA EXEC PGM=d9 ⋮</p> <p><u>APPC/IMS TP Profile</u> (SYS1.APPCTP) ⋮ TPADD TRANCODE=d9 ⋮</p>		EKATPI1	EKATPI1

Parameters for a MERVA Link Connection Independent of CICS or IMS

Table 7 shows the parameters for the connection from MERVA ESA to MERVA USE & Branch. These parameters are independent of the host subsystem used (CICS or IMS).

Table 7. Parameters for a MERVA Link Connection Independent of CICS or IMS

Name	Host (MVS)	Workstation	Value in Sample	Your Value
a1		Product: Communications Server for Windows NT • Window: Node – Field: Fully-qualified CP name a1.a2	NETNAME	
a2	<u>VTAM</u> In the VTAM definition: a2 PU ..., ⋮	Product: Communications Server for Windows NT • Window: Node – Field: Fully-qualified CP name a1.a2	PCPU	
a3	<u>VTAM</u> In the VTAM definition for Windows NT: a2 PU IDBLK= 05D, IDNUM= a3 , ⋮	Product: Communications Server for Windows NT • Window: Node – Field: Physical Unit ID a3 – Field: Block ID=VTAM parameter IDBLK	FFFFF	
a4	<u>Control unit 37x5:</u> In the Network Control Program (NCP): ... LINE ..., LOCADD = a4 , ⋮	Product: Communications Server for Windows NT • Window: Connections – Field: Destination address a4	FFFFFF	
a5	<u>VTAM</u> In the VTAM definition: Startoption Netid = a5 ⋮	Product: Communications Server for Windows NT • Window: Partner LU – Field: Partner LU name a5.a7	NETNAME	
a9	<u>MERVA ESA</u> In the MERVA Link partner table: EKAPT TYPE=MTP, ⋮ ,PARTNER=(..., a9)	Product: Communications Server for Windows NT • Window: Transaction Programs – Field: TP name a9	EKAR1	

Table 7. Parameters for a MERVA Link Connection Independent of CICS or IMS (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
a10		<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> • Window: Transaction Programs <ul style="list-style-type: none"> – Field: Fully-qualified path name a10 <p>The fully-qualified path name is <i>path</i>\USE_Branch\bin\ekatpi.exe</p> <p>where <i>path</i> is the path to the directory where MERVA USE & Branch is installed.</p>		
a13		<p>Product: Communications Server for Windows NT</p> <ul style="list-style-type: none"> • Window: Side Information <ul style="list-style-type: none"> – Field: Symbolic destination name a13 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ISC customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Symbolic destination a13 	HOSTLU	
b1	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <pre>EKAPT TYPE=INITIAL, : : , NODE=b1</pre>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Partner MTN name b1 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Partner node b1 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ISC customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Partner node b1 	MESAMTN	
b2	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <pre>EKAPT TYPE=ASP, : : , NAME=b2 EKAPT TYPE=MTP, : : , ASP=b2</pre>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Partner ASP name b2 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Partner ASP name b2 	MESAASP	

Table 7. Parameters for a MERVA Link Connection Independent of CICS or IMS (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b3	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=ASP, ⋮ ,DEST=(... ,b3)</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Local ASP name b3 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Local ASP name b3 	M2ASPA	
b4	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=ASP, ⋮ ,MTP=b4 EKAPT TYPE=MTP, ⋮ ,NAME=(b4,...)</p>		MTP	
b5	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=MTP, ⋮ ,NAME=(... ,b5)</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Partner MTP name b5 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Partner MTP name b5 	MESAMTP	
b6	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=MTP, ⋮ ,PARTNER=(b6,...)</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Local MTP name b6 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> Local MTP name b6 	M2MTPA	

Table 7. Parameters for a MERVA Link Connection Independent of CICS or IMS (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b8	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, CONTROL=b8, : :</p> <p>In the function table DWSFNTTC: DSLFNT NAME=b8, ROUTE=b18, : :</p>		TX2USECQ	
b11	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, : : ,SECURE=b11</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Select encryption • Select authentication <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Select encryption • Select authentication 	ENCRYPT	
b15	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, MFSEXIT=b15, : :</p> <p>In the MFS program table entry: DSLMPT NAME=EKAMU133, NUMBER=b15, : :</p>		7133	7133
b16	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP,FORMAT=b16, : :</p>		(MCB, TCOV, #)	

Table 7. Parameters for a MERVA Link Connection Independent of CICS or IMS (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b22	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=ASP, ⋮ JOURNAL=b22, ⋮</p>		(OM,IR,IM)	
b24	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table:</p> <p>EKAPT TYPE=ASP, ⋮ ,DEST=(b24,...)</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Local node name b24 	MAIXMTN	

Common Parameter for Telex and USE

Table 8 contains an additional customizable value common for the MERVA ESA USE feature.

Table 8. Common Parameter for Telex and USE

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b18	<p><u>MERVA ESA</u></p> <p>In the function table DWSFNTTC:</p> <pre> DSLNT NAME=b18, ROUTE=b18 : : </pre> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> In the program definition DFHPPTxx: <pre> DFHPPT TYPE=ENTRY, PROGRAM=b18 : : </pre> In the offline utility: <pre> //SYSIN DD * DEFINE PROGRAM(b18) : : </pre> 		EKARTTXU	

Additional Parameters for USE

Table 9 contains additional customizable values for the workstation based USE functions.

Table 9. Additional Parameters for USE

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b7	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, SENDQC=(b7,...,...), ⋮</p> <p>In the function table DWSFNTPC: DSLFNT NAME=b7, ⋮</p> <p>DSLFNT NAME=BKEPREAG, NEXT=b7, ⋮</p> <p>In the following routing tables:</p> <ul style="list-style-type: none"> • DWSRTSK (corresponds to b17) • DWSL1IN • DWSL1OUT • DWSL2IN • DWSL2OUT • DWSL3GPI • DWSL3GPO • DWSL3FII • DWSL3FIO <p>specify: DSLROUTE TYPE=SET, TARGET=('b7'),</p>		TX2USESQ	
b9		<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Send Queue (Urgent) b9 • Send Queue (Normal) b9 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Send Queue (Urgent) b9 • Send Queue (Normal) b9 	<p>MLSNDURG (for urgent send queues)</p> <p>MLSNDNRM (for normal send queues)</p>	

Table 9. Additional Parameters for USE (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b10		<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Receive Queue b10 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Receive Queue b10 	MLRECEIV	
b12	<p><u>MERVA ESA</u></p> <p>In the logical terminal table:</p> <pre>DWSLT TYPE=MAS, USENAME=b12, :</pre> <p>In the routing table DWSRTSK (corresponds to b17)</p> <pre>DSLROUTE TYPE=TEST, COND=(...,b12,...), :</pre>		USEMERVA2	
b13	<p><u>MERVA ESA</u></p> <p>In the logical terminal table:</p> <pre>DWSLT TYPE=MAS, :</pre> <pre>,TFLAG=SLS ,SKEYQ=b13</pre> <p>In the function table DWSFN TTC:</p> <pre>DSL FNT NAME=b13 :</pre>		SLSGPA	
b14	<p><u>MERVA ESA</u></p> <p>In the logical terminal table:</p> <pre>DWSLT TYPE=AP, NAME=FIN, :</pre> <pre>,SKEYQ=b14</pre> <p>In the function table DWSFN TTC:</p> <pre>DSL FNT NAME=b14 :</pre>		SLSFIN	

Table 9. Additional Parameters for USE (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b17	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, ROUTSK=b17, : :</p> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=b17 : : • In the offline utility: //SYSIN DD * DEFINE PROGRAM(b17) : : 		DWSRTSK	
b23	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, NAME=b23, : :</p> <p>In the following routing tables:</p> <ul style="list-style-type: none"> • EKARTTXU (corresponds to b18) • DWSL2OUT • DWSL3GPO <p>specify: DSLROUTE TYPE=TEST, COND=(...,b23,...),</p>		VNDDEBET0A VNDOBET0A VNDPBET0A	

Additional Parameters for Telex

Table 10 contains additional customizable values for the workstation based Telex functions.

Table 10. Additional Parameters for Telex

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b7	<p><u>MERVA ESA</u></p> <p>In the function table DWSFNNTC: DSLNT NAME=b7, : :</p> <p>In the routing table ENLR2TLC (corresponds to b19): DSLROUTE TYPE=SET, TARGET=('b7'), : :</p>		TX2USESQ	
b9		<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Send Queue (Normal) b9 <p><i>and</i></p> <p>In the MERVA Link definitions that are part of the ASP customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Send Queue (Normal) b9 	MLSNDNRM	
b19	<p><u>MERVA ESA</u></p> <p>In the function table ENLFNTT2: DSLNT NAME=TX2TLC, ROUTE=b19, : :</p> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=b19 : : • In the offline utility: //SYSIN DD * DEFINE PROGRAM(b19) : : 		ENLR2TLC	

Table 10. Additional Parameters for Telex (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
b20	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, SENDQC=(b20), ⋮</p> <p>In the following function tables:</p> <ul style="list-style-type: none"> • DWSFNNTC • ENLFNTT2 <p>specify: DSLFNT NAME=b20,</p>		TX2USESQ, TX2URG, TX2NRM	
b21	<p><u>MERVA ESA</u></p> <p>In the MERVA Link partner table: EKAPT TYPE=ASP, IRROUTE=(ACK,b21,CTLQ), ⋮</p> <p>In the function table ENLFNTT2: DSLFNT NAME=b21, ⋮</p> <p>,NEXT=b21, ⋮</p> <p>In the routing table EKARTTXU (corresponds to b18): DSLROUTE TYPE=SET, TARGET=('b21'), ⋮</p>	<p>In the MERVA Link definitions that are part of the customization program in MERVA Link NT:</p> <ul style="list-style-type: none"> • Select Status Report for Telex 	TX2WAIT	

Chapter 3. Administering the USE Functions (MERVA Link)

This chapter describes step-by-step procedures you can use to verify that the USE functions have been correctly installed and customized, and describes the message routing for the USE functions.

Verifying the USE Functions

This section describes a step-by-step procedure you can use to verify that the workstation based USE functions have been installed and customized correctly.

The host part of the USE functions is installed automatically when you install MERVA ESA. See "Chapter 2. Connecting MERVA ESA to the USE Functions via MERVA Link" on page 9 for details of how to customize the host part.

Refer to *MERVA USE & Branch for Windows NT Installation and Customization Guide* for details of the installation and customization procedure for the workstation part. Carry out the following procedures after you have completed installation and customization of the USE functions to verify that the MERVA Link connection and the SLS service are working correctly.

Verifying the MERVA Link Connection with MERVA USE & Branch

1. Start the IBM Communications Server by clicking **Start -> Programs -> IBM Personal Communications -> Administrative and PD Aids -> SNA node operations**.
2. Ensure that MERVA ESA is running.
3. Select LU 6.2 Sessions.
The LU 6.2 session to the partner system is displayed.
4. Press PF5 to refresh the LU 6.2 sessions screen.
5. Start MERVA USE & Branch:
 - a. Click **Start -> Programs -> MERVA USE & Branch -> Control Center**.
 - b. Double-click **Local System**.
 - c. Select a MERVA instance, for example, **merva1**.
 - d. Select **Instance -> Start multi user mode**.
 - e. When the message **ENM2167I MERVA successfully started** appears, click **Start -> Programs -> MERVA USE & Branch -> Main Menu**.
6. From the Main Menu, select **Communications**, then **MERVA Link**.
7. Click on the partner ASP.
8. Select **Selected -> Start ASP**.
The ASP switches to nohold. If there is no information in the **Diag** column, the connection to the partner system was successful.
9. Sign on to MERVA ESA.
The Function Selection panel is displayed.
10. Select the **MSC** function.

Note: The MERVA ESA User Profile must contain the MSC function.

- The MERVA System Control (MSC) panel is displayed.
11. Press PF9 (SWAP).
The MERVA Link List of Message Transfer Applications panel is displayed. Under the field **ASP Name**, locate the ASP name **b2** that you defined in your Partner Table EKAPT.
 12. Select this ASP and press PF5 (Start).
The MERVA Link connection to the workstation is started.
 13. Press Enter one or more times until the current date and time are displayed.
Another indication of successful verification is that no error message, such as "Critical error ...", is displayed on the workstation.

Verifying the SLS Service

To verify that the SLS service is functioning correctly:

1. Sign on to MERVA ESA (see previous description).
The Function Selection panel is displayed.
2. Select the function **CMD** or **MSC**.
The appropriate panel is displayed.
3. At the workstation, log on to MERVA ESA USE feature (see previous description).
4. Select **MERVA Link operating**.
The MERVA Link Operating panel is displayed.
5. Select **Update MERVA Link queue status**.
The Update MERVA Link Queue Status panel is displayed.
6. Type **S** in the input field to set the queue to "nohold" status.
7. Confirm selection by pressing Ctrl+Enter.
8. Ensure that the SCR or BCR is ready, the card reader is connected to the workstation, a USER ICC for the destination is inserted into the card reader, and that the appropriate PIN for the card is entered.
9. In the MERVA ESA panel, enter **login** in the command line.
10. If the **login** command is accepted, enter the **dl** command. In its response, in the column STATUS, "SK PRC" (session key proceeding) is displayed.
11. After some time, enter the **dl** command again. In the response, the status should have changed to "LIN PRC" (login proceeding).
12. Enter the **dm last** command. The response should, for a switched line, show the messages:

```
DWS602I ltname LOGIN proceeding on line l
:
```

If the message DWS602I is displayed, the request for an SLS session key was successfully implemented by the card reader and verification is successful.

Data Flow for the USE Functions

This section describes the data flow for the workstation based USE functions. Sample definitions for all MERVA ESA queues mentioned in this section are included in the copy book DWSFNNTC. The copy book DWSFNNTC is contained in the MERVA ESA macro library.

All MERVA ESA routing tables mentioned in this section are contained in the MERVA ESA source library.

Data Flow of Pregenerated Session Keys for SLS

With the Secure Login and Select (SLS) service, session keys are “pregenerated” on the USE workstation.

Figure 15 shows the data flow for pregenerated session keys.

When pregeneration is complete, you send the session keys from the USE workstation to MERVA ESA with MERVA Link. To do this, an MT 999 with a special format is created, containing up to 34 session keys. Sufficient MT 999 messages are created to send all the session keys.

In the receiving MERVA Link of MERVA ESA, the routing table EKARTTXU, defined for the control queue TX2USECQ, identifies the MT 999 messages and routes them to the session key load queue SLSLOAD.

From the SLSLOAD queue, the session key load program DWSDLSK receives the MT 999 messages and verifies that the logical terminal (LT) name in the basic header is defined in DWSLTT and that session key queues are defined for this LT or its FIN application.

If so, the session keys are stored in the appropriate session key queue, for example, in SLSFIN or SLSGPA.

During LOGIN or SELECT requests, the session keys are retrieved from the session key queues by the program DWSDGPA.

Refer to the *MERVA for ESA Operations Guide* for more detailed information about preloaded session keys and how to maintain them.

Data Flow of Session Keys Requests for SLS

If preloaded session keys are not used, the SWIFT Link generates a special MT 999 for each LOGIN and SELECT request and sends it to MERVA USE & Branch to request the session key needed for the LOGIN or SELECT. Figure 15 contains an overview of the flow of session key request messages for SLS.

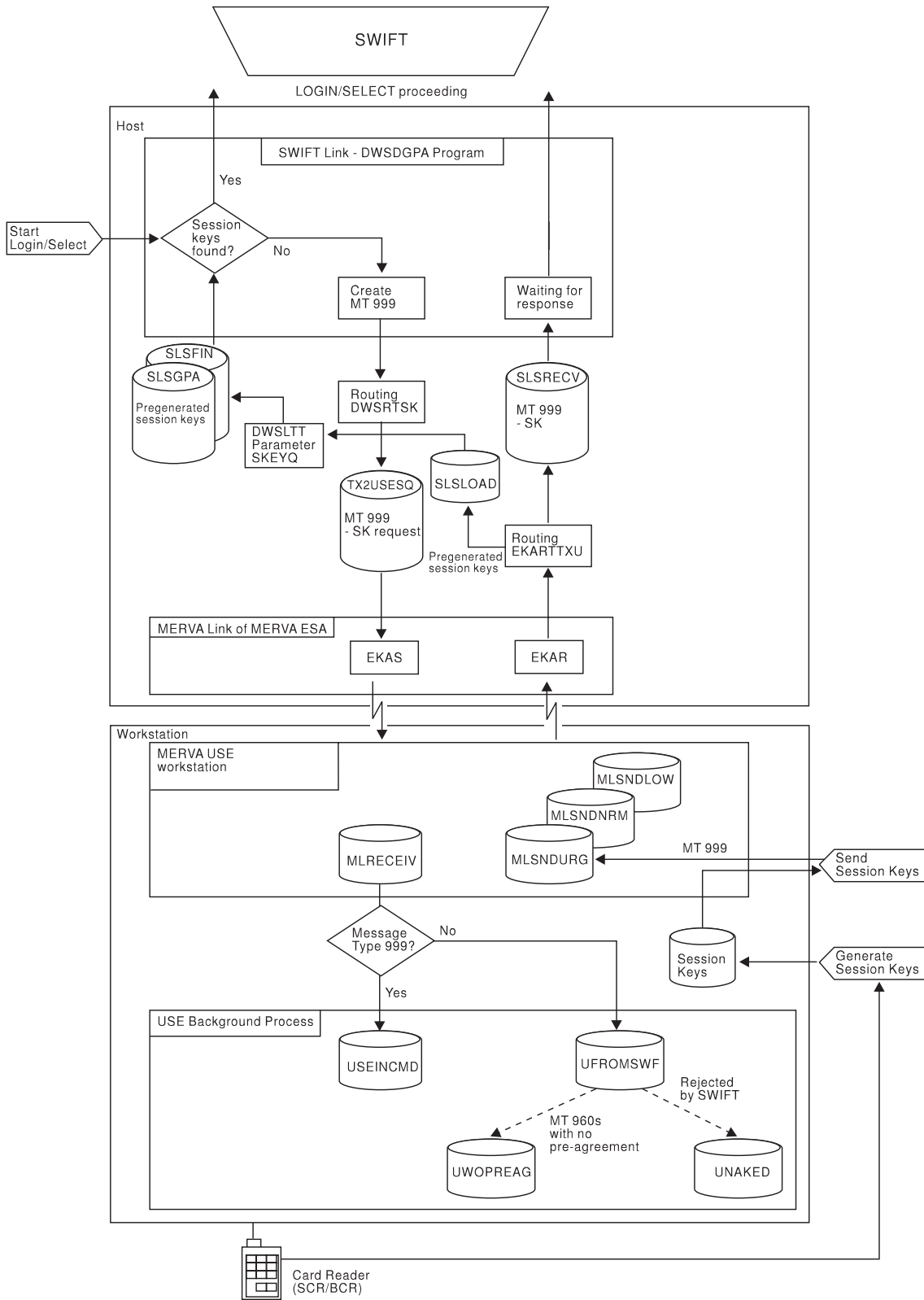


Figure 15. Flow of Session Key Request Messages for SLS and Pre-generated Session Keys

After the LOGIN or SELECT request has been accepted by the SWIFT Link and the command response indicates that the "session key pending" status has been set, the main program of the SWIFT Link, DWSDGPA, proceeds as follows:

1. If there is a session key queue defined for this master LT or its FIN application, DWSDGPA attempts to get the session key for the current LSN or SSN and the current ICC parameters from this queue.
If the session key is found, the login or select can proceed.
2. If there is no session key queue defined, or the session key could not be found in the queue, a special MT 999 message for a session key request is generated.
In this MT 999, the LT name is written to the basic header and the value of the USERNAME parameter of the DWSLT macro written to the application header as the receiving LT name.

When a special MT 999 message is generated, it is routed using the routing table DWSRTSK, which is defined with the ROUTSK parameter of the DWSLT macro. This routing table determines the name of the MERVA Link send queue TX2USESQ for sending the session key request message to MERVA USE & Branch.

If more than one workstation is used for SLS, DWSRTSK evaluates the LT name of the application header for routing to more than one MERVA Link send queue.

When MERVA USE & Branch receives the session key request message, it either uses pregenerated session keys if any are available, or requests the session key from the card reader. If the session key is found, it is written into the session key request message. Otherwise, appropriate error information is added.

The session key request message is then routed to the MERVA Link send queue MLSNDURG, from where MERVA Link sends it to MERVA ESA.

In the receiving MERVA Link in MERVA ESA, the routing table EKARTTXU (defined for the control queue TX2USECQ) identifies the MT 999 messages as being session key requests and routes them to the session key receive queue SLSRECV. This queue is defined both in DWSPRM (RSKQUE parameter of the DWSPARM macro) and in DSLFNTT.

From this queue, the program DWSDGPA gets the MT 999 messages and verifies in DWSLTT that this LT or FIN application is authorized to use this session key. If so, it is stored in the appropriate entry of DWSLTT.

If this LT or FIN application is also waiting for this session key (session key proceeding "SK PRC" status), login or select can proceed.

Data Flow of USE Messages

Both, USE system messages and BKE messages, have the following types:

1. Input messages. These messages are generated by MERVA USE & Branch and sent to MERVA ESA with MERVA Link for sending to the SWIFT network. The USE system messages for input are the message types MT 074, MT 075, MT 085, and MT 090. The BKE messages for input are the message types MT 960 to MT 967 (the same as output messages).
2. Output messages. These messages are received from the SWIFT network and sent to MERVA USE & Branch with MERVA Link. The USE system messages for output are the message types MT 076, MT 087, and MT 092. The BKE messages for output are the message types MT 960 to MT 967 (the same as input messages).

Figure 16 on page 69 contains an overview of the flow of USE system messages and BKE messages.

Flow of Input Messages

USE and BKE input messages are generated in MERVA USE & Branch, either by a USE administration function or by the BKE process. After generation, the messages with message type MT 999 are routed to the MERVA Link send queue MLSNDURG, all other messages are routed to the MERVA Link send queue MLSNDNRM. From there they are sent to MERVA ESA over the MERVA Link.

In the receiving MERVA Link of MERVA ESA, the routing table EKARTTXU defined for the control queue TX2USECQ recognizes the USE and BKE input messages and routes them to the appropriate ready queue for sending to the SWIFT network.

The program DWSDGPA eventually sends them to the SWIFT network and receives an acknowledgment. Sending, however, may not be possible due to an error.

In either case, the routing table (DWSL1IN, DWSL2IN, DWSL3GPI, or DWSL3FII) recognizes the USE and BKE input messages and, in addition to routing the messages to acknowledgment (L1ACK) or correction queues (L1VE0), routes the messages to the MERVA Link send queue TX2USESQ for sending back to MERVA USE & Branch. Acknowledgment or error information is returned in the MSGACK field of the messages.

When the messages are received in MERVA USE & Branch, status information in the USE or BKE processes is updated accordingly.

Flow of Output Messages

On receiving a USE or BKE output message, the SWIFT Link routing table (DWSL1OUT, DWSL2OUT, DWSL3GPO, or DWSL3FIO) identifies the USE or BKE output message and, in addition to the usual routing to display output (L1DO0) or sequential data set output queues (L1SDO), routes the messages to the MERVA Link send queue TX2USESQ for sending to MERVA USE & Branch.

When the messages are received in MERVA USE & Branch, status information in the USE or BKE processes is updated accordingly and a BKE process continued by generating the necessary responses.

On completion of bilateral key exchange, the MERVA USE & Branch authenticator-key file is updated and an update message (a special MT 999) generated and sent to MERVA ESA with MERVA Link.

In MERVA ESA, the receiving MERVA Link routing table EKARTTXU routes this special MT 999 to the queue BKEUPDT. From this queue, the transaction program DWSAUTT receives the message and invokes DWSAUTP to update the MERVA ESA authenticator-key file.

Note: If you want to maintain manual keys in the authenticator-key file, you can specify the parameter AUTHMAN=MERVAESA in the DWSPARM macro. In this case, manual keys are not updated by BKE keys and records with manual keys cannot be deleted.

But if the FROM date of a new BKE key overlaps the TO date of an existing manual key, the TO date of the manual key will be replaced by the FROM date of the BKE key. So you can start the BKE processing at any time without regarding the TO dates of the manual keys.

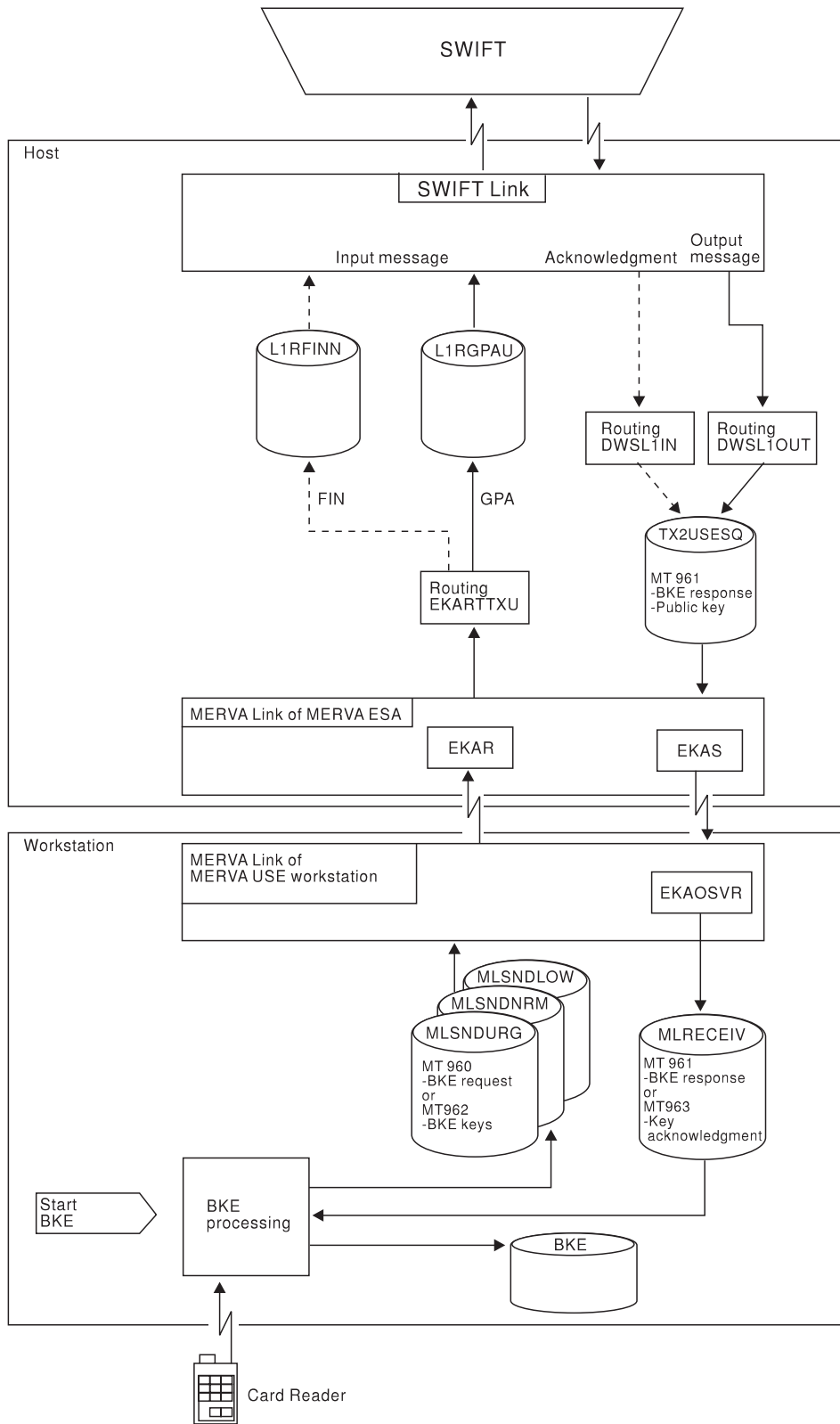


Figure 16. Message Flow for BKE Started from MERVA

Workstation Message Queues

The following MERVA USE & Branch message queues are involved in the data flow of workstation based USE functions:

- MLRECEIV** This is the MERVA Link receive queue for all messages arriving from a partner MERVA Link.
- MLSNDURG, MLSNDNRM, MLSNDLOW**
These are the MERVA Link send queues for all messages sent to a partner MERVA Link.
- USEINCMD** This queue is used for internal USE processing within MERVA USE & Branch, in particular SLS session key requests from MERVA ESA.

Messages of type MT 999 arrive in this queue and are dealt with by the USE background process.
- UFROMSWF** This queue contains USE messages that have been received from the SWIFT network.

BKE messages (MT 96x) arrive in this queue and are dealt with by the USE background process.
- UNAKED** This queue contains USE messages that have been sent to the SWIFT network and been negatively acknowledged.

The USE background process uses the NAK information, which is contained in the MSGACK field of the message, to update the status of the BKE record for the correspondent.
- UWOPREAG** This queue contains BKE initiation request messages.

The messages in this queue could not be handled automatically by the USE background process because no approved preagreement exists for the correspondent.

These messages can be processed using the Incoming MT 960 program in the MERVA USE & Branch UKMO folder or menu.

For a more detailed description of these message queues, refer to the *MERVA USE & Branch for Windows NT Installation and Customization Guide*.

Chapter 4. Telex with Workstation API Functions (MERVA Link)

This chapter describes step-by-step procedures you can use to verify that the Telex functions have been correctly installed and customized, and describes the message routing for the Telex functions.

Installation and Customization of the Telex Functions

MERVA ESA allows you to use the workstation based Telex functions as a means of sending messages to and receiving messages from the public telex network. It uses MERVA API services to do this, as illustrated in Figure 17.

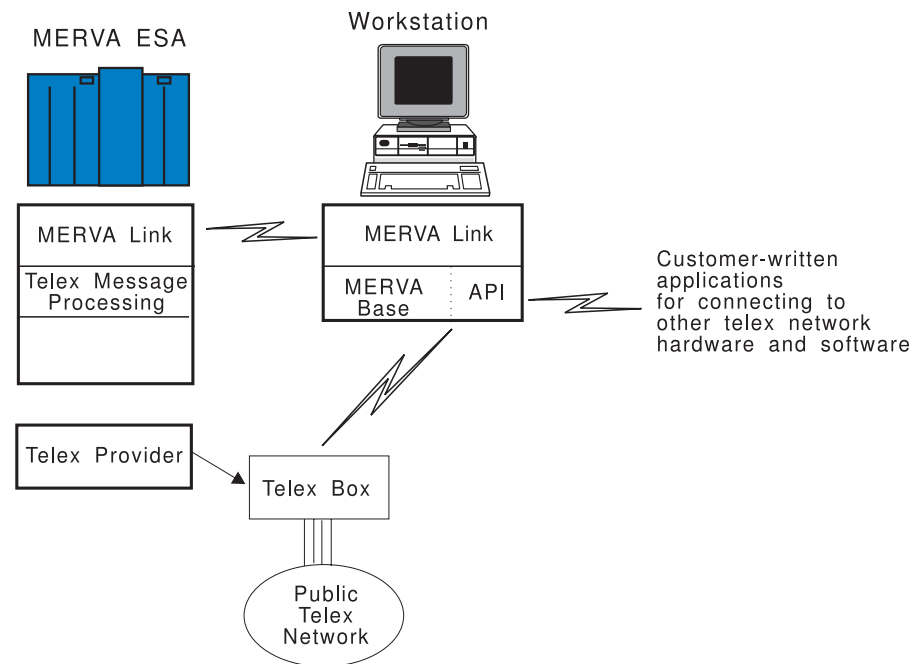


Figure 17. Components of MERVA ESA and MERVA USE Workstation Required for Telex Functions

The host part of the workstation based telex functions is installed automatically when you install MERVA ESA. See “Chapter 2. Connecting MERVA ESA to the USE Functions via MERVA Link” on page 9 and “Additional MERVA ESA Customizing Information” on page 73 for details of how to customize the host part.

On the workstation only API functions are used.

Customizing MERVA ESA Resources for the Telex Functions

When customizing MERVA ESA for telex, you work with the following parameters:

- PC LU Name (a8 and d2)
- PC Receiving TP Name (a9)
- Host MERVA Link Local Node Name (b1)
- Host ASP Name (b2)

- Workstation ASP Name (**b3**)
- Host Internal MTP Name (**b4**)
- Host External MTP Name (**b5**)
- Workstation MTP Name (**b6**)
- Host ASP Control Queue (**b8**)
- Host ASP Security Options (**b11**)
- Host ASP User Exit (**b15**)
- Host Message Format (**b16**)
- Host ASP Routing Table (**b18**)
- Host Telex Station Command Routing Table (**b19**)
- Host ASP Send Queue Cluster (**b20**)
- Host ASP Incoming Report Routing Options (**b21**)
- Host ASP Journaling Options (**b22**)
- CICS Partner System ID (**c1**)
- Host Sending Transaction ID (**c5** and **d5**)
- Host Sending TP Name (**d1**)
- APPC/MVS LU Name (**d3**)
- SNA Mode Name (**d4**)

The following MERVA ESA tables are affected:

- DSLFNNTT
- DSLMPTT
- EKAPT
- EKARTTXU
- ENLR2TLC

Perform the following steps:

1. Define the parameters of an ASP and an MTP entry in your MERVA Link Partner Table EKAPT. These PT entries describe the Telex application and the connection to the workstation in MERVA Link ESA. The source member EKAPTTXU, which is supplied with the MERVA ESA base product, contains sample PT generation statements.

For the LU name (**a8** and **d2**) of the workstation and the mode name (**d4**) use the values specified in “Customizing VTAM Resources” on page 15.

EKAOSVR is the PC Receiving TP Name that must be specified in the MTP entry (parameter **a9**).

There are default values for the Host MERVA Link Local Node Name (**b1**), Host ASP Name (**b2**), Workstation ASP Name (**b3**), Host Internal MTP Name (**b4**), Host External MTP Name (**b5**), and Workstation MTP Name (**b6**). The values are the same as those in the default configuration in MERVA ESA V4 USE feature. You can also choose your own values.

Defaults for the MERVA Link Send Queue Cluster (**b20**) and MERVA Link Control Queue (**b8**) are given. If you change these values, change the function table too.

If you want your telex messages to be authenticated and encrypted during the transfer from the host to the workstation, specify ENCRYPT in the parameter **b11** (Host ASP Security Options).

Parameter **b15** includes the exit number 7133 of the user-exit program EKAMU133, which sends telex messages to and receives telex acknowledgment-related information from MERVA USE & Branch. This program and the related entries in the MFS program table DSLMPTT are available when MERVA ESA is installed.

The format specification `FORMAT=(MCB,TCOV,#)` (**b16**) supports both the telex and the USE message exchange.

The `IRROUTE` parameter represents the Host ASP Incoming Report Routing Options (parameter **b21**). It specifies on which queue incoming reports are correlated. If you change this value, you must also change the function table. Do not, however, modify the other parameters of `IRROUTE`.

The defaults specified for the Host ASP Journaling Options (**b22**) result in the journaling of incoming messages, outgoing messages, and incoming reports. You can choose your own values.

For the sending transaction ID of `MERVA ESA` (**c5** and **d5**), use the names specified in “Customizing CICS Resources” on page 15 and “Customizing IMS Resources” on page 17.

If your `MERVA` system is running under CICS, the system identifier **c1** appears in the statement `LINK=(APPC,c1)` in the `MERVA` Link Partner Table. For **c1**, use the value specified in “Customizing CICS Resources” on page 15.

If your `MERVA` system is running under IMS, the `LINK` statement is `LINK=(APPC,d2,d4)`. For **d2** and **d4**, use the values specified in “Customizing VTAM Resources” on page 15.

For `MERVA ESA IMS`, you must add the parameter `MTPCI=(EKATPO1)` to the MTP generation macro (parameter **d1**) of the partner table.

For `APPC/IMS`, you must add the parameter `LOCLU=d3` to the MTP generation macro and specify the name of the `APPC/IMS` Base LU as parameter **d3**. In the `APPC/MVS` environment the `APPC/MVS` System Base LU is used by default as the local LU for outbound conversations.

Modify the PC LU Name **a8** (`DEST=(PS2LU00,...)`).

Assemble and link-edit your `MERVA` Link Partner Table. The load module name must be `EKAPT`.

See the examples in Figure 42 on page 156, Figure 43 on page 156, Figure 44 on page 156, and Figure 45 on page 157, and the *MERVA for ESA Macro Reference* for further details.

2. Add the new functions (queues) to your function table (`DSLFNNTT`).

An example of a set of functions that is sufficient for telex is included in the copy book `ENLFNNTT2`. In addition, you need copy book `DWSFNNTTC` for the telex command send queue and the `MERVA` Link control queue. Both queues are commonly used by the `MERVA ESA V4 USE` feature. The copy books `ENLFNNTT2` and `DWSFNNTTC` are shipped with your `MERVA ESA` system.

3. Check that you have installed the new routing tables `EKARTTXU` (**b18**) and `ENLR2TLC` (**b19**) in your load library.

A reference to parameter **b18** must be made in the function table copy book `DWSFNNTTC`.

A reference to parameter **b19** must be made in the function table copy book `ENLFNNTT2`.

Additional `MERVA ESA` Customizing Information

The following contains information that compares the Telex functions with `MERVA ESA` functions. It points out the many similarities between the two, for example, the names of the new functions. See “Data Flow for the Telex Functions” on page 76.

It also contains a description of the routing tables for outgoing and incoming telexes.

Telex Correspondents File

The telex correspondents file is used by the Telex functions. See the *MERVA for ESA Installation Guide* for information about the telex correspondents file.

Test-Key Considerations

Test-key processing is not affected by the changes for Telex function support. See the *MERVA for ESA Customization Guide* for detailed information on test-key processing.

Parameter Module—ENLPRM

The following ENLPRM parameters are referenced by the Telex functions:

CMDQ	Queue name for telex commands.
CTLQ	Queue name for telex Control Queue, containing sequence number control message.
LINES	Defines the maximum number of lines for the checking of telexes. To modify the maximum number of text lines in an outgoing telex, you must also change the DACNT parameter in the ENLMTSND MCB.
TESTKEY	Module name of the test-key processing program.

For complete details about these parameters, refer to the *MERVA for ESA Macro Reference*.

MERVA Link MFS User Exit EKAMU034

The Telex functions use the MERVA Link component of MERVA ESA. The telex handling on the workstation requires a MERVA Link-unique message identification of a specific format. This specific message identification is created during the MERVA Link send process by the MERVA Link sample MFS user exit EKAMU034.

Functional Description of Sample User Exit EKAMU034

The EKAMU034 user exit:

- Determines whether a message identification was assigned during a previous send attempt.
- If no message identification was assigned during a previous send attempt, loads ENLPRM to obtain the name of the workstation based Telex functions control queue (parameter CTLQ of macro ENLPARM).
- Gets the control message (message type TCTL, MCB ENLMTCTL), which contains the ASP name in the KEY1 parameter of the workstation based Telex functions connection and checks the validity of the sequence number found there.
- Increments the sequence number by one, writes the updated sequence number into the TOF field ENLP2SEQ of the control message, and replaces the control message in the control queue.
- Writes the message identification into the TOF field EKAAMSID of the telex message that is being sent.
- If no control message is found on the control queue, creates a new control message with sequence number 00000001.

Maintaining the Sequence Number: You can modify the sequence number in the control message manually as follows:

1. Enter the TX2CTL function.

2. Display a queue list. Figure 18 shows an example queue list.

```

                                Queue Key List                                Func TX2CTL
                                MESAASP                                Wait 00000001

Key 1: EKAASP                    Key 2: ENLP2SEQ                    QSN      RBN
> MESAASP                        00000033                        000000001 000011

To select a message, move cursor to ">" and press PF4 (Get QSN)

Select only KEY 1:                                KEY 2:

Command =====>
PF 1=Help      2=Retrieve  3=Return   4=Get QSN   5=Get Next  6=Get First
PF 7=List Back 8=List Fwd  9=Hardcopy 10=List Last 11=List First 12=List Off

```

Figure 18. Queue List Display of Control Queue

Note: The queue list display of the workstation based Telex functions control queue can also be used to monitor the flow of telex messages that are sent.

3. Get the control message that shows the ASP name of the required Telex functions connection.

```

MT TCTL                                Telex Link Sequence Number Control                                Page 00001
                                MESAASP                                Func TX2CTL

ASP Name      : MESAASP      MERVA Link Application Support Process

Sequence Number: TF 00000033  Telex Sequence Number

Command =====>
PF 1=Help      2=Retrieve  3=EOM      4=Repeat   5=Get Next  6=Requeue
PF 7=Page -1   8=Page +1  9=Hardcopy 10=Pro Line 11=Prompt  12=Escape

```

Figure 19. Message Display of Control Message

4. Modify the sequence number shown on the display. Take care that the sequence number that is stored in the control message identifies the telex messages uniquely. Special care must be taken to avoid duplicate message identification sequences if you reduce the sequence number displayed.
5. Enter the **EOM** or **Requeue** command (or press the appropriate PF key) to store the modified control message in the Control Queue.

Exceptional Conditions of EKAMU034: The information about exceptional conditions in EKAMU034 is passed to the MFS return and reason code in the MSC display of the ASP. The display of the reason code is reduced by 512 because only one byte is available for the display. There may be common DSLMFS or DSLTSV reason codes or special reason codes introduced by EKAMU034. These common and special reason codes are described in *MERVA for ESA Messages and Codes*.

Customizing EKAMU034: The name of the control function can be customized using the CTLQ parameter of the ENLPARM macro. By default, TX2CTL is used.

The sample user exit EKAMU034 is distributed in a way that all 8 bytes of the sequence number are increased. However, it is not necessary to have such a large range to identify the telex messages uniquely within a specific period of time. As supplied, EKAMU034 contains instructions to identify the telex messages by the date followed by an ascending sequence number within this day. Refer to the comment section of EKAMU034 for information on how to activate this processing.

Data Flow for the Telex Functions

This section describes the data flow for the workstation based Telex functions.

The workstation based Telex functions implement a different routing path for telex messages from the Telex Link component of MERVA ESA. However, the message flow logic remains similar. Many of the functions have a direct counterpart in the Telex Link component. The addition of a **2** distinguishes the new function names. For example, the TX2DE0 function is the equivalent of the TXDE0 function.

All but two functions are defined in the ENLFNTT2 copy book. The workstation telex command send queue TX2USESQ and the MERVA Link control queue TX2USECQ are defined in the DWSFNNTC copy book.

Functions for Outgoing Telexes

The routing of telexes between the functions for outgoing telexes is controlled by the following routing tables:

ENLR2DE0	Routing of telexes following Data Entry
ENLR2VE0	Routing after SWIFT message verification
ENLR2AI0	Routing after SWIFT message authorization
ENLR2TKC	Routing after test-key calculation.

The functions used in the flow of outgoing telexes are:

TX2DE0	Workstation Telex Data Entry Entry and editing of free-format and formatted telexes, for example, a SWIFT message.
TX2AI0	Workstation Telex Authorization SWIFT Messages For authorizing SWIFT messages being sent as a telex.
TX2VE0	Workstation Telex Verification SWIFT Messages

	For verification, and possibly correction, of authorized SWIFT messages.
TX2TKCA	Workstation Telex Test-Key Calculation Automatic For automatic test-key calculation.
TX2TKC	Workstation Telex Test-Key Calculation For manual control of test-key calculation.
TX2NOTX	Workstation Telex No Telex Messages that are not recognized as either a telex or as a SWIFT message are routed to this queue.
TX2TKCER	Workstation Telex Test-Key Calculation Errors For processing telexes that failed test-key calculation.
TX2ACK	Workstation Telex Positively Acknowledged Telexes Telexes that have been successfully transmitted to the remote subscriber are routed to this queue.
TX2NAK	Workstation Telex Negatively Acknowledged Telexes Telexes that could not be transmitted to the recipient are routed to this queue.
TX2SDI	Workstation Telex Sequential Data Input For telexes loaded from a sequential file by utility DSLSDI.
TX2PR0	Workstation Telex Print Outgoing Telexes For hardcopy printing of telexes successfully transmitted to the recipient.

Figure 20 on page 78 contains an overview of the flow of outgoing telexes.

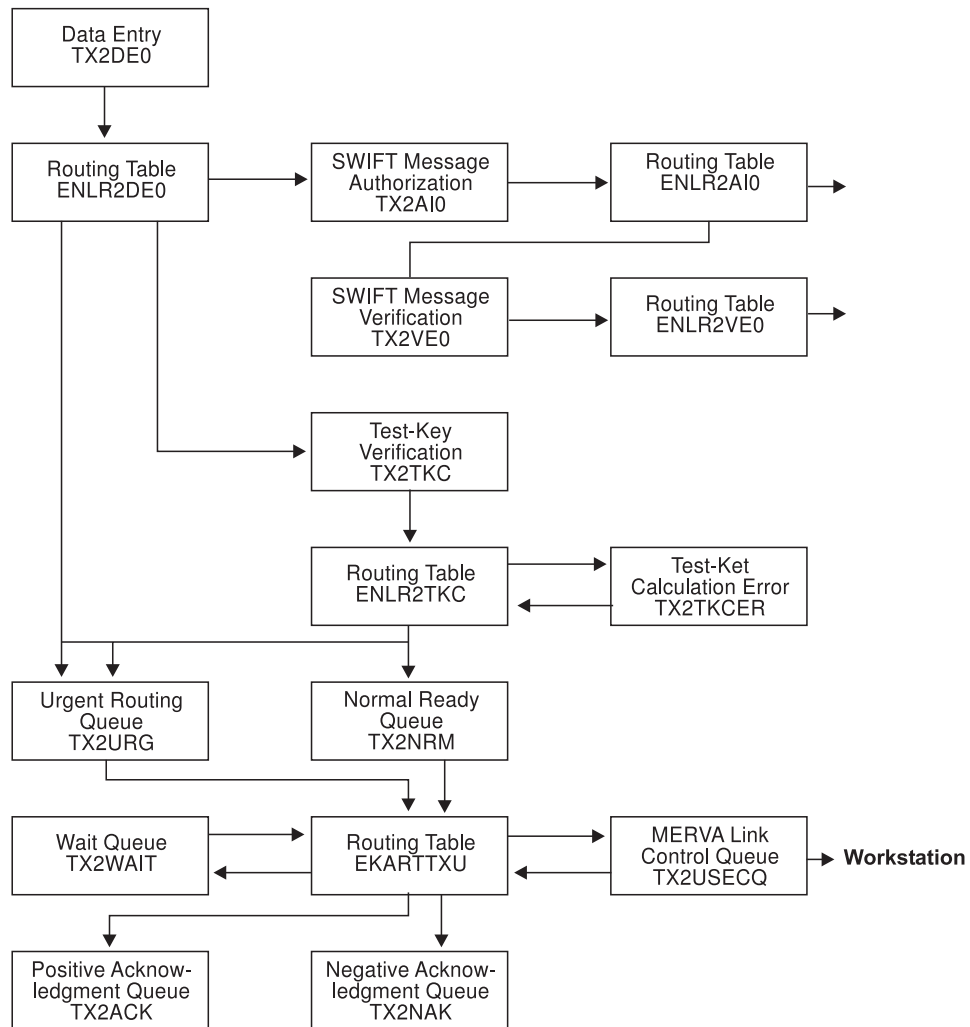


Figure 20. Routing Table Showing the Flow of Outgoing Telexes

Functions for Incoming Telexes

The routing of received telexes is controlled by the following routing tables:

- EKARTTXU** Routing of telexes from MERVA Link ESA.
- ENLR2TKV** Routing after test key verification. This routing table is a copy of the Telex Link of MERVA ESA table ENLR2TKV with the prefix for the target function names changed to **TX2**.

The functions involved in the flow of incoming telexes are:

- TX2RCV** Workstation Telex Received Telexes
All telexes received from MERVA USE & Branch are routed to this queue except those marked as a possible duplicate.
- TX2PDR** Workstation Telex Possible Duplicate Received
All telexes received from MERVA USE & Branch and marked by MERVA USE & Branch as a possible duplicate are routed to this queue.

- TX2TKV** Workstation Telex Test-Key Verification
For received telexes containing a test-key. They are routed to this queue for test-key verification.
- TX2TKVER** Workstation Telex Test-Key Verification Errors
If test-key verification fails, telexes are routed to this queue from TX2TKV.
- TX2DISTR** Workstation Telex Distribute Received Telexes
Final destination for telexes with a successful test-key verification.
- TX2INVR** Workstation Telex Invalid Received
MERVA Link routes acknowledgments to this queue that could not be matched to a telex in the TX2WAIT queue.
- TX2PR1** Workstation Telex Print Received Telexes
Use this function for hardcopy printing of received telexes.
- TX2SDO** Workstation Telex Sequential Data Output
For telexes to be unloaded to a sequential file by utility DSLSDO.
- TX2SDY** Workstation Telex Output to a System Printer
For telexes to be printed on a system printer by utility DSLSDY.

Figure 21 contains an overview of the flow of incoming telexes.

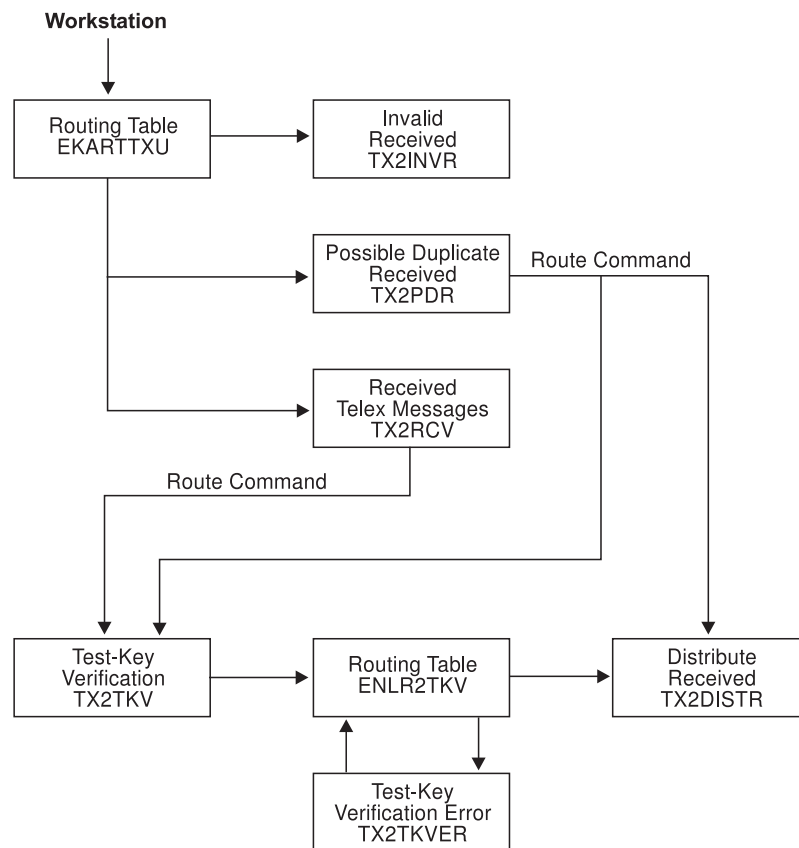


Figure 21. Routing Table Showing the Flow of Incoming Telexes

Functions for Telex Transmission to MERVA USE & Branch

The following functions are involved in the transmission of telexes to MERVA USE & Branch.

- TX2URG** Workstation Telex Urgent Ready Queue
- This function is defined with TRAN=EKAS to initiate the MERVA Link Send transaction. It is defined in the MERVA Link Partner Table as one of the MERVA Link Send Queues.
- TX2NRM** Workstation Telex Normal Ready Queue
- This function is also defined with TRAN=EKAS to initiate the MERVA Link Send transaction. It is also defined in the MERVA Link Partner Table as one of the MERVA Link Send Queues.
- TX2USESQ** Workstation Telex Command Send Queue
- TX2USESQ is also defined as a MERVA Link Send Queue. It is the ready queue for commands from function TX2TLC, and has the highest priority of the queues in the MERVA Link send queue cluster.
- TX2USESQ is defined in copy book DWSFN TTC.
- TX2WAIT** Workstation Telex Wait for Acknowledgments
- TX2WAIT is defined in the MERVA Link Partner Table **IRROUTE** parameter as the ACK Wait queue for incoming receipt reports. Once MERVA Link receives the confirmation that an outgoing message has been transferred to the Workstation, the message is routed to TX2WAIT to await the MERVA Link receipt report, that is the transmission acknowledgment. In the workstation based Telex functions application “logical” acknowledgments are not returned to MERVA ESA.
- When MERVA Link receives a receipt report from the MERVA USE & Branch system, it retrieves the telex with the corresponding message identifier from the TX2WAIT queue, adds the receipt report data to it, and the telex is then routed to the TX2ACK or TX2NAK queue. For this correlation of receipt report and telex, MERVA Link requires KEY1 of the TX2WAIT function to be the MERVA Link IAM message ID field, EKAAMSID.
- TX2USECQ** MERVA Link Control Queue for the workstation based telex functions application
- MERVA Link uses a control queue to support its Message Integrity Protocol. See the *MERVA for ESA Concepts and Components* for more information on message integrity.
- TX2USECQ is defined in copy book DWSFN TTC.

General Functions

- TX2CTL** Sequence number control queue. This function is used to store an ascending sequence number for outgoing telexes.
- TX2ERROR** Workstation Telex Error Queue
- A message is routed to this queue by any routing table that cannot route the message.
- TX2HOLD** Workstation Telexes Recovered from MERVA Link

In-process (IP) messages copied using the MERVA Link **recover** command from the control queue of a halted MERVA Link application are routed to this queue. From here you can **route** the message to another application for transmission, for example, a Telex Link of MERVA ESA application.

The IP message is not removed from the control queue. When the halt condition is cleared, the IP message is marked by MERVA Link as a Possible Duplicate (PDM) and transferred to the partner.

Part 3. Connections via MERVA-MQI Attachment

Chapter 5. Connecting MERVA ESA to the USE Functions via MERVA-MQI Attachment

This chapter describes the steps you must take to establish a connection, using the MERVA-MQI Attachment component, between your MERVA ESA system and the components installed on the workstation.

Overview

To establish a MERVA-MQI Attachment connection between MERVA ESA and MERVA USE & Branch, you need to customize the following:

Transaction Monitor Subsystem	CICS or IMS TM
Communication Software	MQSeries
MERVA-to-MERVA connection	The MERVA-MQI Attachment components of MERVA ESA and of MERVA USE & Branch

Figure 22 illustrates the components involved for CICS or IMS.

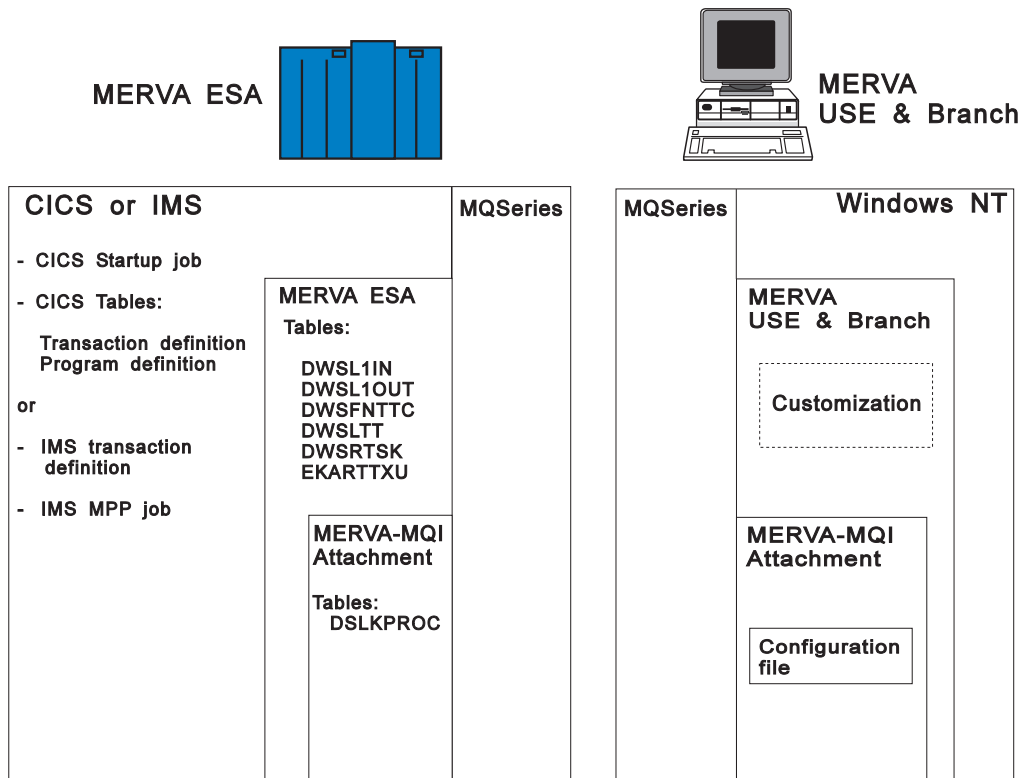


Figure 22. What You Need to Customize for CICS or IMS

Parameter Definitions

The workstation based USE functions require a connection from MERVA ESA on the host to the MERVA USE & Branch on the workstation. This connection is established by the MERVA-MQI Attachment components of MERVA ESA and MERVA USE & Branch. The communication between the host system and the workstation is based on MQSeries.

The parameters used to set up a connection between the host system and the workstation system identify resources or specify processing options. The identified resources are located either on the host or on the workstation. The terms *Host* and *PC* are used in the parameter names when the location of the resource is not evident.

Host Identifies a resource of MERVA ESA, MERVA-MQI Attachment, CICS, IMS, MQSeries, or any other subsystem that executes in the host environment.

PC or Workstation

Identifies a resource on which Windows NT, eNetwork Communications Server for Windows NT, DB2 Universal Database, MERVA USE & Branch, and MERVA-MQI Attachment run.

Refer to *MERVA for ESA Concepts and Components* for a description of the MERVA-MQI Attachment resources and its terminology.

The next tables list and describe all customization parameters that apply in a MERVA ESA installation including the USE functions. They also contain page references to the tables in "Parameter Relationships" on page 112. Thus you can see in which subsystem customization a specific parameter is used.

The parameters are grouped as follows:

Parameter	Group
e..	MQSeries customization (see Table 11)
f..	MERVA ESA and MERVA USE & Branch customization (see Table 12 on page 87)
g..	CICS customization (see Table 13 on page 89)
h..	IMS customization (see Table 14 on page 89)

Table 11. Customization Parameters for MQSeries

Parameter	Name	Description	Page
e1	Host Queue Manager	Name of the host queue manager.	112
e2	Workstation Queue Manager	Name of the workstation queue manager.	113
e3	Host Connection Name	Network address of the host, including the port number.	113
e4	Workstation Connection Name	Network address of the workstation, including the port number.	114
e5	Host Sender Channel	Name of the host sender channel.	114
e6	Host Receiver Channel	Name of the host receiver channel.	114
e7	Workstation Sender Channel	Name of the workstation sender channel.	115

Table 11. Customization Parameters for MQSeries (continued)

Parameter	Name	Description	Page
e8	Workstation Receiver Channel	Name of the workstation receiver channel.	115
e9	Host MQSeries Send Queue	Name of the host send queue which is the local definition of the workstation receive queue. The queue is used to send datagrams to the workstation.	115
e10	Host MQSeries Receive Queue	Name of the host receive queue. It receives incoming datagrams and request messages. The queue must be enabled for triggering.	116
e11	Workstation MQSeries Send Queue	Name of the workstation send queue which is the local definition of the host receive queue. The queue is used to send datagrams and request messages to the host.	116
e12	Workstation MQSeries Receive Queue	Name of the workstation receive queue. It receives incoming datagrams, reply messages, and report messages. The queue must be enabled for triggering.	116
e13	Host Initiation Queue	For MVS, the name of the host initiation queue on which the host queue manager puts trigger messages. The appropriate MQSeries trigger monitor serves the initiation queue.	117
e14	Workstation Initiation Queue (Trigger Queue)	Name of the workstation initiation queue on which the workstation queue manager puts trigger messages. The MERVA-MQI Attachment serves the initiation queue like a trigger monitor.	117
e15	Host Transmission Queue	Name of the host transmission queue.	118
e16	Workstation Transmission Queue	Name of the workstation transmission queue.	119
e17	Host Reply-to Queue	Name of the host reply-to queue. It receives incoming report messages. The queue must be enabled for triggering.	119
e18	Host MQSeries Control Queue	Name of the host control queue. The queue contains control messages used to assure the integrity of outgoing messages.	120
e19	Workstation Reference Queue	Name of the workstation reference queue. The queue contains reference messages used to correlate request messages with replies and reports, and datagrams with reports.	120
e20	Workstation Transaction Queue	Name of the workstation transaction queue. The MERVA-MQI Attachment uses this queue temporarily for its transactional processing. If not defined, the workstation reference queue acts as transaction queue also.	120
e21	Host User ID	Identification of the user who is authorized for the required MQSeries queues on the host.	120
e22	Workstation User ID	Identification of the user who is authorized for the required MQSeries queues on the workstation.	120
e23	Sequence Numbers Wrap	When reached, message sequence numbers wrap to start again at 1.	121

Table 12. Customization Parameters for MERVA ESA and MERVA USE & Branch

Parameter	Name	Description	Page
f1	Host MERVA Send Queue	Name of the MERVA ESA send queue. The MERVA-MQI Attachment copies each message from this queue to the appropriate host MQSeries send queue.	122

Table 12. Customization Parameters for MERVA ESA and MERVA USE & Branch (continued)

Parameter	Name	Description	Page
f2	Host MERVA Control Queue	Name of the MERVA ESA control queue. The queue temporarily contains each message before being sent or, after retrieval, before being routed to its target queue.	123
f3	Workstation MERVA Send Queue	Name of the MERVA USE & Branch send queue within the MERVA API purpose group. The MERVA-MQI Attachment copies each message from this queue to the appropriate workstation MQSeries send queue.	123
f4	Workstation MERVA Receive Queue	Name of the MERVA USE & Branch receive queue within the MERVA API purpose group. After retrieval, the MERVA-MQI Attachment copies each message from the workstation MQSeries receive queue to this queue.	123
f5	Host MERVA COA Wait Queue	Name of the MERVA ESA queue which contains messages waiting for the confirm-on-arrival report from MQSeries.	123
f6	Host MERVA COD Wait Queue	Name of the MERVA ESA queue which contains messages waiting for the confirm-on-delivery report from MQSeries.	123
f7	Workstation MERVA Wait Queue	Name of the MERVA USE & Branch queue within the MERVA API purpose group which contains messages waiting for a response.	123
f8	Host MERVA Start Queue	Name of the MERVA ESA queue to start a send or receive process of the MERVA-MQI Attachment.	124
f9	Host Security Options	Parameter that specifies whether or not the sent and received messages must be authenticated and encrypted.	124
f10	Workstation Security Options	Parameter that specifies whether or not the sent and received messages must be authenticated and encrypted.	124
f11	USE Workstation Name	<p>Defines the name of the USE workstation when SLS is used. Specify 1 to 9 characters conforming to the rules for SWIFT addresses. If you specify less than 9 characters, the name is padded with Xs.</p> <p>This name is used when using SLS connected mode for the routing of a single session key request to a MERVA-MQI Attachment send queue for sending to the USE workstation. A single session key request is a SWIFT message type 999 with a special format handled by the SWIFT Link programs. The name of the USE workstation is used in the application header as destination address and can be evaluated during the routing of this message.</p>	125
f12	Host Pregenerated Session Key Queue for GPA	Name of the MERVA ESA queue that is used for storing of the pregenerated session keys for the LOGIN function of the master logical terminal (GPA).	125
f13	Host Pregenerated Session Key Queue for FIN	Name of the MERVA ESA queue that is used for storing of the pregenerated session keys for the SELECT function of the financial application (FIN).	125
f14	Host User Exit	Number of the MERVA ESA MFS user-exit program DSLKQ133 defined in the MERVA ESA MFS Program Table (DSLMPPT). DSLKQ133 supplies the data, the SWIFT ACK or NAK, for a reply message sent subsequently to the workstation.	126
f15	Host Session Key Request Routing Table	Name of the MERVA ESA routing table which controls the routing for a session key request. The session key requests are routed to the host MERVA send queue (parameter f1).	126

Table 12. Customization Parameters for MERVA ESA and MERVA USE & Branch (continued)

Parameter	Name	Description	Page
f16	Host Routing Table	Name of the MERVA ESA routing table which controls the routing of all messages handled by the MERVA-MQI Attachment. This includes the SLS and BKE messages sent to and received from MERVA USE workstation, and the COA and COD reports received from MQSeries.	127
f17	Host Journaling Options	Parameters in the MERVA-MQI Attachment of MERVA ESA that specify journaling options.	127
f18	Host Master Logical Terminal Name	Name of the master logical terminal used for the LOGIN function (GPA).	127

Table 13. CICS Customization Parameters

Parameter	Name	Description	Page
g1	Host Receiving Transaction ID	CICS transaction ID of the MERVA-MQI Attachment receiving transaction (for example, DSLR). This transaction ID is associated with the Host Receiving Program name.	128
g2	Host Receiving Program Name	The name of MERVA-MQI Attachment program that handles incoming messages.	129
g3	Host Sending Transaction ID	CICS transaction ID of the MERVA-MQI Attachment sending transaction (for example, DSLS). This transaction ID is associated with the Host Sending Program Name.	129
g4	Host Sending Program Name	The name of MERVA-MQI Attachment program that handles outgoing messages.	130

Table 14. IMS Customization Parameters

Parameter	Name	Description	Page
h1	Host Receiving Transaction ID	IMS transaction ID of the MERVA-MQI Attachment receiving transaction (for example, DSLR). This transaction ID is associated with the Host Receiving Program name.	131
h2	Host Receiving Program Name	The name of MERVA-MQI Attachment program that handles incoming messages.	131
h3	Host Sending Transaction ID	IMS transaction ID of the MERVA-MQI Attachment sending transaction (for example, DSLS). This transaction ID is associated with the Host Sending Program Name.	131
h4	Host Sending Program Name	The name of MERVA-MQI Attachment program that handles outgoing messages.	132

Customizing the Host Resources

This section describes how to adapt your host system's parameters for use with the MERVA USE & Branch feature. The host parameters cover the following:

- CICS
- IMS
- MQSeries
- MERVA ESA

Customizing CICS Resources

When customizing CICS resources, you work with the following parameters:

Name	Parameter ID	Comment
Host User Exit	f14	The default value must be used unchanged.
Host Session Key Request Routing Table	f15	The default value might be acceptable.
Host Routing Table	f16	The default value might be acceptable.
Host Receiving Transaction ID	g1	The default value might be acceptable.
Host Receiving Program Name	g2	The default value must be used unchanged.
Host Sending Transaction ID	g3	The default value might be acceptable.
Host Sending Program Name	g4	The default value must be used unchanged.

These parameters are described in Table 12 on page 87 and Table 13 on page 89.

Note: In the following description, *xx* identifies the CICS release level. For example, DSLCSD41 refers to CICS Version 4 Release 1. The identifier *xx=21* is an exception. It is used only for macro level CICS resource definitions which are also valid in a higher release level of CICS.

Do the following:

1. Ensure that your CICS system is enabled for MQSeries. For more information, see the chapter of the *MERVA for ESA Installation Guide* that describes how to verify the installation of MERVA-MQI Attachment. A sample is shown in "Sample CICS Startup Job" on page 190.
2. Do one of the following:
 - Use the CICS macro level interface to modify the following tables in your CICS system (see "Defining CICS Resources using the CICS Macro Level Interface" on page 165):

CICS Transaction Definition

Define the following local MERVA-MQI Attachment transactions for CICS:

DSLS The MERVA-MQI Attachment sending transaction is assigned to program DSLKQS (**g3** and **g4**).

DSLRL The MERVA-MQI Attachment receiving transaction is assigned to program DSLKQR (**g1** and **g2**).

The copy book DSLPCT21, which is shipped with your MERVA ESA system, contains sample values for these parameters. You can include the copy book in your program control table (PCT). "CICS Transactions" on page 165 shows a sample CICS program control table with the appropriate entries.

CICS Program Definition

Define the following programs for CICS:

DSLKPROC MERVA-MQI Attachment process table

DSLKQS	Sending program (g4)
DSLKQR	Receiving program (g2)
DSLKQ133	User-exit program to handle SWIFT acknowledgments and negative acknowledgments (f14)
EKARTTXU	Routing table for the Telex and USE functions to exchange messages with the workstation (f16)
DWSRTSK	Routing table for the USE functions to request session keys from the workstation (f15)
DWSLTT	Logical terminal table

Note: This is not a complete list of all mandatory MERVA-MQI Attachment programs. The complete list can be found in copy book DSLPPT21.

The copy books DSLPPT21 and DWSPPT21, which are shipped with your MERVA ESA system, contain sample values for these parameters. You can include the copy books in your processing program table (PPT). “CICS Processing Programs” on page 165 shows a sample CICS processing program table.

- Use the the offline CSD utility program DFHCSDUP and copy books DSLCSDxx and DWSCSDxx to generate the necessary table entries (see “Defining CICS Resources using the CSD Utility” on page 166).

Usually, these entries are already included in the customization of the MERVA ESA base product.

Customizing IMS Resources

When customizing IMS resources, you work with the following parameters:

Name	Parameter ID	Comment
Host Receiving Transaction ID	h1	The default value might be acceptable.
Host Receiving Program Name	h2	The default value must be used unchanged.
Host Sending Transaction ID	h3	The default value might be acceptable.
Host Sending Program Name	h4	The default value must be used unchanged.

These parameters are described in Table 14 on page 89.

Ensure that your IMS system is enabled for MQSeries. For more information, see the chapter of the *MERVA for ESA Installation Guide* that describes how to verify the installation of MERVA-MQI Attachment. “Sample IMS MPP Job” on page 191 shows a sample MPP job for IMS.

Define the host receiving and sending transaction IDs (parameters **h1** and **h3**) as IMS transactions (for example, DSLR and DSLS):

- The Host Receiving Program Name (parameter **h2**) that must be assigned to the transaction **h1** is DSLKQR.

- The Host Sending Program Name (parameter **h4**) that must be assigned to the transaction **h3** is DSLKQS.

“Sample for Defining IMS Resources” on page 167 shows a sample IMS transaction definition.

Customizing MQSeries Resources

When customizing MQSeries resources, you work with the following parameters:

Name	Parameter ID	Comment
Host Queue Manager	e1	The default value must be changed to the name of the existing queue manager.
Workstation Queue Manager	e2	The default value might be acceptable.
Host Connection Name	e3	The default value must be changed to the name of the existing connection.
Workstation Connection Name	e4	You will definitely need to set the value for this parameter.
Host Sender Channel	e5	The default value might be acceptable.
Host Receiver Channel	e6	The default value might be acceptable.
Host MQSeries Send Queue	e9	The default value might be acceptable.
Host MQSeries Receive Queue	e10	The default value might be acceptable.
Workstation MQSeries Receive Queue	e12	The default value might be acceptable.
Host Initiation Queue	e13	For MVS, you will definitely need to set the value for this parameter.
Host Transmission Queue	e15	The default value might be acceptable.
Host Reply-to Queue	e17	The default value might be acceptable.
Host MQSeries Control Queue	e18	The default value might be acceptable.
Host User ID	e21	The default value might be acceptable.
Sequence Number Wrap	e23	The default value might be acceptable.

These parameters are described in Table 11 on page 86.

The Host Queue Manager (**e1**) and the port number of the Host Connection Name (**e3**) are specified when MQSeries is configured. When setting up the MQSeries connection from the workstation, the Host Queue Manager and Host Connection Name including port number are required.

The Host User ID (**e21**) is the user ID that is authorized to use the required MQSeries resources:

- Under MVS, this can be the user ID associated with the CICS or IMS transactions DSLR (parameter **g1** or **h1**) and DSLS (parameter **g3** or **h3**). This might be the default user ID specified in the CICS startup job, or the PSB name of the MERVA ESA nucleus program DSLNUC under IMS.

Under MVS, you can also specify an alternate user as Host User ID (see parameter **e21** in “Samples for Defining MERVA-MQI Attachment Resources” on page 173). The alternate user can use the authority of another CICS or IMS user to use an MQSeries resource.

- Under VSE, the Host User ID is the ID of the CICS user associated with the transaction DSLR (**g1**) and DSLS (**g3**).

All other parameters are specified when you define MQSeries resources such as queues and channels:

- Under MVS, you can define the MQSeries resources using the MQSeries batch utility CSQUTIL. How to do this is described in “Defining MQSeries Resources under MVS”.
- Under VSE, use the MQSeries CICS transaction MQMT to define the MQSeries resources. How to do this is described in “Defining MQSeries Resources under VSE” on page 95.

Defining MQSeries Resources under MVS

To define MQSeries resources under MVS:

1. Do one of the following:

- If you are using CICS/ESA, modify the member DSLKCSQC, which is shown in “DSLKCSQC (CICS only)” on page 168. In DSLKCSQC, the resources required for USE are:

DSL.MQI.RECEIVE

Receive queue (**e10**), which refers to the initiation queue (**e13**)

DSL.MQI.REPLY_TO_Q

Reply-to queue (**e17**), which refers to the initiation queue (**e13**)

DSL.MQI.CONTROL

Control queue (**e18**)

DSL.MQI.TRIGGER

Trigger process object containing the host receiving transaction ID (**g1**)

- If you are using IMS, modify the member DSLKCSQI, which is shown in “DSLKCSQI (IMS only)” on page 169. In DSLKCSQI, the resources required for USE are:

DSL.MQI.RECEIVE

Receive queue (**e10**), which refers to the initiation queue (**e13**)

DSL.MQI.REPLY_TO_Q

Reply-to queue (**e17**), which refers to the initiation queue (**e13**)

DSL.MQI.CONTROL

Control queue (**e18**)

DSL.MQI.TRIGGER.DSLR

Trigger process object containing the host receiving transaction ID (**h1**)

DSL.MQI.TRIGGER.DSLR1

Trigger process object containing the host receiving transaction ID (**h1**)

The comments in the members contain hints on how to modify the critical parameters.

2. Modify the member DSLKCSQU, which is shown in “DSLKCSQU” on page 170 . DSLKCSQU contains the commands that create the resources required for a connection to the remote MQSeries on Windows NT:

CSQ1.NTESAATT

Sender channel (e5)

NTESAATT.CSQ1

Receiver channel (e6)

DSL.MQI.SEND.USE

Send queue (e9)

DSL.MQI.INITQ.NTESAATT

Initiation queue (e13)

NTESAATT

Transmission queue (e15), which refers to the initiation queue (e13)

DSL.MQI.TRIGGER.PROCESS.NTESAATT

Trigger process object for transmission queue

Note that:

- The receive queue (e10) and reply-to queue (e17) use a common initiation queue.
- The transmission queue (e15) uses an initiation queue that is different from that used by the receive and reply-to queues.

DSLKCSQU defines the host send queue, which points to the remote receive queue, as shown in the following example:

```
*****
* Local Definition of the Remote Receive Queue *
*****
* If required, modify the following parameters:
* QREMOTE('DSL.MQI.SEND.USE')          <= Param. e9
* RNAME('RCV.FROM.CSQ1')              <= Param. e12
* RQMNAME('NTESAATT')                 <= Param. e2
* XMITQ('NTESAATT')                  <= Param. e15
DELETE QREMOTE('DSL.MQI.SEND.USE')
DEFINE QREMOTE('DSL.MQI.SEND.USE')      +
DESCR('Queue for Accessing RCV.FROM.CSQ1 on NTESAATT') +
RNAME('RCV.FROM.CSQ1')                 +
RQMNAME('NTESAATT')                   +
XMITQ('NTESAATT')                     +
PUT(ENABLED)                           +
DEFPSIST(YES)                           +
```

3. Run the MQSeries batch utility CSQUTIL once for each of the two copy books you are using. The following example shows sample JCL for running CSQUTIL.

```
//CSQUTIL JOB (Account),Name,...
//COMMAND EXEC PGM=CSQUTIL,PARM='CSQ1'          <= Param. e1
//STEPLIB DD DSN=MQSERIES.SCSQANLE,DISP=SHR
// DD DSN=MQSERIES.SCSQAUTH,DISP=SHR
//CSQUCMD DD DSN=MERVA.SDSLMAC0(XXX),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(CSQUCMD)
/*
```

- Under CICS/ESA, run it once with xxx replaced with DSLKCSQC, and once with xxx replaced with DSLKCSQU.
- Under IMS, run it once with xxx replaced with DSLKCSQI, and once with xxx replaced with DSLKCSQU.

Defining MQSeries Resources under VSE

For USE, you define four types of resources using the CICS transaction MQMT supplied by MQSeries:

- Local queue
- Remote queue
- Sender channel
- Receiver channel

Defining a Local Queue: You define the following local queues:

DSL.MQI.RECEIVE

Receive queue (e10)

DSL.MQI.REPLY_TO_Q

Reply-to queue (e17)

DSL.MQI.CONTROL

Control queue (e18)

DSL.MQI.CQ

Control queue (e18)

NTESAATT

Transmission queue (e15)

The steps to define the queue DSL.MQI.RECEIVE are:

1. Run the MQSeries master terminal transaction MQMT.

```
06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
15:52:30        *** Master Terminal Main Menu ***      CICS
MQMMTP                                     A001

                                SYSTEM IS ACTIVE

                                1. Configuration
                                2. Operations
                                3. Monitoring
                                4. Browse Queue Records

                                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
CLEAR/PF3 = Exit                            ENTER=Select
```

2. Select option 1 to configure.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:07:15       *** Configuration Main Menu ***             CICS
MQMMCFG                                               A001

                SYSTEM IS ACTIVE

                Maintenance Options :
                  1. Global System Definition
                  2. Queue Definitions
                  3. Channel Definitions

                Display Options   :
                  4. Global System Definition
                  5. Queue Definitions
                  6. Channel Definitions

                Option:

Please enter one of the options listed.
      5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
ENTER = Process          PF2 = Main Menu          PF3 = Quit

```

3. Select option 2 to work with queue definitions.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:12:43       Queue Main Options                       CICS
MQMMQUE                                               A001

                SYSTEM IS ACTIVE

                Default Q Manager : DVSCQMGR

                Object Type: L      L=Local Q, R=Remote Q, AQ=Alias Queue,
                                     AM=Alias Manager,
                                     AR=Alias Reply Q

                Object Name: DSL.MQI.RECEIVE

ENTER NEEDED INFORMATION.

PF2=Main Config PF3 = Quit PF4/ENTER = Read   PF5 = Add           PF6 = Update
                PF9 = List   PF11= Reorg.     PF12= Delete

```

4. Specify Object Type=L and the name of the queue.
5. Press PF5.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:21:55        Queue Definition Record                  CICS
MQMMQUE         QM - DVSCQMGR                             A001

                LOCAL QUEUE DEFINITION

Object Name. . . . . : DSL.MQI.RECEIVE
Description line 1 . . . . : RECEIVE QUEUE FROM NTESAATT
Description line 2 . . . . :

Put Enabled . . . . . : Y          Y=Yes, N=No
Get Enabled . . . . . : Y          Y=Yes, N=No

Default Inbound status . . : A      Outbound .. : A      A=Active,I=Inactive

Dual Update Queue . . . . . :

Automatic Reorganize (Y/N) : N

Record being added - Press ADD key again.

PF2=Main Config PF3 = Quit PF4/ENTER = Read PF5 = Add          PF6 = Update
                  PF9 = List PF10= Queue    PF11= Reorg.       PF12= Delete

```

6. Enter a descriptive text.
7. Press PF5 again.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:30:07        Queue Extended Definition                CICS
MQMMQUE         QM - DVSCQMGR                             A001
Object Name. . . . . : DSL.MQI.RECEIVE
                Physical Queue Information
Usage Mode . . . . . : N          N=Normal, T=Transmission
Share Mode . . . . . : Y          Y=Yes, N=No
Physical File Name . . . . : MQFI001  MQSERIES.MQFI001
                Maximum Values
Maximum Q Depth. . . . . : 01000000  Global Lock Entries . . : 00000500
Maximum Message Length . . : 00010240  Local Lock Entries. . . : 00000500
Maximum Concurrent Accesses: 00001000  Checkpoint Threshold : 1000

                Trigger Information
Trigger Enable . . . . . : Y          Y=yes, N=No
Trigger Type . . . . . : F          F=First, E=Every
Maximum Trigger Starts . . : 0001
Allow Restart of Trigger : Y          Y=Yes, N=No
Trans ID : DSLR                                     Term ID:
Program ID :                                         Channel Name:

***** File not found *****
PF2=Main Config PF3 = Quit PF4/ENTER = Read PF5 = Add          PF6 = Update
                  PF9 = List PF10= Queue    PF11= Reorg.       PF12= Delete

```

8. In 'Physical File Name', specify the name of a CICS file to store messages for this queue.
9. For queues to be triggered (DSL.MQI.RECEIVE and DSL.MQI.REPLY_TO_Q), specify the trigger information as shown, including the trans ID DSLR (parameter **g1**). For queues not to be triggered (DSL.MQI.CONTROL and DSL.MQI.CQ), omit the 'Trigger Information' or specify N where appropriate.
10. For the transmission queue NTESAATT, specify:
 - Usage Mode=T
 - Trigger Enable=Y
 - Trigger Type=F
 - Allow Restart of Trigger=Y
 - Program ID=MQPSEND

- Channel Name=DVSCQMGR.NTESAATT (this is the value of parameter e5)

For a normal queue, specify Usage Mode=N.

11. Press PF5 again.

Defining a Remote Queue: You define the remote queue DSL.MQI.SEND.USE (send queue, parameter e9). The necessary steps to define this queue are:

1. Run the MQSeries master terminal transaction MQMT.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
15:52:30       *** Master Terminal Main Menu ***          CICS
MQMMTP                                                A001

                SYSTEM IS ACTIVE

                1. Configuration
                2. Operations
                3. Monitoring
                4. Browse Queue Records

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
CLEAR/PF3 = Exit                          ENTER=Select

```

2. Select option 1 to configure.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:07:15       *** Configuration Main Menu ***          CICS
MQMMCFCG                                                A001

                SYSTEM IS ACTIVE

                Maintenance Options :
                1. Global System Definition
                2. Queue Definitions
                3. Channel Definitions

                Display Options      :
                4. Global System Definition
                5. Queue Definitions
                6. Channel Definitions

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
ENTER = Process      PF2 = Main Menu      PF3 = Quit

```

3. Select option 2 to work with queue definitions.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:12:43        Queue Main Options                          CICS
MQMMQUE                                                A001

                SYSTEM IS ACTIVE

Default Q Manager : DVSCQMGR

Object Type: R    L=Local Q, R=Remote Q, AQ=Alias Queue,
                  AM=Alias Manager,
                  AR=Alias Reply Q

Object Name: DSL.MQI.SEND.USE

ENTER NEEDED INFORMATION.

PF2=Main Config PF3 = Quit PF4/ENTER = Read   PF5 = Add           PF6 = Update
                  PF9 = List                   PF11= Reorg.       PF12= Delete

```

4. Specify Object Type=R and the name of the queue.
5. Press PF5.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
17:36:39        Queue Definition Record                          CICS
MQMMQUE         QM - DVSCQMGR                              A001

                REMOTE QUEUE DEFINITION

Object Name. . . . . : DSL.MQI.SEND.USE
Description line 1 . . . . : QUEUE FOR ACCESSING THE REMOTE
Description line 2 . . . . : RECEIVE QUEUE ON NTESAATT

Put Enabled . . . . . : Y           Y=Yes, N=No
Get Enabled . . . . . : N           Y=Yes, N=No

Remote Queue Name . . . . . : RCV.FROM.DVSCQMGR
Remote QM Name. . . . . : NTESAATT
Transmission Q Name . . . . . : NTESAATT

Record being added - Press ADD key again.

PF2=Main Config PF3 = Quit PF4/ENTER = Read   PF5 = Add           PF6 = Update
                  PF9 = List                   PF11= Reorg.       PF12= Delete

```

6. Enter a descriptive text, specify:
 - Get Enabled=N
 - Remote Queue Name=RCV.FROM.DVSCQMGR (parameter e12)
 - Remote QM Name=NTESAATT (parameter e2)
 - Transmission Q Name=NTESAATT (parameter e15)
7. Press PF5 again.

Defining a Sender Channel: You define the sender channel DVSCQMGR.NTESAATT (parameter e5). The steps to define this channel are:

1. Run the MQSeries master terminal transaction MQMT.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
15:52:30       *** Master Terminal Main Menu ***      CICS
MQMMTP                                                A001

                SYSTEM IS ACTIVE

                1. Configuration
                2. Operations
                3. Monitoring
                4. Browse Queue Records

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
CLEAR/PF3 = Exit                          ENTER=Select

```

2. Select option 1 to configure.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:07:15       *** Configuration Main Menu ***      CICS
MQMMCFCG                                              A001

                SYSTEM IS ACTIVE

                Maintenance Options :
                1. Global System Definition
                2. Queue Definitions
                3. Channel Definitions

                Display Options      :
                4. Global System Definition
                5. Queue Definitions
                6. Channel Definitions

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
ENTER = Process                          PF2 = Main Menu      PF3 = Quit

```

3. Select option 3 to work with channel definitions.


```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
18:00:42              Channel Record      DISPLAY      CICS
MQMMCHN      Last Check Point      Last Update 00000000      A001
MSN 00000000 Time 00:00:00 Interv 000000 Create Date 20010518
Name : DVSCQMGR.NTESAATT
Protocol : T (L/T) Port : 1414 Type : S (S/R/C)
Partner : 9.164.171.19

Allocation Retries      Get Retries
Number of Retries: 00000005      Number of Retries : 00000000
Delay Time - fast: 00000003      Delay Time      : 00000015
Delay Time - slow: 00000006

Max Messages per Batch : 000050      Max Transmission Size : 032000
Message Sequence Wrap : 999999      Max Message Size      : 0010240

Mess Seq Req(Y/N): Y      Convers Cap (Y/N): Y      Split Msg(Y/N): N

Transmission Queue Name : NTESAATT
TP Name:
Checkpoint Values:      Frequency: 0000      Time Span: 0000
Enable(Y/N) Y      Dead Letter Store(Y/N) Y
Channel record displayed.
PF2 =Menu PF3 =Quit PF4 =Read PF5 =Add PF6=Update PF9 =List PF12 =Delete

```

4. Complete the parameter fields as follows:

Name Specify the channel name DVSCQMGR.NTESAATT (parameter **e5**).

Protocol

Because TCP/IP is assumed, specify T.

Port The port number must match the port number configured for the remote host on Windows NT. The default port number for MQSeries for VSE/ESA is 1414.

Type Specify S (for sender channel).

Partner

Specify the IP address of the remote host (parameter **e4**).

Message Sequence Wrap

This number (parameter **e23**) must match the message sequence wrap configured for the remote host on Windows NT.

Transmission queue name

Specify the transmission queue name NTESAATT (parameter **e15**).

Enter values in the other fields as needed.

5. Press PF5.

Defining a Receiver Channel: You define the receiver channel NTESAATT.DVSCQMGR (parameter **e6**). The necessary steps to define this channel are:

1. Run the MQSeries master terminal transaction MQMT.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
15:52:30       *** Master Terminal Main Menu ***      CICS
MQMMTP                                                A001

                SYSTEM IS ACTIVE

                1. Configuration
                2. Operations
                3. Monitoring
                4. Browse Queue Records

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
CLEAR/PF3 = Exit                          ENTER=Select

```

2. Select option 1 to configure.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
16:07:15       *** Configuration Main Menu ***      CICS
MQMMCFCG                                              A001

                SYSTEM IS ACTIVE

                Maintenance Options :
                1. Global System Definition
                2. Queue Definitions
                3. Channel Definitions

                Display Options      :
                4. Global System Definition
                5. Queue Definitions
                6. Channel Definitions

                Option:

Please enter one of the options listed.
5686-A06 (C) Copyright IBM Corp. 1998      All Rights Reserved.
ENTER = Process                          PF2 = Main Menu      PF3 = Quit

```

3. Select option 3 to work with channel definitions.

```

06/28/2001      IBM MQSeries for VSE/ESA Version 2.1.0      DBDCCICS
18:41:29              Channel Record      DISPLAY      CICS
MQMMCHN      Last Check Point      Last Update 00000000      A001
MSN 00000000 Time 00:00:00 Interv 000000 Create Date 20010518
Name : NTESAATT.DVSCQMGR
Protocol : T (L/T) Port : 0000 Type : R (S/R/C)
Partner :

Allocation Retries      Get Retries
Number of Retries: 00000000      Number of Retries : 00000000
Delay Time - fast: 00000000      Delay Time      : 00000000
Delay Time - slow: 00000000

Max Messages per Batch : 000050      Max Transmission Size : 032000
Message Sequence Wrap : 999999      Max Message Size      : 0010240

Mess Seq Req(Y/N): Y      Convers Cap (Y/N): Y      Split Msg(Y/N): N

Transmission Queue Name :
TP Name:
Checkpoint Values:      Frequency: 0000      Time Span: 0000
Enable(Y/N) Y      Dead Letter Store(Y/N) Y
Channel record displayed.
PF2 =Menu PF3 =Quit PF4 =Read PF5 =Add PF6=Update PF9 =List PF12 =Delete

```

4. Specify the parameter fields as follows:

Name Specify the channel name NTESAATT.DVSCQMGR (e6).

Protocol

Because TCP/IP is assumed, specify T.

Type Specify R (for receiver channel).

Message Sequence Wrap

This number (e23) must match the message sequence wrap configured for the remote host on Windows NT.

Enter values in the other fields as needed.

5. Press PF5.

Customizing MERVA ESA Resources for the USE Functions

When customizing MERVA ESA resources, you work with the following parameters:

Name	Parameter ID	Comment
Host MQSeries Send Queue	e9	The default value might be acceptable.
Host MQSeries Receive Queue	e10	The default value might be acceptable.
Host Reply-to Queue	e17	The default value might be acceptable.
Host MQSeries Control Queue	e18	The default value might be acceptable.
Host User ID	e21	The default value might be acceptable.
Host MERVA Send Queue	f1	The default value might be acceptable.
Host MERVA Control Queue	f2	The default value might be acceptable.
Host MERVA COA Wait Queue	f5	The default value might be acceptable.

Name	Parameter ID	Comment
Host MERVA COD Wait Queue	f6	The default value might be acceptable.
Host MERVA Start Queue	f8	The default value might be acceptable.
Host Security Options	f9	The default value might be acceptable.
USE Workstation Name	f11	The default value might be acceptable.
Host Pregenerated Session Key Queue for GPA	f12	The default value might be acceptable.
Host Pregenerated Session Key Queue for FIN	f13	The default value might be acceptable.
Host User Exit	f14	The default value must be used unchanged.
Host Session Key Request Routing Table	f15	The default value might be acceptable.
Host Routing Table	f16	The default value might be acceptable.
Host Journaling Options	f17	The default value might be acceptable.
Host Master Logical Terminal Name	f18	You will definitely need to set the value for this parameter.

These parameters are described in Table 11 on page 86 and Table 12 on page 87.

The following MERVA ESA tables are affected:

- DSLFNNTT
- DSLKPROC
- DSLMPTT
- DWSLTT
- DWSRTSK
- EKARTTXU

Do the following:

1. Define the parameters of send and receive processes in your MERVA-MQI Attachment process table (DSLKPROC). The process table entries describe the USE application and the connection to the workstation. The source members DSLKPUSE (MVS) and DSLKPUSV (VSE), which are supplied with the MERVA ESA base product, contain sample process definitions.

Defaults are supplied for the following MQSeries queues:

- Host MQSeries Send Queue (e9)
- Host MQSeries Receive Queue (e10)
- Host Reply-to Queue (e17)
- Host MQSeries Control Queue (e18)

If you change these values, define the appropriate queues in MQSeries.

Defaults are supplied for the following MERVA ESA queues:

- Host MERVA Send Queue (f1)
- Host MERVA Control Queue (f2)
- Host MERVA COA Wait Queue (f5)

- Host MERVA COD Wait Queue (**f6**)
- Host MERVA Start Queue (**f8**)

If you change these values, change the function table as well.

For VSE, if you want incoming and outgoing messages to be encrypted and authenticated, specify AUTHENCR in the parameter **f9** (Host Security Options). For MVS, specify parameter **f9** as MSGEXIT('DSLKMEA') in the MQSeries sender and receiver channel definition. The channel message exit DSLKMEA must be copied to the appropriate data set of the MQSeries channel initiator. The CSQXLIB DD statement in the JCL of the started task for the MQSeries channel initiator points to this data set.

Parameter **f14** includes the exit number 8133 of the user exit program DSLKQ133, which passes the SWIFT acknowledgment from MERVA ESA to the MERVA USE & Branch on the workstation. This program and the related entries in the MFS program table DSLMPTT are available when MERVA ESA is installed.

The defaults specified for the Host Journaling Options (**f17**) result in the journaling of all incoming and outgoing messages. You can choose your own values.

The appropriate Host User ID (**e21**) must be defined on the workstation in group MQM. For MVS, a default alternate user ID is specified as the Host User ID in parameter ALTUID; you can modify this value as needed. If alternate user security is not active in your MQSeries installation, omit parameter ALTUID.

2. Assemble and link-edit your MERVA-MQI Attachment process table. The load module name must be DSLKPROC.

Samples for both MVS and VSE are shown in "Samples for Defining MERVA-MQI Attachment Resources" on page 173. For more information, refer to the *MERVA for ESA Macro Reference*.

3. Add the new functions (queues) in your function table (DSLFNNTT).

An example of a set of functions that is sufficient for USE is included in the copy book DWSFNNTTC, which is shipped with your MERVA ESA system.

4. Check that you have installed the routing tables DWSRTSK (**f15**) and EKARTTXU (**f16**) in your load library:

- A reference to parameter **f15** must be made in the logical terminal table (DWSLTT).
- A reference to parameter **f16** must be made in the function table copy book DWSFNNTTC.

5. Change the logical terminal table. First modify the name of the master logical terminals (**f18**) according to the names used in your installation. The modified master logical terminal names must also be used in the routing table EKARTTXU (**f16**). You should check, in addition, whether these names are used in the routing tables specified by the parameters ROUTIN and ROUTOUT.

There are parameters for the USE functions, for example, TFLAG, ROUTSK (**f15**), USERNAME (**f11**), and SKEYQ (**f12** and **f13**):

- Set TFLAG equal to SLS.
- Set parameter **f15** equal to DWSRTSK, for example, to determine the routing table for the SLS request.

- Specify a name for USERNAME **f11** (for example, USEMERVA2). Parameter **f11** is a routing condition (see “Routing Table for USE (DWSRTSK)” on page 179) to determine the send queue to route the LOGIN or SELECT request to. The send queue identifies the MERVA-MQI Attachment connection. This connection determines the workstation that provides the session keys from the assigned card reader.
 - Assign your session key queue name to the parameter SKEYQ (for example, **f12** = SLSGPA and **f13** = SLSFIN). This is also defined in the function table. The session key queue is filled with pregenerated session keys when a session key preload request is started from the workstation.
6. Assemble and link-edit the logical terminal table. The program name must be DWSLTT. If you specify a different name, it must be specified in the SWIFT Link customization parameter module DWSPRM using the DWSPARM macro. See Table 16 on page 122, the examples in “Logical Terminal Table for USE (DWSLTT)” on page 178, and the *MERVA for ESA Macro Reference*.

Customizing the Windows NT Based Resources

This section describes how to set up the workstation to run the MERVA USE & Branch, including how to define an MQSeries connection to a host system using MERVA-MQI Attachment.

Configuring Windows NT

In this part of the configuration, you work with the following parameter:

Name	Parameter ID	Comment
Host User ID	e21	For MVS, the alternate user ID (remote) is the default value, and might be acceptable. For VSE, you will definitely need to set the value for this parameter.

This parameter is described in Table 11 on page 86.

To configure your Windows NT system for MERVA-MQI Attachment:

1. Add the user (**e21**) using Windows NT User Manager
2. Add the group **mqm** to the user (**e21**)

Customizing MERVA USE & Branch Resources

In this part of the configuration, you work with the following parameters:

Name	Parameter ID	Comment
Workstation MERVA Send Queue	f3	The default value might be acceptable.
Workstation MERVA Receive Queue	f4	The default value might be acceptable.
Workstation MERVA Wait Queue	f7	The default value might be acceptable.

These parameters are described in Table 12 on page 87.

To customize your MERVA USE & Branch system for MERVA-MQI Attachment, do either of the following:

- Create and configure queues as described here:

1. Create the following message queues:
 - MERVA send queues (**f3**) within the MERVA API purpose group
 - MERVA receive queues (**f4**) within the MERVA API purpose group
 - MERVA wait queues (**f7**) within the MERVA API purpose group
2. Assign an alarm to each MERVA send queue.
3. Define the following routing conditions and default queues:
 - For the MERVA send queues:
 - Route the message to the MERVA wait queue if MSGOK="MVQWAIT"
 - Route the message to the final target queue if MSGOK<>"MVQWAIT"
 - For the MERVA receive queues:
 - Route the message to the final target queue if MSGOK<>"MVQWAIT"
 - Otherwise leave it in the receive queue (default queue)
 - For the MERVA wait queues:
 - Route the message to the final target queue if MSGOK<>"MVQWAIT"
 - Otherwise leave it in the wait queue (default queue)

Note: Ensure that all ready messages are routed to target queues other than those used by MERVA-MQI Attachment. This will reduce the amount of time needed to start MERVA-MQI Attachment, because the recovery processing that takes place during startup checks all MERVA-MQI Attachment queues for ready messages, and the fewer ready messages it finds, the more quickly it runs.

4. Define the MERVA user ID and password.
5. Define the MERVA user access rights to the queues defined in step 1.
- Import the standard routing for MERVA-MQI Attachment that is delivered with MERVA USE & Branch:
 1. Start MERVA in customization mode
 2. Change to the directory <ENM_PATH>\Samples\USE_MQI
 3. Enter


```
enmcximp -f enmcxmqa -c 1
```

This loads the file **enmcxmqa.rou**, which is located in the directory from step 2.

For more information about the import utility, refer to the chapter of the *MERVA USE & Branch for Windows NT Installation and Customization Guide* that describes how to export and import customization data.

For more information on how to customize MERVA, refer to the *MERVA USE & Branch for Windows NT Installation and Customization Guide*.

Customizing MQSeries Resources

In this part of the configuration, you work with the following parameters:

Name	Parameter ID	Comment
Host Queue Manager	e1	You will definitely need to set the value for this parameter. It is recommended that you set parameter e16 to the same value.
Workstation Queue Manager	e2	The default value might be acceptable.

Name	Parameter ID	Comment
Host Connection Name	e3	You will definitely need to set the value for this parameter.
Workstation Sender Channel	e7	The default value might be acceptable.
Workstation Receiver Channel	e8	The default value might be acceptable.
Host MQSeries Receive Queue	e10	The default value might be acceptable.
Workstation MQSeries Send Queue	e11	The default value might be acceptable.
Workstation MQSeries Receive Queue	e12	The default value might be acceptable.
Workstation Initiation Queue (Trigger Queue)	e14	The default value might be acceptable.
Workstation Transmission Queue	e16	It is recommended that you assign this parameter the same value as e1.
Workstation Reference Queue	e19	The default value might be acceptable.
Sequence Numbers Wrap	e23	The default value might be acceptable.
Workstation Security Options	f10	The default value might be acceptable.

These parameters are used in the configuration file to create and configure MQSeries channels and queues, and are described in Table 11 on page 86 and Table 12 on page 87.

Note: The sample MQSeries definitions shown in the following example apply to Version 5.0.1 of the IBM MQSeries for Windows NT. For more information about such definitions, refer to the *MQSeries MQSC Command Reference*.

To define the MQSeries connection:

1. Create, configure, and start an MQSeries queue manager by doing either of the following:
 - Open a Windows NT command prompt window and enter the following commands:


```
crtmqm -d DXQ -u DLQ NTESAATT
strmqm NTESAATT
```
 - Use the MQSeries Explorer:
 - a. Start the MQSeries Explorer (select **Start > Programs > IBM MQSeries > MQSeries Explorer**).
 - b. Select the queue manager item in the tree, and click the right mouse button.
 - c. Select **New > Queue Manager**. The Queue Manager creating window (Step 1) is displayed.
 - d. Enter the name of the queue manager (**e2**), for example NTESAATT.
 - e. Enter the names of the default transmission and dead-letter queues, for example DXQ and DLQ.

- f. Click the Next button. The Queue Manager creating window (Step 2) is displayed. Adapt the protocol path to your needs.
 - g. Click the Next button. The Queue Manager creating window (Step 3) is displayed. Make sure both the following checkboxes are selected:
 - Start queue manager
 - Channel for server connection
 - h. Click the Next button. The Queue Manager creating window (Step 4) is displayed.
 - i. If TCP/IP is the preferred network protocol, specify the port number for the TCP/IP listener, for example 1414.
 - j. Click the Finish button to create and start the queue manager.
2. Configure MQSeries channels and queues used by MERVA-MQI Attachment. You can do this manually, or you can use a configuration script file. If you use one of the configuration script files that are provided with MERVA USE & Branch, do the following:
- a. Open a Windows NT command prompt window.
 - b. Make a working directory in a drive of your choice, for example:
D:\MERVA-MQI Attachment
 - c. Change to the working directory.
 - d. Copy the configuration script file NTMVSATT.MQ (for MVS) or NTVSEATT.MQ (for VSE) from the directory <ENM_PATH>\Samples\USE_MQI to the working directory.
 - e. Modify the configuration script file:
 - Set the host queue manager (**e1**) and the host connection name (**e3**).
 - It is recommended that you set the name of the workstation transmission queue (**e16**) to the same name as the host queue manager (**e1**).
 - Set the queue names of the default transmission and dead-letter queues to the names you specified when you created the queue manager.
 - Modify other values if necessary.
 - f. Enter the following command:
runmqsc NTESAATT < NTxxxATT.MQ > NTxxxATT.OUT
- where:
- NTESAATT**
The name of the queue manager created in step 1
- xxx** The characters **MVS** or **VSE** (depending on the target operating system)
- NTxxxATT.MQ**
The name of the configuration script file you modified in 2e
- NTxxxATT.OUT**
The name of the file in which MQSeries is to record the result of the configuration script generation
- g. Check the results text file for errors.

For more information on how to configure MQSeries channels and queues used by MERVA-MQI Attachment, refer to the chapter of the *MERVA MQI Attachment User's Guide* that describes how to customize MERVA and MQSeries for use with MERVA-MQI Attachment.

Configuring MERVA-MQI Attachment

In this part of the configuration, you work with the following parameters:

Name	Parameter ID	Comment
Workstation Queue Manager	e2	The default value might be acceptable.
Workstation Reference Queue	e19	The default value might be acceptable.
Workstation Transaction Queue	e20	The default value might be acceptable.
Host User ID	e21	For MVS, the alternate user ID (remote) is the default value, and might be acceptable. For VSE, you will definitely need to set the value for this parameter.
Workstation User ID	e22	The default value might be acceptable.
Workstation MERVA Send Queue	f3	The default value might be acceptable.
Workstation MERVA Receive Queue	f4	The default value might be acceptable.
Workstation MERVA Wait Queue	f7	The default value might be acceptable.

These parameters are used in MERVA-MQI Attachment configuration file, and are described in Table 11 on page 86 and Table 12 on page 87.

To make a connection between MERVA USE & Branch and MQSeries, you must provide a MERVA-MQI Attachment configuration file and define the sections in the configuration file as needed:

- If you are using one of the configuration files NTMVSATT.ATN or NTVSEATT.ATN that are provided with MERVA USE & Branch, do the following:
 1. Switch to a command prompt window.
 2. Create a working directory in a drive of your choice, for example, D:\MQIA.
 3. Change to the working directory.
 4. Copy the configuration file NTMVSATT.ATN (for MVS) or NTVSEATT.ATN (for VSE) from the directory <ENM_PATH>\Samples\USE_MQI into the working directory.
 5. Modify the configuration file as required.
 6. Enter the following command to start MERVA-MQI Attachment:

```
CMEMQAT -f NTxxxATT.ATN
```

where NTxxxATT.ATN is the name of the configuration file you modified in the previous step.

- If you are not using NTMVSATT.ATN or NTVSEATT.ATN, or if you need more information about the configuration file and its use, refer to the chapter of the *MERVA MQI Attachment User's Guide* that describes how to customize MERVA-MQI Attachment.

Message Security

MERVA-MQI Attachment offers the following security services:

- Encrypting and creating an authentication checksum for messages to be sent
- Decrypting and authenticating received messages

It uses proprietary algorithms to encrypt, decrypt, and authenticate the message data. How you request these security services depends on your environment:

- For Windows NT, you activate an MQSeries channel exit.

To do this, specify the name of the exit in the MSGEXIT parameter of the DEFINE CHANNEL command:

```
MSGEXIT('c:\merva\USE_Branch\bin\cmenemqe.dll(MQAMsgExit)')
```

This corresponds to parameter **f10** (Workstation Security Options). The channel message exit has been installed during installation of MERVA ESA USE & Branch.

- For MVS, you activate an MQSeries channel exit.

To do this, specify the name of the exit in the MSGEXIT parameter of the DEFINE CHANNEL command:

```
MSGEXIT('DSLKMEA')
```

This corresponds to parameter **f9** (Host Security Options). Ensure that channel message exit DSLKMEA is installed in the non-authorized libraries defined by a CSQXLIB DD statement in the JCL of the started task for the MQSeries channel initiator.

- For VSE, you set a parameter in the DSLKPROC table.

To do this, specify parameter SECURE=AUTHENCR in each send and receive process. This corresponds to parameter **f9** (Host Security Options). Outgoing messages requesting a COA or COD report always request these reports without data.

You must either request the security services on both ends of the communication, or else omit them completely.

How authentication errors are handled depends on your environment:

- Under Windows NT, if the authentication of an incoming message fails, MERVA-MQI Attachment cannot process that message and puts it in the dead-letter queue. A dead-letter queue header is added to the message, and the Reason field in this header is filled with the reason code 13, which indicates an authentication error.
- Under MVS or VSE, if the authentication of an incoming message fails, MERVA-MQI Attachment assigns status AUTER to the message and routes it to an error queue. TOF field DSLKSTAT contains the message status AUTER. The sample routing table DSLKQRT illustrates the error handling. For USE, the routing tables EKARTTXU and DWSRTCT recognize a message with status AUTER as incorrect and route it to the error queue USEERROR. You can define your own error queue in the function table DSLFNNTT and use it as target queue in your routing table.

Parameter Relationships

The following parameter relationship tables show where parameters occur across all the involved systems: CICS, IMS, MQSeries, MERVA ESA, and MERVA USE & Branch. They describe in which subsystem each parameter is defined, where within the subsystem each parameter must be specified, and the default values supplied with the sample configuration. If you have completed the initial configuration and the connection does not work, you can use the tables to quickly locate and check the value of a parameter that you suspect is causing the problem.

The tables have the following columns:

Name	The parameter name, for example, e1 . Each parameter is described in "Parameter Definitions" on page 86.
Host	The host application in which the parameter occurs, for example, CICS.
Workstation	The workstation fields in which the parameter must be adapted.
Value in Sample	The initial value set in the sample code.
Your value	If you change a value for your system, write the new value in this column. Do not change the value of a parameter that has a value already printed in this column.

Parameters for a MERVA-MQI Attachment Connection—MQSeries

Table 15 shows the necessary parameters for the MQSeries based connection from MERVA ESA to MERVA USE & Branch.

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries

Name	Host	Workstation	Value in Sample	Your Value
e1	<u>MERVA ESA</u> In the process table (IMS and VSE only): DSLKPROC TYPE=INITIAL, MQMGR= e1 , e1 must already have been created.	In the MQSeries configuration file: DEFINE QREMOTE(e11) RNAME(e10) RQMNAME(e1) XMITQ(e16) It is recommended that you set parameters e1 and e16 to the same value.	CSQ1 (MVS) or DVSCQMGR (VSE)	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e2	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QREMOTE('e9') RNAME('e12') RQMNAME('e2') XMITQ('e15')</p> <p>It is recommended that you set parameters e2 and e15 to the same value.</p> <p><u>VSE</u></p> <p>On the MQMT screen REMOTE QUEUE DEFINITION: Object Name. . . . : e9 Remote Queue Name. : e12 Remote QM Name. . . : e2 Transmission Q Name: e15</p> <p>It is recommended that you set parameters e2 and e15 to the same value.</p>	<p>In the MERVA-MQI Attachment configuration file: [MVQAttachment] MQQMgrName = e2</p>	NTESAATT	
e3	<p>TCP/IP definition</p> <p><u>MVS</u></p> <p>In the utility CSQUTIL: START LISTENER TRPTYPE(TCP) PORT(1414)</p> <p><u>VSE</u></p> <p>On the MQMT screen Global System Definition: TCP/IP Listener Port : 01414</p>	<p>In the MQSeries configuration file: DEFINE CHANNEL(e7) CHLTYPE(SDR) CONNNAME('e3') TRPTYPE(TCP)</p> <p>The port number, which is part of e3, must match the port number configured on the host for listening for incoming TCP/IP requests from the workstation.</p>	1.123.123.0(1414)	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e4	<p>MVS</p> <p>In the utility CSQUTIL: DEFINE CHANNEL('e4') CHLTYPE(SDR) CONNAME('e4') TRPTYPE(TCP)</p> <p>The port number, which is part of e4, must match the port number configured on the workstation for listening for incoming TCP/IP requests from MVS.</p> <p>VSE</p> <p>On the MQMT screen Channel Record: Name : e5 Protocol: T Type : S Partner : e4 Port : 1414</p> <p>The port number, which is a separate field with the name Port, must match the port number configured on the workstation for listening for incoming TCP/IP requests from VSE.</p>	<p>TCP/IP definition</p> <p>In the MQSeries configuration file: STA LISTENER TRPTYPE(TCP) PORT(1414)</p>	9.164.171.19	
e5	<p>MVS</p> <p>In the utility CSQUTIL: DEFINE CHANNEL('e5') CHLTYPE(SDR)</p> <p>VSE</p> <p>On the MQMT screen Channel Record: Name : e5 Type : S</p>	Must match the parameter e8	CSQ1.NTESAATT (MVS) or DVSCQMGR.NTESAATT (VSE)	
e6	<p>MVS</p> <p>In the utility CSQUTIL: DEFINE CHANNEL('e6') CHLTYPE(RCVR)</p> <p>VSE</p> <p>On the MQMT screen Channel Record: Name : e6 Type : R</p>	Must match the parameter e7	NTESAATT.CSQ1 (MVS) or NTESAATT.DVSCQMGR (VSE)	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e7	Must match the parameter e6	In the MQSeries configuration file: DEFINE CHANNEL(e7) CHLTYPE(SDR) XMITQ(e16)	NTESAATT.CSQ1 (MVS) or NTESAATT.DVSCQMGR (VSE)	
e8	Must match the parameter e5	In the MQSeries configuration file: DEFINE CHANNEL(e8) CHLTYPE(RCVR) STA CHL(e8)	CSQ1.NTESAATT (MVS) or DVSCQMGR.NTESAATT (VSE)	
e9	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QREMOTE('e9') RNAME('e12')</p> <p><u>VSE</u></p> <p>On the MQMT screen REMOTE QUEUE DEFINITION: Object Name : e9 Remote Queue Name : e12</p> <p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, ALLSNDQ=((f1,e9)), ⋮</p>		DSL.MQI.SEND.USE	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e10	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QLOCAL('e10') INITQ('e13') PROCESS(...) TRIGGER TRIGTYPE(FIRST)</p> <p><u>VSE</u></p> <p>On the MQMT screen LOCAL QUEUE DEFINITION: Object Name. : e10</p> <p>On the MQMT screen Queue Extended Definition: Trigger Enable : Y Trigger Type : F Allow Restart of Trigger : Y</p> <p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=RECEIVE, MQIRCVQ=(e10,e17), :</p>	<p>In the MQSeries configuration file: DEFINE QREMOTE(e11) RNAME(e10) RQMNAME(e1) XMITQ(e16)</p>	DSL.MQI.RECEIVE	
e11		<p>In the MQSeries configuration file: DEFINE QREMOTE(e11) RNAME(e10) RQMNAME(e1) XMITQ(e16)</p> <p>In the MERVA-MQI Attachment configuration file: [MVQLink1] MQSendQName = e11</p>	SND.TO.CSQ1 (MVS) or SND.TO.DVSCQMGR (VSE)	
e12	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QREMOTE('e9') RNAME('e12')</p> <p><u>VSE</u></p> <p>On the MQMT screen REMOTE QUEUE DEFINITION: Object Name. : e9 Remote Queue Name : e12</p>	<p>In the MQSeries configuration file: DEFINE QLOCAL(e12)</p> <p>In the MERVA-MQI Attachment configuration file: [MVQLink1] MQRcvQName = e12</p>	RCV.FROM.CSQ1 (MVS) or RCV.FROM.DVSCQMGR (VSE)	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e13	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QLOCAL('e10') INITQ('e13') DEFINE QLOCAL('e17') INITQ('e13')</p> <p>Refer to copybook DSLKCSQU for an example of how to define an initiation queue: DEFINE QLOCAL ('DSL.MQI.INITQ.NTESAATT')...</p> <p><u>CICS</u></p> <p>In the DFHSIT table or, as SYSIN override, in the startup job: INITPARM=(CSQCPARM='SN=...,TN=001, IQ=e13')</p> <p><u>IMS</u></p> <p>In the JCL of the MQSeries-supplied trigger monitor, DD-name CSQQUT1 points to a data set in which e13 is named.</p>		CICS01.INITQ (CICS) or IMS01.INITQ.IMS61 (IMS)	
e14		In the MQSeries configuration file: DEFINE QLOCAL(e14)	TRIGQ	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e15	<p><u>MVS</u></p> <p>In the utility CSQUTIL:</p> <pre> DEFINE QLOCAL('e15') USAGE(XMITQ) INITQ(...) PROCESS(...) TRIGGER TRIGTYPE(FIRST) DEFINE QREMOTE('e9') RQMNAME('e2') XMITQ('e15') DEFINE CHANNEL('e5') CHLTYPE(SDR) XMITQ('e15') </pre> <p>It is recommended that you set parameters e2 and e15 to the same value.</p> <p><u>VSE</u></p> <p>On the MQMT screen LOCAL QUEUE DEFINITION:</p> <p>Object Name : e15</p> <p>On the MQMT screen Queue Extended Definition:</p> <pre> Usage Mode : T Trigger Enable : Y Trigger Type : F </pre> <p>On the MQMT screen REMOTE QUEUE DEFINITION:</p> <pre> Object Name : e9 Remote QM Name : e2 Transmission Q Name : e15 </pre> <p>It is recommended that you set parameters e2 and e15 to the same value.</p> <p>On the MQMT screen Channel Record:</p> <pre> Name : e5 Type : S Transmission Queue Name : e15 </pre>		NTESAATT	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e16		<p>In the MQSeries configuration file:</p> <pre> DEFINE QLOCAL (e16) USAGE (XMITQ) DEFINE QREMOTE (e11) RNAME (e10) RQMNAME (e1) XMITQ (e16) DEFINE CHANNEL (e7) CHLTYPE (SDR) XMITQ (e16) </pre> <p>It is recommended that you set parameters e1 and e16 to the same value.</p>	<p>CSQI (MVS) or DVSCQMGR (VSE)</p>	
e17	<p><u>MVS</u></p> <p>In the utility CSQUTIL:</p> <pre> DEFINE QLOCAL ('e17') INITQ ('e13') PROCESS (...) TRIGGER TRIGTYPE (FIRST) </pre> <p><u>VSE</u></p> <p>On the MQMT screen LOCAL QUEUE DEFINITION: Object Name. : e17</p> <p>On the MQMT screen Queue Extended Definition: Trigger Enable : Y Trigger Type : F Allow Restart of Trigger : Y</p> <p><u>MERVA ESA</u></p> <p>In the process table:</p> <pre> DSLKPROC TYPE=SEND, REPLYTQ=e17, : : DSLKPROC TYPE=RECEIVE, MQIRCQ=(e10,e17), : : </pre>		<p>DSL.MQI.REPLY_TO_Q</p>	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e18	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE QLOCAL('e18')</p> <p><u>VSE</u></p> <p>On the MQMT screen LOCAL QUEUE DEFINITION: Object Name. : e18</p> <p>Note: Under VSE, each MERVA-MQI Attachment send process must have a unique control queue. Therefore, two control queues must be specified.</p> <p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, MQICTLQ=e18, :</p>		DSL.MQI.CONTROL (MVS and VSE) and DSL.MQI.CQ (VSE only)	
e19		<p>In the MQSeries configuration file: DEFINE QLOCAL(e19)</p> <p>In the MERVA-MQI Attachment configuration file: [MVQAttachment] MQRefQName = e19</p>	REFQ	
e20		<p>In the MERVA-MQI Attachment configuration file: [MVQAttachment] # MQTactQName = e20</p>	TACTQ	
e21	<p><u>MQSeries</u></p> <p>e21 requires authorization to utilize (open, get, put, ...) an MQSeries resource.</p> <p><u>MERVA ESA</u></p> <p>In the process table (MVS only): DSLKPROC TYPE=SEND or RECEIVE, ALTUID=e21, :</p>	<p>In the MERVA-MQI Attachment configuration file: [MVQAttachment] MVQName = e21 [MVQLink1] RmtUser = e21</p> <p>e21 must be defined in the group mqm.</p>	remote (MVS) or CICS user ID associated to the transactions g1 and g3 (VSE)	
e22		<p>In the MQSeries configuration file: [MVQAttachment] MRVUser = e22</p>	merva1	

Table 15. Parameters for a MERVA-MQI Attachment Connection—MQSeries (continued)

Name	Host	Workstation	Value in Sample	Your Value
e23	<p><u>MVS</u></p> <p>In the utility CSQUTIL:</p> <pre>DEFINE CHANNEL('e5') CHLTYPE(SDR) SEQWRAP(e23) DEFINE CHANNEL('e6') CHLTYPE(RCVR) SEQWRAP(e23)</pre> <p><u>VSE</u></p> <p>On the MQMT screen Channel Record:</p> <p>Name : e5 Type : S Message Sequence Wrap : e23</p> <p>and</p> <p>Name : e6 Type : R Message Sequence Wrap : e23</p>	<p>In the MQSeries configuration file:</p> <pre>DEFINE CHANNEL(e7) CHLTYPE(SDR) SEQWRAP(e23) DEFINE CHANNEL(e8) CHLTYPE(RCVR) SEQWRAP(e23)</pre>	<p>999999999 (MVS) or 9999999 (VSE)</p>	

Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS

Table 16 shows the parameters for the MERVA-MQI Attachment connection from MERVA ESA to MERVA USE & Branch that are independent of the host subsystem used (CICS or IMS).

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS

Name	Host	Workstation	Value in Sample	Your Value
f1	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, ALLSNDQ=((f1,e9)), ⋮</p> <p>In the function table DWSFNTTC: DSLFNT NAME=f1, ⋮</p> <p>DSLFNT NAME=BKEPRE, NEXT=f1, ⋮</p> <p>In the following routing tables:</p> <ul style="list-style-type: none"> • DWSRTSK (corresponds to f15) • DWSL1IN • DWSL1OUT • DWSL2IN • DWSL2OUT • DWSL3GPI • DWSL3GPO • DWSL3FII • DWSL3FIO <p>specify: DSLROUTE TYPE=SET, TARGET=('f1'),</p> <p>Note: For the initial values set for f1 in the sample code:</p> <ul style="list-style-type: none"> • USESQ is not specified in the following routing tables: <ul style="list-style-type: none"> - DWSL1IN - DWSL2IN - DWSL3GPI - DWSL3FII • USESQACK is not specified in the following: <ul style="list-style-type: none"> - Parameter NEXT of DSLFNT - The following routing tables: <ul style="list-style-type: none"> - DWSRTSK - DWSL1OUT - DWSL2OUT - DWSL3GPO - DWSL3FIO 		<p>USESQ and USESQACK</p>	

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS (continued)

Name	Host	Workstation	Value in Sample	Your Value
f2	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND or RECEIVE, MRVCTLQ=f2, ⋮</p> <p>In the function table DWSFNTPC: DSLFNT NAME=f2, ROUTE=f16, ⋮</p>		USECQS or USECQR	
f3		In the MERVA-MQI Attachment configuration file: [MVQLink1] MVSendQName = f3	MQSND1 and MQSND2	
f4		In the MERVA-MQI Attachment configuration file: [MVQLink1] MVRcvQName = f4	MQRCV	
f5	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, COAWQ=f5, ⋮</p> <p>In the function table DSLFNTPC: DSLFNT NAME=f5, ⋮</p>		DSLMRCOA	
f6	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, CODWQ=f6, ⋮</p> <p>In the function table DSLFNTPC: DSLFNT NAME=f6, ⋮</p>		DSLMRCOD	
f7		In the MQSeries configuration file: [MVQLink1] MVWaitQName = f7	MQWAIT	

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS (continued)

Name	Host	Workstation	Value in Sample	Your Value
f8	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND or RECEIVE, MRVSTAQ=f8, ⋮</p> <p>In the function table DSLFNTTC: DSLFNT NAME=f8, TRAN=g1, (CICS receive) or TRAN=g3, (CICS send) or TRAN=h1, (IMS receive) or TRAN=h3, (IMS send) QUEUE=DUMMY, ⋮</p>		DSLMRSTS or DSLMRSTR	
f9	<p><u>MVS</u></p> <p>In the utility CSQUTIL: DEFINE CHANNEL('e5') CHLTYPE(SDR) MSGEXIT('f9') DEFINE CHANNEL('e6') CHLTYPE(RCVR) MSGEXIT('f9')</p> <p>In the JCL of the MQSeries channel initiator, the DD-name CSQXLIB points to a data set that contains channel message exit f9.</p> <p><u>MERVA ESA</u></p> <p>In the process table (VSE only): DSLKPROC TYPE=SEND or RECEIVE, SECURE=f9, ⋮</p>	If f9 has been specified, f10 is required.	DSLKMEA (MVS) or AUTHENCR (VSE)	
f10	If f10 has been specified, f9 is required.	In the MQSeries configuration file: DEFINE CHANNEL(e7) CHLTYPE(SDR) MSGEXIT('f10') DEFINE CHANNEL(e8) CHLTYPE(RCVR) MSGEXIT('f10')	c:\merva\USE_Branch\bin\cmenemqe.dll (MQAMsgExit)	

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS (continued)

Name	Host	Workstation	Value in Sample	Your Value
f11	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, USENAME=f11, : :</p> <p>In the routing table DWSRTSK (corresponds to f15) DSLROUTE TYPE=TEST, COND=(...,f11,...), : :</p>		USEMERVA2	
f12	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, TFLAG=SLS, SKEYQ=f12, : :</p> <p>In the function table DWSFNTTC: DSL FNT NAME=f12, : :</p>		SLSGPA	
f13	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=AP, NAME=FIN, SKEYQ=f13, : :</p> <p>In the function table DWSFNTTC: DSL FNT NAME=f13, : :</p>		SLSFIN	

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS (continued)

Name	Host	Workstation	Value in Sample	Your Value
f14	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, EXIT=f14, ⋮</p> <p>In the MFS program table DSLMPTTC: DSLMPT NAME=DSLKQ133, NUMBER=f14, LINK=NO, ⋮</p> <p><u>CICS (optional)</u></p> <p>Because DSLMPT LINK=NO is specified, DSLKQ133 must be defined to CICS. Specify one of the following:</p> <ul style="list-style-type: none"> • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=DSLKQ133, ⋮ • In the offline utility: DEFINE PROGRAM(DSLKQ133), ⋮ 		8133	8133
f15	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, ROUTSK=f15, ⋮</p> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=f15, ⋮ • In the offline utility: DEFINE PROGRAM(f15), ⋮ 		DWSRTSK	

Table 16. Parameters for a MERVA-MQI Attachment Connection Independent of CICS or IMS (continued)

Name	Host	Workstation	Value in Sample	Your Value
f16	<p><u>MERVA ESA</u></p> <p>In the function table DWSFNTTC: DSL FNT NAME=f2, ROUTE=f16, : :</p> <p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=f16, : : • In the offline utility: DEFINE PROGRAM(f16), : : 		EKARTTXU	
f17	<p><u>MERVA ESA</u></p> <p>In the process table: DSLKPROC TYPE=SEND, JRNDGRM=f17, or JRNRPLY=f17, : :</p> <p>DSLKPROC TYPE=RECEIVE, JRNDGRM=f17, JRNRCOA=f17, JRNRCOD=f17, JRNRQST=f17, : :</p>		YES	
f18	<p><u>MERVA ESA</u></p> <p>In the logical terminal table: DWSLT TYPE=MAS, NAME=f18, : :</p> <p>In the following routing tables:</p> <ul style="list-style-type: none"> • EKARTTXU (corresponds to f16) • DWSL2OUT • DWSL3GPO <p>specify: DSLROUTE TYPE=TEST, COND=(..., f18, ...),</p>		VNDEBET0A VNDOBET0A VNDPBET0A	

Parameters for a MERVA-MQI Attachment Connection—CICS to Workstation

Table 17 shows the necessary parameters for the connection from MERVA ESA under CICS to MERVA USE & Branch.

Table 17. Parameters for a MERVA-MQI Attachment Connection—CICS to Workstation

Name	Host	Workstation	Value in Sample	Your Value
g1	<p>CICS</p> <p>One of the following:</p> <ul style="list-style-type: none"> In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=g1, ⋮ In the offline utility: DEFINE TRANSACTION(g1) <p>MQSeries</p> <p>One of the following:</p> <ul style="list-style-type: none"> For MVS, in the utility CSQUTIL: DEFINE PROCESS(...) APPLICID(g1) APPLTYPE(CICS) For VSE, on the MQMT screen Queue Extended Definition: Trans ID : g1 <p>MERVA ESA</p> <p>In the function table DSLFNTTC: DSLFNT NAME=f8, TRAN=g1, QUEUE=DUMMY, ⋮</p>		DSLRL	

Table 17. Parameters for a MERVA-MQI Attachment Connection—CICS to Workstation (continued)

Name	Host	Workstation	Value in Sample	Your Value
g2	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=g1, PROGRAM=g2, ⋮ • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=g2, ⋮ • In the offline utility: DEFINE TRANSACTION(g1) PROGRAM(g2) ⋮ DEFINE PROGRAM(g2) ⋮ 		DSLKQR	DSLKQR
g3	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=g3, ⋮ • In the offline utility: DEFINE TRANSACTION(g3) <p><u>MERVA ESA</u></p> <p>In the function table DWSFNNTC: DSLFNT NAME=f1, TRAN=g3, ⋮</p> <p>In the function table DSLFNNTC: DSLFNT NAME=f8, TRAN=g3, QUEUE=DUMMY, ⋮</p>		DSL	

Table 17. Parameters for a MERVA-MQI Attachment Connection—CICS to Workstation (continued)

Name	Host	Workstation	Value in Sample	Your Value
g4	<p><u>CICS</u></p> <p>One of the following:</p> <ul style="list-style-type: none"> • In the transaction definition DFHPCTxx: DFHPCT TYPE=ENTRY, TRANSID=g3, PROGRAM=g4, ⋮ • In the program definition DFHPPTxx: DFHPPT TYPE=ENTRY, PROGRAM=g4, ⋮ • In the offline utility: DEFINE TRANSACTION(g3) PROGRAM(g4) ⋮ <p>DEFINE PROGRAM(g4) ⋮</p>		DSLKQS	DSLKQS

Parameters for a MERVA-MQI Attachment Connection—IMS to Workstation

Table 18 shows the necessary parameters for the connection from MERVA ESA under IMS to MERVA USE & Branch.

Table 18. Parameters for a MERVA-MQI Attachment Connection—IMS to Workstation

Name	Host (MVS)	Workstation	Value in Sample	Your Value
h1	<p>IMS</p> <p>In the transaction definition: APPLCTN PSB=h2, ⋮</p> <p>TRANSACT CODE=h1,</p> <p>MQSeries</p> <p>In the utility CSQUTIL: DEFINE PROCESS(...) APPLICID(h1) APPLTYPE(IMS)</p> <p>MERVA ESA</p> <p>In the function table DSLFNTTC: DSLFNT NAME=f8, TRAN=h1, QUEUE=DUMMY, ⋮</p>		DSLRL and DSLRL1	
h2	<p>IMS</p> <p>In the transaction definition: APPLCTN PSB=h2, ⋮</p>		DSLKQR	DSLKQR
h3	<p>IMS</p> <p>In the transaction definition: APPLCTN PSB=h4, ⋮</p> <p>TRANSACT CODE=h3,</p> <p>MERVA ESA</p> <p>In the function table DWSFNTTC: DSLFNT NAME=f1, TRAN=h3, ⋮</p> <p>In the function table DSLFNTTC: DSLFNT NAME=f8, TRAN=h3, QUEUE=DUMMY, ⋮</p>		DSLRL	

| Table 18. Parameters for a MERVA-MQI Attachment Connection—IMS to Workstation (continued)

Name	Host (MVS)	Workstation	Value in Sample	Your Value
h4	<u>IMS</u> In the transaction definition: APPLCTN PSB=h4, ⋮		DSLKQS	DSLKQS

Chapter 6. Administering the USE Functions (MERVA-MQI Attachment)

This chapter describes step-by-step procedures you can use to verify that the USE functions have been correctly installed and customized, and describes the message routing for the USE functions.

Verifying the USE Functions

This section describes a step-by-step procedure you can use to verify that the workstation based USE functions have been installed and customized correctly.

The MERVA ESA related host part of the USE functions is installed automatically when you install MERVA ESA. You define and install the MQSeries related host part of the USE functions yourself. See “Chapter 5. Connecting MERVA ESA to the USE Functions via MERVA-MQI Attachment” on page 85 for details of how to customize the host part.

Refer to the *MERVA USE & Branch for Windows NT Installation and Customization Guide* for details of the installation and customization procedure for the workstation part. Carry out the following procedures after you have completed installation and customization of the USE functions.

Verifying the MERVA-MQI Attachment Connection and the SLS Services

If you are an experienced user of MQSeries and the MERVA-MQI Attachment, skip step 1. Beginning with step 2, you can verify that the MERVA-MQI Attachment connection and the SLS service are working correctly. If you are unfamiliar with MQSeries and MERVA-MQI Attachment on the host, proceed as follows:

1. Verify the installation of the MERVA-MQI Attachment on the host. This installation verification demonstrates the essential functions of the MERVA-MQI Attachment locally; that is, without using MQSeries channels. See the detailed procedure in the chapter on verifying the installation of MERVA-MQI Attachment in the *MERVA for ESA Installation Guide*.
2. If running under MVS, copy the channel message exit DSLKMEA to the appropriate data set of the MQSeries channel initiator. The CSQXLIB DD statement in the JCL of the started task for the MQSeries channel initiator points to this data set.
3. Activate the process table DSLKPROC on the host:
 - For MVS, execute the SMP/E USERMOD job DSLKPEYC, which is part of the installation library SDSLINS0.
 - For VSE, assemble the source member DSLKPUSV, then run the linkbook LSLKPUSV. Both the source member and the linkbook are parts of the sublibrary MERVA.LIBS.
4. If running under:
 - MVS, ensure that a connection to MQSeries exists. How to do this is described in the chapter on verifying and starting a connection to MQSeries in the *MERVA for ESA Installation Guide*.

- VSE, ensure that MQSeries is operable. To do this, you must be authorized to use the transaction code MQSE. After CICS is started, enter **MQSE I**. If MQSeries is operable, the response will be:

```
MQSE: MQSeries environment setup and initialized
```

5. At the workstation, ensure that the host user ID (for example 'remote' or the CICS User ID) exists as a Windows NT user and that the group MQM is added to that user.
6. Ensure that MERVA ESA is running.
7. Sign on to MERVA ESA. The Function Selection panel is displayed.
8. Select the function **CMD**. The appropriate panel is displayed.
9. At the workstation:
 - a. Start MERVA USE & Branch in multi user mode.
 - b. Start MQSeries and the queue manager defined for MERVA-MQI Attachment.
 - c. Start MERVA-MQI Attachment with the appropriate configuration file (for example, enter **cmemqat -f ntmvsatt.atn**).
10. Ensure that the SCR or BCR is ready, the card reader is connected to the workstation, a USER ICC for the destination is inserted into the card reader, and the appropriate PIN for the card is entered.
11. In the MERVA ESA panel, enter **login** in the command line.
12. If the **login** command is accepted, enter the **dl** command. In its response, in the column STATUS, "SK PRC" (session key proceeding) is displayed.
13. After some time, enter the **dl** command again. In the response, the status should have changed to "LIN PRC" (login proceeding).
14. Enter the **dm last** command. The response should, for a switched line, show the messages:

```
DWS602I ltname LOGIN proceeding on line 1
:
```

If the message DWS602I is displayed, the request for an SLS session key was successfully implemented by the card reader and verification is successful.

Data Flow for the USE Functions

This section describes the data flow for the workstation based USE functions. Sample definitions for all MERVA ESA queues mentioned in this section are included in the copy book DWSFNNTC. The copy book DWSFNNTC is contained in the MERVA ESA macro library.

All MERVA ESA routing tables mentioned in this section are contained in the MERVA ESA source library.

Data Flow of Pregenerated Session Keys for SLS

With the Secure Login and Select (SLS) service, session keys are "pregenerated" on the USE workstation.

Figure 23 on page 136 shows the data flow for pregenerated session keys.

When pregeneration is complete, you send the session keys from the USE workstation to MERVA ESA with MERVA-MQI Attachment. To do this, an MT 999 with a special format is created, containing up to 34 session keys. Sufficient MT 999 messages are created to send all the session keys.

| In the receiving MERVA-MQI Attachment of MERVA ESA, the routing table
| EKARTTXU, defined for the control queue USECQR, identifies the MT 999
| messages and routes them to the session key load queue SLSLOAD.

| From the SLSLOAD queue, the session key load program DWSDLSK receives the
| MT 999 messages and verifies that the logical terminal (LT) name in the basic
| header is defined in DWSLTT and that session key queues are defined for this LT
| or its FIN application.

| If so, the session keys are stored in the appropriate session key queue, for example,
| in SLSFIN or SLSGPA.

| During LOGIN or SELECT requests, the session keys are retrieved from the session
| key queues by the program DWSDGPA.

| Refer to the *MERVA for ESA Operations Guide* for more detailed information about
| preloaded session keys and how to maintain them.

Data Flow of Session Keys Requests for SLS

| If preloaded session keys are not used, the SWIFT Link generates a special MT 999
| for each LOGIN and SELECT request and sends it to MERVA USE & Branch to
| request the session key needed for the LOGIN or SELECT. Figure 23 on page 136
| contains an overview of the flow of session key request messages for SLS.
|

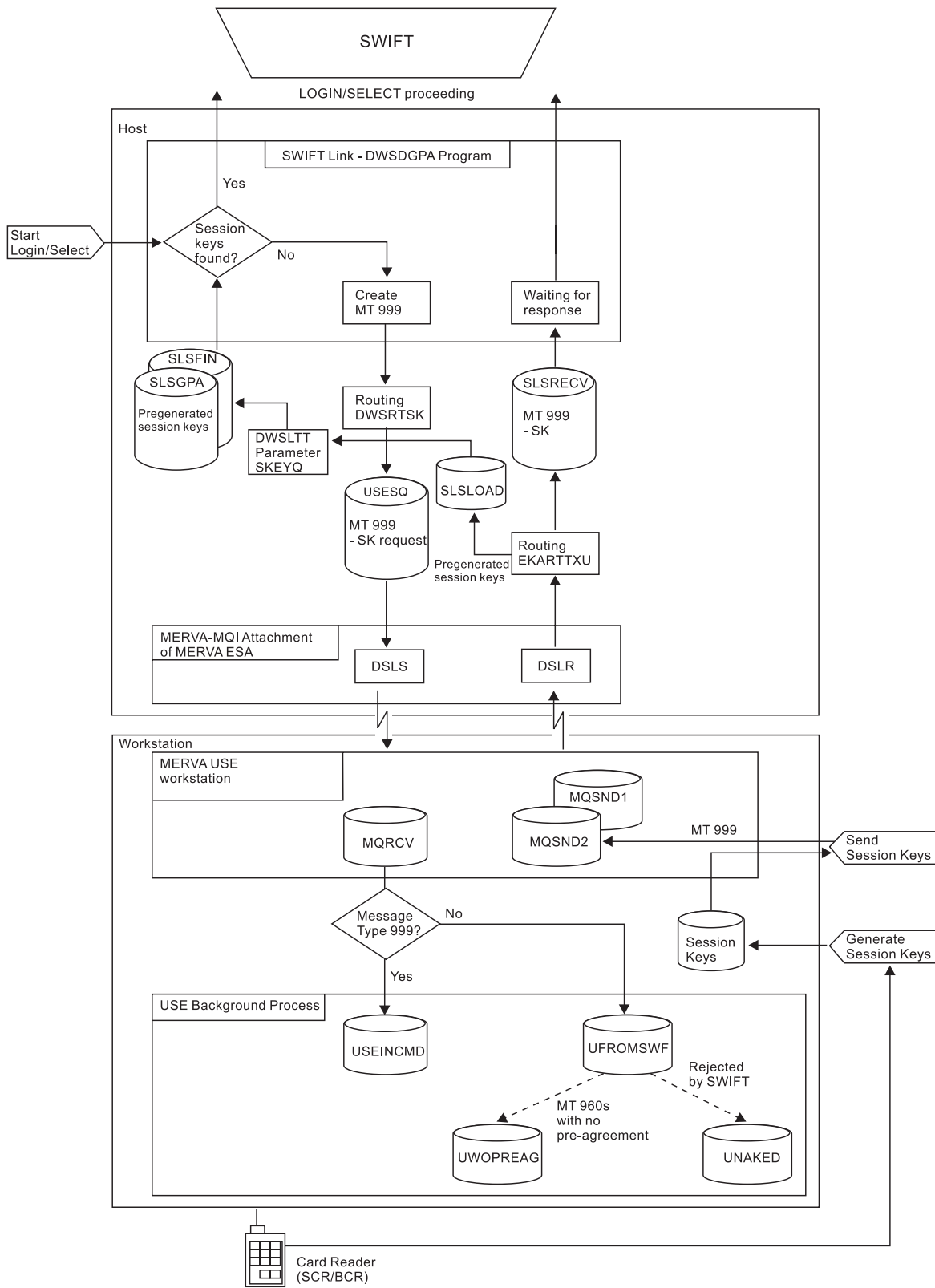


Figure 23. Flow of Session Key Request Messages for SLS and Pregenerated Session Keys

After the LOGIN or SELECT request has been accepted by the SWIFT Link and the command response indicates that the "session key pending" status has been set, the main program of the SWIFT Link, DWSGDPA, proceeds as follows:

1. If there is a session key queue defined for this master LT or its FIN application, DWSDGPA attempts to get the session key for the current LSN or SSN and the current ICC parameters from this queue.
If the session key is found, the login or select can proceed.
2. If there is no session key queue defined, or the session key could not be found in the queue, a special MT 999 message for a session key request is generated.
In this MT 999, the LT name is written to the basic header and the value of the USERNAME parameter of the DWSLT macro written to the application header as the receiving LT name.

When a special MT 999 message is generated, it is routed using the routing table DWSRTSK, which is defined with the ROUTSK parameter of the DWSLT macro. This routing table determines the name of the MERVA-MQI Attachment send queue USESQ for sending the session key request message to MERVA USE & Branch.

If more than one workstation is used for SLS, DWSRTSK evaluates the LT name of the application header for routing to more than one MERVA-MQI Attachment send queue.

When MERVA USE & Branch receives the session key request message, it either uses pregenerated session keys if any are available, or requests the session key from the card reader. If the session key is found, it is written into the session key request message. Otherwise, appropriate error information is added.

The session key request message is then routed to the MERVA-MQI Attachment send queue MQSND2, from where MERVA-MQI Attachment sends it to MERVA ESA.

In the receiving MERVA-MQI Attachment in MERVA ESA, the routing table EKARTTXU (defined for the control queue USECQR) identifies the MT 999 messages as being session key requests and routes them to the session key receive queue SLSRECV. This queue is defined both in DWSPRM (RSKQUE parameter of the DWSPARM macro) and in DSLFNTT.

From this queue, the program DWSDGPA gets the MT 999 messages and verifies in DWSLTT that this LT or FIN application is authorized to use this session key. If so, it is stored in the appropriate entry of DWSLTT.

If this LT or FIN application is also waiting for this session key (session key proceeding "SK PRC" status), login or select can proceed.

Data Flow of USE Messages

Both, USE system messages and BKE messages, have the following types:

1. Input messages. These messages are generated by MERVA USE & Branch and sent to MERVA ESA with MERVA-MQI Attachment for sending to the SWIFT network. The USE system messages for input are the message types MT 074, MT 075, MT 085, and MT 090. The BKE messages for input are the message types MT 960 to MT 967 (the same as output messages).
2. Output messages. These messages are received from the SWIFT network and sent to MERVA USE & Branch with MERVA-MQI Attachment. The USE system messages for output are the message types MT 076, MT 087, and MT 092. The BKE messages for output are the message types MT 960 to MT 967 (the same as input messages).

Figure 24 on page 140 gives an overview of the flow of USE system messages and BKE messages.

Flow of Input Messages

USE and BKE input messages are generated in MERVA USE & Branch, either by a USE administration function or by the BKE process. After generation, the messages with message type MT 999 are routed to the MERVA-MQI Attachment send queue MQSND2, all other messages are routed to the MERVA-MQI Attachment send queue MQSND1. From there they are sent to MERVA ESA over the MERVA-MQI Attachment.

In the receiving MERVA-MQI Attachment of MERVA ESA, the routing table EKARTTXU defined for the control queue USECQR recognizes the USE and BKE input messages and routes them to the appropriate ready queue for sending to the SWIFT network.

The program DWSDGPA eventually sends them to the SWIFT network and receives an acknowledgment. Sending, however, may not be possible due to an error.

In either case, the routing table (DWSL1IN, DWSL2IN, DWSL3GPI, or DWSL3FII) recognizes the USE and BKE input messages and, in addition to routing the messages to acknowledgment (L1ACK) or correction queues (L1VE0), routes the messages to the MERVA-MQI Attachment send queue USESQACK for sending back to MERVA USE & Branch. Acknowledgment or error information is returned in the MSGACK field of the messages.

When the messages are received in MERVA USE & Branch, status information in the USE or BKE processes is updated accordingly.

Flow of Output Messages

On receiving a USE or BKE output message, the SWIFT Link routing table (DWSL1OUT, DWSL2OUT, DWSL3GPO, or DWSL3FIO) identifies the USE or BKE output message and, in addition to the usual routing to display output (L1DO0) or sequential data set output queues (L1SDO), routes the messages to the MERVA-MQI Attachment send queue USESQ for sending to MERVA USE & Branch.

When the messages are received in MERVA USE & Branch, status information in the USE or BKE processes is updated accordingly and a BKE process continued by generating the necessary responses.

On completion of bilateral key exchange, the MERVA USE & Branch authenticator-key file is updated and an update message (a special MT 999) generated and sent to MERVA ESA with MERVA-MQI Attachment.

In MERVA ESA, the receiving MERVA-MQI Attachment routing table EKARTTXU routes this special MT 999 to the queue BKEUPDT. From this queue, the transaction program DWSAUTT receives the message and invokes DWSAUTP to update the MERVA ESA authenticator-key file.

Note: If you want to maintain manual keys in the authenticator-key file, you can specify the parameter AUTHMAN=MERVAESA in the DWSPARM macro. In this case, manual keys are not updated by BKE keys and records with manual keys cannot be deleted.

| But if the FROM date of a new BKE key overlaps the TO date of an existing
| manual key, the TO date of the manual key will be replaced by the FROM date of
| the BKE key. So you can start the BKE processing at any time without regarding
| the TO dates of the manual keys.
|

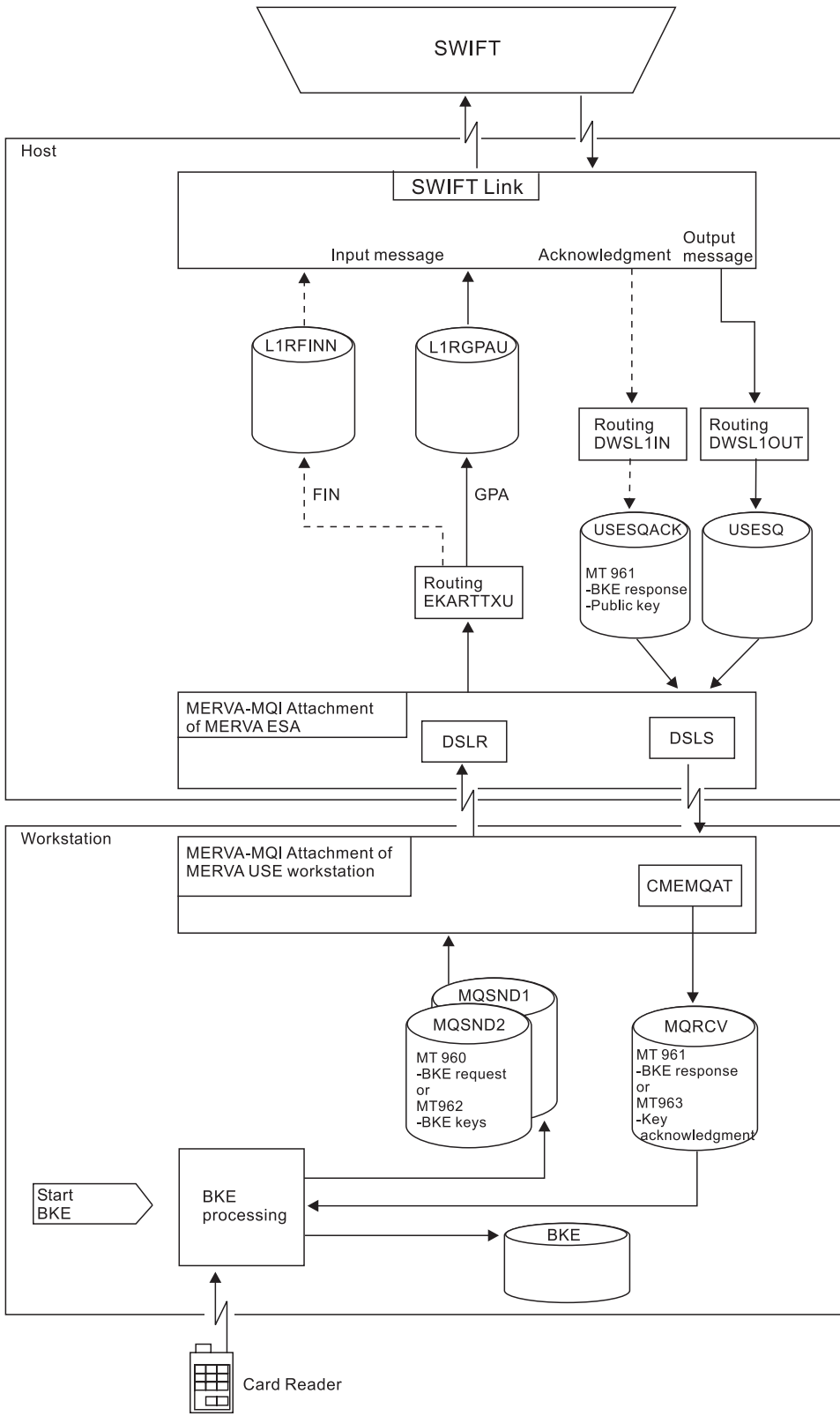


Figure 24. Message Flow for BKE Started from MERVA

Workstation Message Queues

The following MERVA USE & Branch message queues are involved in the data flow of workstation based USE functions:

MQSND1 Messages to be sent to MQSeries awaiting SWIFT ACK or NAK.

This queue contains SWIFT USE messages (for example MT96X and MT075) as well as SWIFT System and FIN messages created on the workstation. Messages from this queue will be sent to MQSeries and are routed to MQWAIT queue waiting for ACK or NAK from SWIFT.

Contributing queues: USESEND, SMAUTH1, SMCREATE, SMINCMPT

MQSND2 Messages to be sent to MQSeries

This queue contains free-format messages (MT999) that are created by the USE Background process. These messages include:

- Session key responses
- BK update messages
- Responses to pre-agreement update messages from other MERVA systems

Contributing queue: USESEND

MQRCV Messages received from MQSeries

This queue contains USE messages received from SWIFT via SWIFT Link on another MERVA system and via MQSeries. SWIFT ACK and NAK information is also received in this queue and is correlated with the messages in the MQWAIT queue. Free-format messages (MT999) from other MERVA systems are received in this queue to be routed to the USE Incoming commands queue of the USE Background process. Examples of such messages are:

- Session key requests
- BK Update messages
- Pre-agreement update messages

Contributing queue: none

MQWAIT Messages awaiting SWIFT ACK or NAK information

Each message created by MERVA USE & Branch that is supposed to be sent to SWIFT will be routed to this queue. As soon as the ACK or NAK information is received from SWIFT via the MQRCV queue, the messages waiting in this queue will be correlated.

Contributing queues: MQSND1, MQRCV

Part 4. Appendixes

Appendix A. Samples for Customizing Telex and USE Functions for MERVA Link on the Host System

This appendix contains examples of files used to customize the following areas of the workstation based functions environment for the Telex and USE functions:

- VTAM
- CICS
- IMS
- APPC/MVS
- APPC/IMS
- MERVA Link
- MERVA ESA

It also contains examples of startup jobs.

Samples for VTAM Definitions

The following sections contain examples of VTAM definitions for various resources and configurations.

VTAM Application: CICS LU

This example shows the VTAM definition of a CICS LU that is used by CICS for all terminal communication and intersystem connections.

CICS does not use the APPC services provided by VTAM. This is why APPC=YES must not be specified.

```
*-----*
*      VTAM Application - CICS LU      *
*-----*
          VBUILD TYPE=APPL
*
HOSTLU  APPL  ACBNAME=HOSTLU,          <= Param. a7 *
          AUTH=(PASS,ACQ,VPACE),VPACING=5,      *
          DLOGMOD=APPCLU62,                  <= Param. a12 *
          MODETAB=MODETB,PARSESS=YES
```

Figure 25. CICS LU Definition

VTAM Application: APPC/MVS LU

This example shows the VTAM definition of an APPC/MVS LU. The APPC/MVS LU must be used if IMS/VS is installed.

The VTAM parameters of the sample APPC/IMS LU are the same as the VTAM parameters of the sample APPC/MVS LU. An APPC/IMS LU, however, is defined in APPC/MVS with other parameters than an APPC/MVS LU. This is why the two LUs are named differently in this book.

```

*=====*
*      VTAM Application - APPC/MVS LU      *
*=====*
          VBUILD TYPE=APPL
*
HOSTLU  APPL  ACBNAME=HOSTLU,    APPC/MVS LU NAME      <= Param. d3 *
          APPC=YES,              VTAM APPC SUPPORT IS REQUIRED *
          AUTOSES=0,              *
          DDRAINL=NALLOW,        *
          DRESPL=NALLOW,        *
          EAS=20,                 *
          DSESLIM=99,            *
          MODETAB=MODETB,        *
          SECACPT=ALREADYV,      <= Param. d7 *
          SRBEXIT=YES,           *
          VERIFY=OPTIONAL,       *
          VPACING=2

```

Figure 26. APPC/MVS LU Definition

VTAM Application: APPC/IMS LU

This example shows the VTAM definition of an APPC/IMS LU.

The VTAM parameters of the sample APPC/IMS LU are the same as the VTAM parameters of the sample APPC/MVS LU. An APPC/IMS LU, however, is defined in APPC/MVS with other parameters than an APPC/MVS LU. This is why the two LUs are named differently in this book.

```

*=====*
*      VTAM Application - APPC/IMS LU      *
*=====*
          VBUILD TYPE=APPL
*
HOSTLU  APPL  ACBNAME=HOSTLU,    APPC/IMS LU NAME      <= Param. d3 *
          APPC=YES,              VTAM APPC SUPPORT IS REQUIRED *
          AUTOSES=0,              *
          DDRAINL=NALLOW,        *
          DRESPL=NALLOW,        *
          EAS=20,                 *
          DSESLIM=99,            *
          MODETAB=MODETB,        *
          SECACPT=ALREADYV,      <= Param. d7 *
          SRBEXIT=YES,           *
          VERIFY=OPTIONAL,       *
          VPACING=2

```

Figure 27. APPC/IMS LU Definition

VTAM Logmode Table Entries

This example shows the VTAM definitions of the two logmodes that are required for an APPC connection. The two logmodes are applicable for the connection of a workstation to all, CICS, APPC/IMS, and APPC/MVS systems.

```

*=====
* - Logmode table entries for an APPC connection *
* - Both Logmodes APPCLU62 and SNASVCMG are required *
*=====
MODETB  MODETAB
        .....

APPCLU62 MODEENT LOGMODE=APPCLU62,          <= Param. a12=d4 *
        COS=INTERACT,                        *
        TYPE=X'00',                          *
        FMPROF=X'13',                        *
        TSPROF=X'07',                        *
        PRIPROT=X'B0',                       *
        SECPROT=X'B0',                       *
        COMPROT=X'D0B1',                     *
        SSNDPAC=X'01',                       *
        SRCVPAC=X'04',                       *
        PSNDPAC=X'04',                       *
        RUSIZES=X'8585',                     8 X 2**5 = 256 *
        PSERVIC=X'06020000000000000000300'
*
*      141516171819202122232425'
SNASVCMG MODEENT LOGMODE=SNASVCMG,          *
        FMPROF=X'13',                        *
        TSPROF=X'07',                        *
        PRIPROT=X'B0',                       *
        SECPROT=X'B0',                       *
        COMPROT=X'D0B1',                     *
        SSNDPAC=X'01',                       *
        SRCVPAC=X'04',                       *
        PSNDPAC=X'04',                       *
        ENCR=B'0000',                        *
        RUSIZES=X'8585',                     8 X 2**5 = 256 *
        PSERVIC=X'06020000000000000000300'
*
*      141516171819202122232425'
        .....

END      MODEEND
        END  MODETB

```

Figure 28. Logmode Table Entries for an APPC Connection

VTAM PU and LU Definition for a PC Token Ring Network

This example shows the VTAM definitions of a PC PU and a PC APPC LU for a token ring attachment.

```

*=====*
* - PU-Definitions of the PC *
* - LU-Definitions of the PC *
*=====*
          VBUILD TYPE=SWNET, *
          MAXNO=12, *
          MAXGRP=5 *
*
PS2PU   PU   ADDR=01,PUTYPE=2,      <= Param. a2 *
          IDBLK=05D, *
          IDNUM=FFFFF, *
          MAXDATA=512, *
          MAXOUT=1,MAXPATH=2, *
          MODETAB=MODETB, *
          USSTAB=USSSNA,DISCNT=NO, *
          VPACING=2,PACING=1 *
*
PCLU00  LU   LOCADDR=0,      Independent LU      <= Param. a8=d2 *
          DLOGMOD=APPCLU62,RESSCB=10 *
          <= Param. a12=d4 *

```

Figure 29. PU and LU Definition for a PC in a Token Ring

VTAM PU and LU Definition for a PC with SDLC Line

This example shows the VTAM definitions of a PC connected to the SNA network via an SDLC line.

```

*=====*
* - PU-Definitions of the PC with SDLC line *
* - LU-Definitions of the PC with SDLC line *
*=====*
          . . . . .
F38L0036 LINE ADDRESS=(036,HALF) *
          SERVICE ORDER=(PS2PU) *
          <= SDLC line 036 *
          <= Param. a2 *
*
PS2PU   PU   ADDR=C1,PUTYPE=2, *
          MAXDATA=512, *
          MAXOUT=7, *
          PASSLIM=7, *
          SSCPFM=USSSCS, *
          ISTATUS=ACTIVE, *
          MODETAB=MODETB, *
          XID=YES *
*
PCLU00  LU   LOCADDR=0,      Independent LU      <= Param. a8=d2 *
          DLOGMOD=APPCLU62, *
          ISTATUS=ACTIVE *
          <= Param. a12=d4 *
*

```

Figure 30. PU and LU Definition of PC with SDLC Line

Samples for Defining CICS Resources

Depending on the version of your CICS system, you can define CICS resources by using either:

- The CICS macro level interface
- The CSD utility

Defining CICS Resources using the CICS Macro Level Interface

CICS Transactions

Figure 31 shows sample CICS program control table entries for MERVA Link on MERVA ESA.

```

.
.
.
*-----*
*   MERVA Link Transactions   *
*-----*
*
EKAR  DFHPCT TYPE=ENTRY,TRANSID=EKAR,           <= Param. c3 *
      PROGRAM=EKATR10,RSL=PUBLIC                <= Param. c4
*
EKAS  DFHPCT TYPE=ENTRY,TRANSID=EKAS,           <= Param. c5 *
      PROGRAM=EKAAS10                          <= Param. c6
*
.
.
.

```

Figure 31. CICS DFHPCT Entries for MERVA Link

CICS Processing Programs

Figure 32 shows sample CICS Processing Program Table entries for MERVA Link and MERVA USE on MERVA ESA.

```

*-----*
*   MERVA Link Programs and Tables   *
*-----*
*
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKAPT,RES=YES      Partner Table
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKAAS10           <= Param. c6
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKATR10           <= Param. c4
.
.
.
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKAMU133           <= Param. b15
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKAMU034           Called by EKAMU133
*
*-----*
*   MERVA USE Programs and Tables   *
*-----*
*
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKARTTXU           <= Param. b18
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DWSRTSK           <= Param. b17
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DWSLTT
.
.
.

```

Figure 32. CICS DFHPPT Entries for MERVA Link and USE

The entries in the processing program table (PPT) for the user-exit programs EKAMU034 and EKAMU133 are only required if the parameter LINK=NO is specified in the MFS program table definition, for example:

```

DSLMPPT NAME=EKAMU034,NUMBER=7034,TYPE=U,LINK=NO
DSLMPPT NAME=EKAMU133,NUMBER=7133,TYPE=U,LINK=NO      <= Param. b15

```

See "Message Format Service (MFS) Program Table (DSLMPPTT)" on page 157 for more information.

CICS APPC Connection

Figure 33 shows an extract from an example CICS Terminal Control Table entry. Only the part that is relevant for a VTAM APPC connection is shown.

```

          DFHTCT TYPE=INITIAL,          DEFINE TCT          *
              SUFFIX=xx,              *
              ACCMETH=VTAM             *
          .
          .
          .
*-----*
*      APPC Connection to a PC System  *
*-----*
*
          DFHTCT TYPE=SYSTEM,          *
              TRMTYPE=LUTYPE62,        *
              ACCMETH=VTAM,            *
              SYSIDNT=PS22,            <= Param. c1 *
              NETNAME=PCLU00,          <= Param. a8 *
              CONNECT=AUTO,            *
              FEATURE=PARALLEL,        *
              RUSIZE=256,               *
              BUFFER=256                *
          DFHTCT TYPE=MODESET,          *
              SYSIDNT=PS22,            <= Param. c1 *
              MAXSESS=(2,1),           *
              CONNECT=AUTO,            *
              MODENAM=APPCLU62         <= Param. a12
          .
          .
          .

```

Figure 33. CICS Terminal Control Table Entries

Defining CICS Resources using the CSD Utility

The following sections show examples of CICS resource definitions using the CICS CSD utility interface.

CICS Transactions and Programs

Figure 34 shows a number of resources that must be defined to CICS to perform USE. These resources are installed if MERVA Link and SWIFT Link are installed on your MERVA ESA system.

```

//CICSCSD JOB (Account),Name,....
.
.
//CSDUP EXEC PGM=DFHCSDUP,REGION=4096K
//STEPLIB DD DSN=CICS330.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHCSD DD DSN=CICSxx.CSD,DISP=SHR
//SYSIN DD *
.
.
DEFINE TRANSACTION(EKAR) GROUP(EKAGROUP) <= Param. c3
PROGRAM(EKATR10) RESSEC(NO) <= Param. c4
SPURGE(NO) TPURGE(NO) PRIORITY(1)
DEFINE TRANSACTION(EKAS) GROUP(EKAGROUP) <= Param. c5
PROGRAM(EKAAS10) RESSEC(NO) <= Param. c6
SPURGE(NO) TPURGE(NO) PRIORITY(1)
.
.
DEFINE PROGRAM(EKAPT) GROUP(EKAGROUP) Partner Table
LANGUAGE(ASSEMBLER)
EXECKEY(CICS) RESIDENT(YES)
DEFINE PROGRAM(EKAMU034) GROUP(EKAGROUP) Called by EKAMU133
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
DEFINE PROGRAM(EKAMU133) GROUP(EKAGROUP) <= Param. b15
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
DEFINE PROGRAM(EKAAS10) GROUP(EKAGROUP) <= Param. c6
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
DEFINE PROGRAM(EKATR10) GROUP(EKAGROUP) <= Param. c4
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
.
.
DEFINE PROGRAM(EKARTTXU) GROUP(DWSGROUP) <= Param. b18
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
DEFINE PROGRAM(DWSRTSK) GROUP(DWSGROUP) <= Param. b17
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
DEFINE PROGRAM(DWSLTT) GROUP(DWSGROUP)
LANGUAGE(ASSEMBLER)
EXECKEY(CICS)
.
.
//

```

Figure 34. CICS CSD Utility Example for Transactions and Programs for MERVA Link and SWIFT Link

CICS APPC Connection

Figure 35 shows the definitions for an APPC connection from a CICS/ESA system to a remote workstation and for the APPC sessions used by this connection.

```

//CICSCSD JOB (Account),Name,....
.
.
.
//CSDUP EXEC PGM=DFHCSDUP,REGION=4096K
//STEPLIB DD DSN=CICS330.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHCSD DD DSN=CICSxx.CSD,DISP=SHR
//SYSIN DD *
.
.
.
DEFINE CONNECTION(PS22) <= Param. c1
GROUP(TCT41)
NETNAME(PCLU00) <= Param. a8
ACCESSMETHOD(VTAM) PROTOCOL(APPC)
SINGLESESS(NO) DATASTREAM(USER) RECORDFORMAT(U)
AUTOCONNECT(ALL)
INSERVICE(YES) ATTACHSEC(LOCAL) BINDSECURITY(NO)
DEFINE SESSIONS(PS22AAC)
GROUP(TCT41)
CONNECTION(PS22) <= Param. c1
MODENAME(APPCLU62) <= Param. a12
PROTOCOL(APPC) MAXIMUM(2,1)
SENDSIZE(256) RECEIVESIZE(256) SESSPRIORITY(0)
AUTOCONNECT(ALL)
BUILDCHAIN(YES) USERAREALEN(0) IOAREALEN(0,0) RELREQ(NO)
DISCREQ(NO) NEPCCLASS(0) RECOVPTION(SYSDEFAULT)
.
.
.
//

```

Figure 35. CICS CSD Utility Example for a Connection and Sessions

Sample for Defining IMS Resources

The only resource that must be defined in IMS for MERVA USE & Branch is the transaction code of the MERVA Link sending transaction. The MERVA Link receiving transaction and the communication resources for MERVA USE & Branch are not defined in IMS.

A sample definition of the MERVA Link sending transaction EKAS in IMS is shown in Figure 36.

```

APPLCTN PSB=EKAAS10,          Sending ASP <= Param. d6
        SCHDTYP=PARALLEL
TRANSACT CODE=(EKAS,....,.... ) <= Param. d5
        MODE=SNGL,
        MSGTYPE=(SNGLSEG,,33),
        PARLIM=1,MAXRGN=1

```

Figure 36. IMS Transaction Definition for MERVA Link

Samples for Defining APPC/MVS Resources

APPC/MVS definitions apply if APPC/IMS of IMS/ESA is not installed in your system. All APPC functions for MERVA USE & Branch are provided by APPC/MVS in this case.

APPC/MVS LU in SYS1.PARMLIB

This example shows the definition of the APPC/MVS System Base LU in the member APPCPMxx of the MVS/ESA parameter library SYS1.PARMLIB.

```
LUADD ACBNAME(HOSTLU)          <= Param. d3
      TPLEVEL(USER)
      BASE                      <= Mandatory
      SCHED(ASCH)              <= APPC/MVS Scheduler
      TPDATA(SYS1.APPCTP)
```

Figure 37. APPC/MVS System Base LU Definition in APPCPMxx

APPC/MVS TP Profile

The following example contains utility statements to add a TP profile for the MERVA Link receiving TP EKATPI1 to the APPC/MVS TP profile data set SYS1.APPCTP. The profile defines that APPC/MVS must schedule the receiving transaction EKAR in an APPC/MVS initiator.

```
//APPCMVS JOB (Account),Name,...
      .
      .
//*****
//*  DEFINE A SYSTEM LEVEL STANDARD TP PROFILE          **
//*****
//DEFTP EXEC PGM=ATBSDFMU
//SYSPRINT DD SYSOUT=*
//SYSSDOUT DD SYSOUT=*
//SYSSDLIB DD DSN=SYS1.APPCTP,DISP=SHR
//SYSIN DD DATA,DLM=XX
      TPDELETE
          TPNAME(EKAR)                                <= Param. d8
          SYSTEM
      TPADD
          TPNAME(EKAR)                                <= Param. d8
          SYSTEM
          ACTIVE(YES)
          TPSCHED_DELIMITER(##)
          TPSCHED_TYPE(STANDARD)
          JCL_DELIMITER(END_OF_JCL)
//APPC JOB (Account),'Name',...
//
//EKARST EXEC PGM=EKATPI1                              <= Param. d9
//*****
//* The Library MERVA.SDSLLOD0 must contain the      ****
//* MERVA ESA Parameter Table DSLPRM and the        ****
//* MERVA Link Partner Table EKAPT that is used     ****
//* by all other MERVA ESA and MERVA Link processes. ****
//*****
//STEPLIB DD DSN=MERVA.SDSLLOD0,DISP=SHR
//SYSUDUMP DD DSN=xxx.SYSUDUMP.DATA,DISP=OLD
//DSLSNAP DD DSN=xxx.DSLSNAP.DATA,DISP=OLD
//EKASNAP DD DSN=xxx.EKASNAP.DATA,DISP=OLD
END_OF_JCL
##
XX
```

Figure 38. APPC/MVS TP Profile Definition

Note: If you use a security system, for example, RACF[®], you should give the permission for the data sets and resources.

Samples for Defining APPC/IMS Resources

APPC/IMS definitions apply if APPC/IMS of IMS/ESA is installed in your system. The APPC functions for MERVA USE & Branch are provided by APPC/IMS and APPC/MVS in this case.

APPC/IMS LU in SYS1.PARMLIB

The APPC/IMS Base LU must be used when APPC/IMS of IMS/ESA is installed. The following example shows the definition of the APPC/IMS Base LU in the member APPCPMxx of the MVS/ESA parameter library SYS1.PARMLIB.

```
LUADD ACBNAME(HOSTLU)           <= Param. d3
      TPLEVEL(USER)
      BASE                       <= Mandatory
      SCHED(I410)                <= APPC/IMS Scheduler
      TPDATA(SYS1.APPCTP)
```

Figure 39. APPC/IMS Base LU Definition in APPCPMxx

APPC/IMS TP Profile

The following example contains utility statements to add a TP profile for the MERVA Link receiving TP EKATPI1 to the APPC/MVS TP profile data set SYS1.APPCTP. The profile defines that APPC/IMS must schedule the receiving transaction EKARI4 in an IMS message processing region (MPR).

```
//APPCMVS JOB (Account),Name,....
      .
      .
      .
//*****
//*   DEFINE A SYSTEM LEVEL STANDARD TP PROFILE   **
//*****
//DEFTP EXEC PGM=ATBSDFMU
//SYSPRINT DD SYSOUT=*
//SYSSDOUT DD SYSOUT=*
//SYSSDLIB DD DSN=SYS1.APPCTP,DISP=SHR
//SYSIN DD DATA,DLM=XX
      TPDELETE
          TPNAME(EKARI4)           <= Param. d8
          SYSTEM
      TPADD TPSCHED_EXIT(DFSTPPE0)
          TPNAME(EKARI4)           <= Param. d8
          SYSTEM
          ACTIVE(YES)
          TPSCHED_DELIMITER(##)
          TRANCODE=EKATPI1         <= Param. d9
          CLASS=24
          MAXRGN=1
          RACF=FULL
##
XX
```

Figure 40. APPC/IMS TP Profile Definition

An IMS MPR supporting IMS message class 24 must be active in this sample to start transaction EKARI4. The startup job of that MPR must include a data definition statement for the MERVA Link SNAP dump data set (//EKASNAP DD ...).

Samples for Defining MERVA Link Resources

MERVA Link resources are defined in the MERVA Link Partner Table EKAPT. The EKAPT consists of a header and of ASP and MTP entries. The name of the local MERVA Link node is defined in the EKAPT header.

The parameters of the application (Telex, or USE, or both Telex and USE) are defined in an ASP entry. An EKAPT ASP entry is always associated with an MTP entry.

The parameters of the connection to the PC partner system are defined in an MTP entry. These parameters depend on the data communication subsystem (CICS, APPC/MVS, or APPC/IMS) that is used by MERVA Link in the MERVA ESA environment.

An EKAPT MTP entry is always associated with an ASP entry.

The following sections show an example of an EKAPT header, an example of an ASP entry for Telex and USE, as well as examples of MTP entries for use with CICS, APPC/MVS, and APPC/IMS.

EKAPT Header

Figure 41 shows a sample MERVA Link partner table header and names the other entries that must be part of the EKAPT.

A sample MERVA Link Partner Table for Telex and USE is provided in the member EKAPTTXU.

```
*-----*
*      Example of a MERVA Link Partner Table
*-----*
      EKAPT TYPE=INITIAL,
              NODE=MESAMTN      LOCAL MERVA Link NODE NAME <= Param. b1      *
      EKAPT TYPE=ASP,           APPLICATION SUPPORT PROCESS                    *
      .
      .
      EKAPT TYPE=MTP,           MESSAGE TRANSFER PROCESS                      *
      .
      .
      EKAPT TYPE=FINAL          END OF THE PARTNER TABLE
      END
```

Figure 41. MERVA Link Partner Table EKAPT

ASP Entry for Telex and USE

Figure 42 shows a sample EKAPT ASP entry for Telex and USE. This ASP entry can be used in all data communication environments.

```

*-----
* ASP for Telex and USE
*-----
EKAPT TYPE=ASP,          APPLICATION SUPPORT PROCESS *
NAME=(MESAASP,'APPC CONNECTION TO PCLU00'),          *<= Param. b2
DEST=(PCLU00,M2ASPA),          DESTINATION ADDRESS *<= Param. a8=d2/b24,b3
SENDQC=(TX2USESQ,TX2URG,TX2NRM),          SEND QUEUES *<= Param. b20
CONTROL=TX2USECQ,          MERVA Link CONTROL QUEUE *<= Param. b8
SECURE=ENCRYPT,          ENCRYPT MESSAGE TEXT *<= Param. b11
IRROUTE=(ACK,TX2WAIT,CTLQ),          REC REPORT CORRELAT. *<= Param. b21
JOURNAL=(OM,IR,IM),          JOURNAL TRAFFIC *<= Param. b22
MFSEXIT=7133,          MFS EXIT NUMBER *<= Param. b15
TRAN=EKAS,          SENDING TRANSACTION *<= Param. c5=d5
FORMAT=(MCB,TCOV,#),          TELEX OR SWIFT FORMAT *<= Param. b16
MTP=MTP          RELATED MTP          <= Param. b4

```

Figure 42. EKAPT ASP Entry for Telex and USE

MTP Entry for CICS

Figure 43 shows a sample EKAPT MTP entry for a connection to a workstation in the MERVA ESA CICS environment.

```

*-----
* MTP for a Connection of CICS and PC or RS/6000
*-----
EKAPT TYPE=MTP,          MESSAGE TRANSFER PROCESS *
NAME=(MTP,MESAMTP),          MTP NAMES *<= Param. b4/b5
LINK=(APPC,PS22),          PARTNER SYSTEM FOR CICS *<= Param. c1
CONNECT=QUEUE,          ALLOCATE W/O NOQUEUE OPTION *
PARTNER=(M2MTPA,EKAOSVR),          PARTNER MTP INFO. *<= Param. b6/a9
ASP=MESAASP          RELATED ASP          <= Param. b2

```

Figure 43. EKAPT MTP Entry for MERVA ESA CICS

MTP Entry for APPC/MVS

Figure 44 shows a sample EKAPT MTP entry for a connection to a workstation in the MERVA ESA APPC/MVS environment. This MTP entry applies if APPC/IMS of IMS/ESA is not installed.

```

*-----
* MTP for a Connection of APPC/MVS and PC or RS/6000
*-----
EKAPT TYPE=MTP,          MESSAGE TRANSFER PROCESS *
NAME=(MTP,MESAMTP),          MTP NAMES *<= Param. b4/b5
MTPCI=(,EKATP01),          SENDING MT PROGRAM *<= Param. d1
LINK=(APPC,PCLU00,APPCLU62),          PARTNER SYSTEM *<= Param. d2/d4
PARTNER=(M2MTPA,EKAOSVR),          PARTNER MTP INFO. *<= Param. b6/a9
ASP=MESAASP          RELATED ASP          <= Param. b2

```

Figure 44. EKAPT MTP Entry for MERVA ESA APPC/MVS

MTP Entry for APPC/IMS

Figure 45 shows a sample EKAPT MTP entry for a connection to a workstation in the MERVA ESA APPC/IMS environment. This MTP entry applies if APPC/IMS of IMS/ESA is installed in the MVS/ESA SP Version 4.3 environment at least.

APPC/IMS is not supported by MERVA Link if it is installed in the MVS/ESA SP Version 4.2 environment.

```

*-----
* MTP for a Connection of APPC/IMS and PC or RS/6000
*-----
      EKAPT TYPE=MTP,          MESSAGE TRANSFER PROCESS      *
      NAME=(MTP,MESAMTP),    MTP NAMES                    *<= Param. b4/b5
      MTPCI=(,EKATP01),      SENDING MT PROGRAM          *<= Param. d1
      LOCLU=HOSTLU,          LOCAL APPC/IMS BASE LU        *<= Param. d3
      LINK=(APPC,PCLU00,APPCLU62), PARTNER SYSTEM          *<= Param. d2/d4
      PARTNER=(M2MTPA,EKAOSVR), PARTNER MTP INFO.         *<= Param. b6/a9
      ASP=MESAASP            RELATED ASP                    <= Param. b2

```

Figure 45. EKAPT MTP Entry for MERVA ESA APPC/IMS

Samples for Defining MERVA ESA Resources

MERVA ESA resources must be defined for the MERVA USE & Branch. These resources are:

- MFS programs (MFS user exits)
- Function table entries
- Logical terminal table
- Routing tables.

Examples for definitions of these resources are shown in the following sections.

Message Format Service (MFS) Program Table (DSLMPPT)

Figure 46 shows an example MFS program table. It is used to define, for example, a MERVA Link user exit. All programs are identified by module numbers. By default, the MERVA Link user exits are loaded dynamically from the load library (LINK=NO). The table is used exclusively by the module DSLMMFS to set up the link to MFS user exits.

A sample for MERVA Link user exits is provided in the copy book EKAMPPTC.

```

DSLMPPT DSLMPT TYPE=INITIAL
      .
      .
      .
*-----
*          MERVA Link MFS User Exits
*-----
      DSLMPT NAME=EKAMU034,NUMBER=7034,TYPE=U,LINK=NO
      DSLMPT NAME=EKAMU133,NUMBER=7133,TYPE=U,LINK=NO   <= Param. b15
      .
      .
      .
      DSLMPT TYPE=FINAL
      END

```

Figure 46. MFS Program Table (DSLMPPT)

The user exit EKAMU133 provides most of the user exit functions required by MERVA USE & Branch. It is the only user exit that is called by MERVA Link for MERVA USE & Branch.

The user exit EKAMU034 is called by EKAMU133 only when a telex message is sent from MERVA ESA to MERVA USE & Branch (refer to "MERVA Link MFS User Exit EKAMU034" on page 74). EKAMU034 must not be associated with a MERVA Link ASP (MFSEXIT=7034 must never be specified in a MERVA Link ASP entry).

Function Table for Telex and USE (DSLFNTT)

Figure 47 shows the functions that are necessary for Telex only. Figure 48 shows the common functions for Telex and USE, and the functions necessary for USE only.

Note: All functions are defined in the copy books ENLFNTT2 and DWSFNNTC.

```
*****
*
*           FUNCTIONS FOR TELEX
*
*****
*           MERVA Link Send Queues for Telex Message Processing
*           DSLFNT NAME=TX2URG,QUEUE=YES,THRESH=10,           <= Param. b20  +
*           TRAN=EKAS,                                       <= Param. c5=d5  +
*           PROT=YES,NOPR=NO,                                  +
*           NEXT=TX2URG,PRINT=TX2PR0,MSGID=TCOV,             +
*           DESCR='Send Queue for Urgent Telex Messages'
*           DSLFNT NAME=TX2NRM,QUEUE=YES,THRESH=100,         <= Param. b20  +
*           TRAN=EKAS,                                       <= Param. c5=d5  +
*           PROT=YES,NOPR=NO,                                  +
*           NEXT=TX2NRM,PRINT=TX2PR0,MSGID=TCOV,             +
*           DESCR='Send Queue for Normal Telex Messages'
*           MERVA Link Wait Queue for Telex Message Processing
*           DSLFNT NAME=TX2WAIT,                               <= Param. b21  +
*           DE=NO,                                           +
*           FRAME=(0TOP,0BOT),                                +
*           KEY1=(EKAAMSID,10,1), THIS KEY1 REQD.BY MERVA Link +
*           KEY2=(ENLTXREF,16,1),                             +
*           NEXT=TX2WAIT,                                     <= Param. b21  +
*           PRFORM=(E,0),                                     +
*           PRINT=TX2PR0,                                     +
*           PROT=YES,                                         +
*           QUEUE=YES,                                        +
*           NOPR=NO,MSGID=TCOV,                               +
*           THRESH=50,                                        +
*           DESCR='Workstation Telex Wait for Acknowledgments'
*           Telex Link Station Command Queue
*           DSLFNT NAME=TX2TLC,                                +
*           DE=YES,                                           +
*           FRAME=(0TOP,0BOT),                                +
*           ROUTE=ENLR2TLC,                                   <= Param. b19  +
*           PRINT=TX2PR0,                                     +
*           QUEUE=YES,                                        +
*           KEY1=(ENLP2KY1,24),                               +
*           KEY2=(ENLP2UCM,24),                               +
*           SPCMND=DEL,                                       +
*           THRESH=100,                                       +
*           NOPR=NO,MSGID=TCOV,                               +
*           DESCR='Workstation Telex Link Station Commands'
```

Figure 47. Function Table Entries (ENLFNTT2)

```

*****
*
*      COMMON FUNCTIONS FOR TELEX AND SWIFT USE
*
*****
*      MERVA Link SEND QUEUE FOR TELEX AND USE MESSAGES
*      DSLFNT NAME=TX2USESQ,                               <= Param. b7  *
*      TRAN=EKAS,                                           <= Param. c5=d5 *
*      QUEUE=YES,KEY1=(SWBHLT,12),                          *
*      STORE=(SMALL,31900),                                NO LARGE MESSAGES *
*      NOPR=NO,PROT=YES,PRINT=TX2PR0,                      *
*      MSGID=TCOV,                                          *
*      DESCR='Send Queue for Telex and USE'
*
*      MERVA Link CONTROL QUEUE FOR TELEX AND USE MESSAGE PROCESSING
*      DSLFNT NAME=TX2USECQ,                               <= Param. b8  *
*      KEY1=(EKACCLASS,2,1),ROUTE=EKARTTXU,               <= Param. b18 *
*      QUEUE=YES,DQFILL=NO,                                *
*      MSGID=MCOV,                                          *
*      DESCR='Control Queue for Telex and USE'
*
*****
*
*      FUNCTIONS FOR SWIFT USE
*
*****
*      DELETE PROCESSED SWIFT SESSION KEY REQUESTS AND RESPONSES
*      DSLFNT NAME=USEDEL,QUEUE=DUMMY,                      *
*      DESCR='Delete Processed SWIFT USE Messages'
*
*      SWIFT USE MESSAGES ERROR QUEUE
*      DSLFNT NAME=USEERROR,QUEUE=YES,KEY1=(SWBHLT,12),    *
*      KEY2=(SWAHMT,3),                                    *
*      THRESH=10,SPCMND=DEL,                               *
*      DESCR='SWIFT USE Messages Errors'
*
*      AUTHENTICATOR KEY FILE UPDATES RECEIVED FROM THE USE PC
*      DSLFNT NAME=BKEUPDT,QUEUE=YES,THRESH=10,PROT=YES,   *
*      NEXT=USEERROR,TRAN=DWSA,STATUS=NOHOLD,              *
*      CHECK=NO,                                           *
*      DESCR='BKE Authenticator Key File Updates'
*
*      QUEUE FOR CREATING AND UPDATING BKE PRE-AGREEMENTS
*      THIS QUEUE IS BOTH DATA ENTRY AND FORMS QUEUE
*      DSLFNT NAME=BKEPREAG,QUEUE=YES,NOPR=YES,PRINT=L1PR0, *
*      NEXT=TX2USESQ,                                       <= Param. b7  *
*      SPCMND=DEL,DE=YES,                                    *
*      KEY1=(SW20,16),KEY2=(SW21,16), TRN, OWN ID/CORR ID *
*      CHECK=NO,COPY=BKEPREAG,                              *
*      DESCR='BKE Pre-Agreement Creation and Update'
*

```

Figure 48. Function Table Entries (DWSFN TTC) (Part 1 of 2)

```

* RECEIVED SINGLE SWIFT SESSION KEYS
  DSLFNT NAME=SLSRECV,QUEUE=YES,NEXT=USEERROR,KEY1=(SWBHLT,12), *
    DQFILL=NO, *
    DESCR='Received SWIFT Session Keys'
*
* RECEIVED SWIFT PRELOAD SESSION KEYS
  DSLFNT NAME=SLSLOAD,QUEUE=YES,NEXT=USEERROR,KEY1=(SWBHLT,12), *
    DQFILL=NO, *
    DESCR='Received SWIFT Preload Session Keys'
*
* PRELOADED GPA SESSION KEYS
* KEY 1 IS LTNAME + G, BLANK, ICCPARMS (6), BLANK, LSN/SSN (4)
* KEY 2 IS LTNAME + G, BLANK, LSN/SSN (4), BLANK, ICCPARMS (6)
  DSLFNT NAME=SLSGPA,QUEUE=YES,KEY1=(,22),KEY2=(,22), *
    DQFILL=NO,TOF=NO,MSGSEL=LIST, NO TOFS STORED *
    SPCMND=DEL,STORE=(SMALL,20), DELETE WITH QSN *
    DESCR='Preloaded GPA Session Keys'
*
* PRELOADED FIN SESSION KEYS
* KEY 1 IS LTNAME + F, BLANK, ICCPARMS (6), BLANK, LSN/SSN (4)
* KEY 2 IS LTNAME + F, BLANK, LSN/SSN (4), BLANK, ICCPARMS (6)
  DSLFNT NAME=SLSFIN,QUEUE=YES,KEY1=(,22),KEY2=(,22), *
    DQFILL=NO,TOF=NO,MSGSEL=LIST, NO TOFS STORED *
    SPCMND=DEL,STORE=(SMALL,20), DELETE WITH QSN *
    DESCR='Preloaded FIN Session Keys'
*
* QUEUE TO KEEP LSN/SSN FROM TERMINATION TO NEXT STARTUP
  DSLFNT NAME=SLSLTT,QUEUE=YES,KEY1=(,10), LT + G/F *
    DQFILL=NO,TOF=NO,SPCMND=DEL, NO TOFS STORED *
    DESCR='Keep SWIFT LSN/SSN'
  EJECT
  DSLGENS
* END OF SWIFT DEFINITIONS FOR FUNCTION TABLE

```

Figure 48. Function Table Entries (DWSFNNTC) (Part 2 of 2)

Logical Terminal Table for USE (DWSLTT)

| This table is used by both MERVA Link and MERVA-MQI Attachment, and is
| described in "Logical Terminal Table for USE (DWSLTT)" on page 178.

Routing Table for USE (DWSRTSK)

| This table is used by both MERVA Link and MERVA-MQI Attachment, and is
| described in "Routing Table for USE (DWSRTSK)" on page 179.

Routing Table for Telex (ENLR2TLC)

Figure 49 shows an example routing table that is used for the Telex Link station command queue.

```

          TITLE 'ROUTING TABLE FOR TELEX LINK STATION COMMAND QUEUE'
*****
ENLR2TLC DSLROUTE TYPE=DEFINE, FIELD=(TYPE, ENLP2TYP, , , , VFIRST),          *
          NOTFND=COMMAND
*         IF ENLP2TYP EXISTS IT IS A RESPONSE.
*         IF IT DOES NOT EXIST IT IS AN OUTGOING COMMAND.
          DSLROUTE TYPE=SET, TARGET='TX2TLC', GOTO=END
COMMAND  DSLROUTE TYPE=SET, TARGET='TX2USESQ', GOTO=END          <= Param. b7
END      DSLROUTE TYPE=FINAL, TARGET='TX2ERROR'
          END

```

Figure 49. Routing Table (ENLR2TLC)

Routing Table EKARTTXU for Telex and USE

This table is used by both MERVA Link and MERVA-MQI Attachment, and is described in “Routing Table for Telex and USE (EKARTTXU)” on page 179.

Telex Only Scenario

If you prefer to have a customization for telex only instead of for telex and USE together, you have to do the following:

1. Use the MERVA Link partner table EKAPTTLS instead of EKAPTTXU.
Modify EKAPTTLS according to your needs. Assemble and link-edit it as EKAPT.
2. Use the routing table EKARTTLS instead of EKARTTXU.
EKARTTLS is associated to the MERVA Link control queue EKATX2CQ.
If you define another control queue than EKATX2CQ in the partner table EKAPTTLS, you must also change the name of the control queue in the routing table EKARTTLS.
3. Replace the MERVA Link send queue TX2USESQ by TX2TLCSQ in the routing table ENLR2TLC.
Assemble and link-edit the routing table.
4. Generate the MERVA ESA function table DSLFNNTT.
Use the copy book ENLFNTT2. The appropriate entries are shown in Figure 50.
Copy book ENLFNTTC references ENLFNTT2. Therefore you must include ENLFNTTC for the function table generation.
Assemble and link-edit the function table.

Note: You can omit the copy book DWSFNNTTC.

```

DSL FNT NAME=EKATX2CQ,KEY1=(EKAClass,2,1),ROUTE=EKARTTLS,      *
    QUEUE=YES,THRESH=50,                                         *
    MSGID=MCOV,                                                  *
    DESCR='Telex Link Station Control Queue'
*
DSL FNT NAME=TX2TLCsq,      SEND QUEUE FOR TXIP/2 COMMANDS      *
    NEXT=TX2TLCsq,                                               *
    PRINT=TX2PR0,                                                *
    PROT=YES,                                                     *
    QUEUE=YES,                                                    *
    TRAN=EKAS,                                                    *
    THRESH=100,                                                  *
    NOPR=NO,MSGID=TCOV

```

Figure 50. Function Table Entries for Telex (ENLFNTT2)

USE Only Scenario

When using MERVA Link, if you prefer to have a customization for USE only instead of for telex and USE together, you have to do the following:

1. Use the MERVA Link partner table EKAPTUSE instead of EKAPTTXU.
Modify EKAPTUSE according to your needs. Assemble and link-edit it as EKAPT.
2. Use the routing table DWSRTCT instead of EKARTTXU.
DWSRTCT is associated to the MERVA Link control queue USEMLCT.
3. Replace the MERVA Link send queue TX2USESQ by USESEND in the following routing tables:
 - DWSRTSK
 - DWSL1IN
 - DWSL1OUT
 - DWSL2IN
 - DWSL2OUT
 - DWSL3GPI
 - DWSL3GPO
 - DWSL3FII
 - DWSL3FIO.

Assemble and link-edit these routing tables.

4. Replace the MERVA Link send queue TX2USESQ by USESEND in the copy book DWSFNNTC.

This applies to the parameter NEXT of the function BKEPREAG.

5. Generate the MERVA ESA function table DSLFNNT.

Use the copy book DWSFNNTC. The appropriate entries are shown in Figure 51.

Assemble and link-edit the function table.

Note: You can omit the copy book ENLFNNTC that references ENLFNNT2.

```

*      MERVA Link SEND QUEUE FOR ALL USE MESSAGES
      DSLFNT NAME=USESEND,TRAN=EKAS,QUEUE=YES,KEY1=(SWBHLT,12),      *
          STORE=(SMALL,31900),      NO LARGE MESSAGES      *
          DESCR='Send USE Messages to the USE PC'
*
*      MERVA Link CONTROL QUEUE FOR USE MESSAGE PROCESSING
      DSLFNT NAME=USEMLCT,KEY1=(EKACCLASS,2,1),ROUTE=DWSRTCT,      *
          QUEUE=YES,DQFILL=NO,      *
          DESCR='SWIFT Session Key Control Queue'
*
*      QUEUE FOR CREATING AND UPDATING BKE PRE-AGREEMENTS
      THIS QUEUE IS BOTH DATA ENTRY AND FORMS QUEUE
*      DSLFNT NAME=BKEPREAG,QUEUE=YES,NOPR=YES,PRINT=L1PR0,      *
          NEXT=USESEND,      *
          SPCMND=DEL,DE=YES,      *
          KEY1=(SW20,16),KEY2=(SW21,16), TRN, OWN ID/CORR ID      *
          CHECK=NO,COPY=BKEPREAG,      *
          DESCR='BKE Pre-Agreement Creation and Update'

```

Figure 51. Function Table Entries for USE (DWSFNNTC)

Sample Startup Jobs for MERVA Link

| The way that MERVA Link is started depends on whether it is running under CICS
| or IMS.

Starting MERVA Link under CICS

| Under CICS, MERVA Link is started together with the rest of MERVA ESA. The
| following example shows excerpts from a sample startup job for CICS/ESA
| Version 4.1 for use with MERVA Link.

```
//CICSESA JOB (Account),Name,....
      .
      .
//DFHSIP EXEC PGM=DFHSIP,REGION=6144K,PARM=SYSIN
//STEPLIB DD DSN=CICS410.SDFHAUTH,DISP=SHR
//DFHRPL DD DSN=MERVA.SDSLLODC,DISP=SHR
//DFHRPL DD DSN=MERVA.SDSLLODB,DISP=SHR
//      DD DSN=CICS410.SDFHLOAD,DISP=SHR
      .. more DD-statements ..
      .....
//-----
//*      MERVA Link Conversation Trace Data Set
//-----
//EKACTRC DD DSN=MERVA.DSLCTRC1,DISP=SHR
//*
//-----
//*      CICS Startup Parameters
//-----
//SYSIN DD *
          SIT=6$,
          APPLID=HOSTLU,          CICS VTAM APPLID          <= Param. a7
          DCT=41,
          DMBPL=4,          Max Size of DMB Pool
          DSALIM=4M,
          EDSALIM=80M,
          FCT=NO,
          GRPLIST=(DFHLIST,MLIST),
          ICV=100,
          ISC=YES,          Include APPC Support          <= Param. c2
          START=COLD,
          .
          .
          .
/*
//
```

Starting MERVA Link under IMS

| Under IMS, MERVA Link is using an MPP job; no special definitions are required.

Appendix B. Samples for Customizing USE Functions for MERVA-MQI Attachment on the Host System

This appendix contains examples of files used to customize the following areas of the workstation based functions:

- CICS
- IMS
- MQSeries
- MERVA-MQI Attachment
- MERVA ESA

It also contains examples of startup jobs.

Samples for Defining CICS Resources

Depending on the version of your CICS system, you can define CICS resources by using either:

- The CICS macro level interface
- The CSD utility

Defining CICS Resources using the CICS Macro Level Interface

CICS Transactions

Figure 52 shows sample CICS program control table entries for MERVA-MQI Attachment on MERVA ESA.

```
.  
. .  
. .  
*-----*  
*      MQI Attachment Transactions                *  
*-----*  
*  
DFHPCT TYPE=ENTRY,TRANSID=DSLRL,                <= Param.  g1 *  
        PROGRAM=DSLKQR,TRNPRTY=20,PRIVATE=YES,   <= Param.  g2 *  
        RSL=PUBLIC,RSLC=NO  
DFHPCT TYPE=ENTRY,TRANSID=DSLRL,                <= Param.  g3 *  
        PROGRAM=DSLKQS,TRNPRTY=20,PRIVATE=YES,   <= Param.  g4 *  
        RSL=PUBLIC,RSLC=NO
```

Figure 52. CICS DFHPCT Entries for MERVA-MQI Attachment

CICS Processing Programs

Figure 53 shows sample CICS Processing Program Table entries for MERVA-MQI Attachment and MERVA USE on MERVA ESA.

```

*-----*
*      MQI Attachment Programs and Tables      *
*-----*
*
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKPROC      Process Table
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKQR        <= Param. g2
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKSUBR      Called by g2
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKQS        <= Param. g4
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKSUBS      Called by g4
.
.
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DSLKQ133      <= Param. f14
*
*-----*
*      MERVA USE Programs and Tables          *
*-----*
*
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=EKARTTXU      <= Param. f16
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DWSRTSK      <= Param. f15
DFHPPT TYPE=ENTRY,RSL=PUBLIC,PROGRAM=DWSLTT
.
.
.

```

Figure 53. CICS DFHPPT Entries for MERVA-MQI Attachment and USE

The entry in the processing program table (PPT) for the user-exit program DSLKQ133 is only required if the parameter LINK=NO is specified in the MFS program table definition, for example:

```

DSLMPPT NAME=DSLKQ133,TYPE=U,NUMBER=8133,LINK=NO,<= Param. f14 *
LANG=ASM

```

See “Message Format Service (MFS) Program Table (DSLMPPT)” on page 175 for more information.

Defining CICS Resources using the CSD Utility

The following example shows how to use the CSD utility to define the CICS resources needed by the USE functions. These resources are installed if MERVA-MQI Attachment and SWIFT Link are installed on your MERVA ESA system.

```

//CICSCSD JOB (Account),Name,...
.
.
//CSDUP EXEC PGM=DFHCSDUP,REGION=4096K
//STEPLIB DD DSN=CICS410.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHCSD DD DSN=CICSxx.CSD,DISP=SHR
//SYSIN DD *
.
.
DEFINE TRANSACTION(DSLR) GROUP(DSLGROUP) <= Param. g1
PROGRAM(DSLKQR) RESSEC(NO) <= Param. g2
SPURGE(NO) TPURGE(NO) PRIORITY(20)
ISOLATE(NO) TASKDATAKEY(CICS)
DEFINE TRANSACTION(DSLQ) GROUP(DSLGROUP) <= Param. g3
PROGRAM(DSLKQS) RESSEC(NO) <= Param. g4
SPURGE(NO) TPURGE(NO) PRIORITY(20)
ISOLATE(NO) TASKDATAKEY(CICS)
.

```


DSLKCSQC (CICS only)

The following example shows excerpts from the member DSLKCSQC.

```
*****
* Control Queue *
*****
DELETE QLOCAL(DSL.MQI.CONTROL) PURGE
DEFINE QLOCAL(DSL.MQI.CONTROL)                <= Param. e18 +
  DESCR('MERVA-MQI Control Queue for a Send Process') +
  DEFPSIST(YES) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  PUT(ENABLED) +
  MAXDEPTH(32767) +
  QDPMAXEV(ENABLED) +
  SHARE +
  NOHARDENBO +
  NOTRIGGER +
  TRIGTYPE(NONE) +
  USAGE(NORMAL) +
  REPLACE

*****
* Receive Queue *
*****
*   If required, modify the following parameter:
*   INITQ(CICS01.INITQ)                <= Param. e13
*
DELETE QLOCAL(DSL.MQI.RECEIVE) PURGE
DEFINE QLOCAL(DSL.MQI.RECEIVE)                <= Param. e10 +
  DESCR('MERVA-MQI Receive Queue for a Receive Process') +
  DEFPSIST(YES) +
  DEFSOPT(SHARED) +
  GET(ENABLED) +
  PUT(ENABLED) +
  QDPMAXEV(ENABLED) +
  SHARE +
  NOHARDENBO +
  INITQ(CICS01.INITQ)                <= Param. e13 +
  PROCESS(DSL.MQI.TRIGGER) +
  TRIGGER +
  TRIGMPRI(0) +
  TRIGTYPE(FIRST) +
  USAGE(NORMAL) +
  REPLACE

*****
* Send and Receive Queue *
*****
DELETE QLOCAL(DSL.MQI.SNDRCV) PURGE
DEFINE QLOCAL(DSL.MQI.SNDRCV)                +
  LIKE(DSL.MQI.RECEIVE) +
  DESCR('MERVA-MQI Send and Receive Demo Queue') +
  REPLACE

*****
* Reply-to Queue *
*****
DELETE QLOCAL(DSL.MQI.REPLY_TO_Q) PURGE
DEFINE QLOCAL(DSL.MQI.REPLY_TO_Q)                <= Param. e17 +
  LIKE(DSL.MQI.SNDRCV) +
  DESCR('MERVA-MQI Reply-to Queue') +
  REPLACE

*****
* Trigger Process *
*****
```

```

*   If required, modify the following parameter:
*   APPLICID(DSLR)                               <= Param. g1
*
DELETE PROCESS(DSL.MQI.TRIGGER)
DEFINE PROCESS(DSL.MQI.TRIGGER)                  +
DESCR('MERVA-MQI Trigger Process for MQI Receive Queue') +
APPLICID(DSLR)                                  <= Param. g1 +
APPLTYPE(CICS)                                   +
REPLACE                                          +

```

DSLKCSQI (IMS only)

The following example shows excerpts from the member DSLKCSQI.

```

*****
* Control Queue *
*****
DELETE QLOCAL(DSL.MQI.CONTROL) PURGE
DEFINE QLOCAL(DSL.MQI.CONTROL)                  <= Param. e18 +
DESCR('MERVA-MQI Control Queue for a Send Process') +
DEFPSIST(YES)                                   +
DEFSOPT(SHARED)                                 +
GET(ENABLED)                                    +
PUT(ENABLED)                                    +
MAXDEPTH(32767)                                 +
QDPMAXEV(ENABLED)                              +
SHARE                                           +
NOHARDENBO                                     +
NOTRIGGER                                       +
TRIGTYPE(NONE)                                 +
USAGE(NORMAL)                                  +
REPLACE                                          +

*****
* Receive Queue *
*****
*   If required, modify the following parameter:
*   INITQ(IMS01.INITQ.IMS61)                     <= Param. e13
*
DELETE QLOCAL(DSL.MQI.RECEIVE) PURGE
DEFINE QLOCAL(DSL.MQI.RECEIVE)                  <= Param. e10 +
DESCR('MERVA-MQI Receive Queue for a Receive Process') +
DEFPSIST(YES)                                   +
DEFSOPT(SHARED)                                 +
GET(ENABLED)                                    +
PUT(ENABLED)                                    +
QDPMAXEV(ENABLED)                              +
SHARE                                           +
NOHARDENBO                                     +
INITQ(IMS01.INITQ.IMS61)                       <= Param. e13 +
PROCESS(DSL.MQI.TRIGGER.DSLR)                  +
TRIGGER                                         +
TRIGMPRI(0)                                    +
TRIGTYPE(FIRST)                                +
USAGE(NORMAL)                                  +
REPLACE                                          +

*****
* Send and Receive Queue *
*****
DELETE QLOCAL(DSL.MQI.SNDRCV) PURGE
DEFINE QLOCAL(DSL.MQI.SNDRCV)                  +
LIKE(DSL.MQI.RECEIVE)                          +
DESCR('MERVA-MQI Send and Receive Demo Queue') +
REPLACE                                          +

*****
* Reply-to Queue *

```

```

*****
DELETE QLOCAL(DSL.MQI.REPLY_TO_Q) PURGE
DEFINE QLOCAL(DSL.MQI.REPLY_TO_Q)          <= Param. e17  +
      LIKE(DSL.MQI.SNDRCV)                  +
      DESCR('MERVA-MQI Reply-to Queue')    +
      PROCESS(DSL.MQI.TRIGGER.DSLR1)       +
      REPLACE

*****
* Trigger Process for Transaction Code DSLR *
*****
*   If required, modify the following parameters:
*   APPLICID(DSLR)                          <= Param. h1
*   USERDATA('MSGLIM=65535')
*
DELETE PROCESS(DSL.MQI.TRIGGER.DSLR)
DEFINE PROCESS(DSL.MQI.TRIGGER.DSLR)      +
      DESCR('MERVA-MQI Trigger Process for MQI Receive Queue') +
      APPLICID(DSLR)                          <= Param. h1  +
      APPLTYPE(IMS)                            +
      USERDATA('MSGLIM=65535')                +
      REPLACE

*****
* Trigger Process for Transaction Code DSLR1 *
*****
*   If required, modify the following parameter:
*   APPLICID(DSLR1)                          <= Param. h1
*
DELETE PROCESS(DSL.MQI.TRIGGER.DSLR1)
DEFINE PROCESS(DSL.MQI.TRIGGER.DSLR1)    +
      LIKE(DSL.MQI.TRIGGER.DSLR)              +
      APPLICID(DSLR1)                          <= Param. h1  +
      REPLACE

```

DSLKCSQU

The following example shows excerpts from the member DSLKCSQU.

```

*****
* Initiation Queue for the Transmission Queue NTESAATT *
*****
*   If required, modify the following parameter:
*   STGCLASS('SYSTEM')
*
DELETE QLOCAL('DSL.MQI.INITQ.NTESAATT') PURGE
DEFINE QLOCAL('DSL.MQI.INITQ.NTESAATT')    <= Param. e13  +
      DESCR('MERVA-MQI Initiation Queue for NTESAATT')      +
      PUT(ENABLED)                                           +
      DEFPRTY(5)                                             +
      DEFPSIST(YES)                                          +
      GET(ENABLED)                                           +
      SHARE                                                  +
      DEFSOPT(EXCL)                                         +
      MSGDLVSQ(FIFO)                                        +
      RETINTVL(999999999)                                    +
      MAXDEPTH(1000)                                        +
      MAXMSGL(4194304)                                       +
      NOHARDENBO                                           +
      BOTHRESH(0)                                           +
      BOQNAME(' ')                                         +
      STGCLASS('SYSTEM')                                    +
      INDXTYPE(NONE)                                        +
      QDPMAXEV(ENABLED)                                     +
      QDPHIEV(DISABLED)                                     +
      QDEPTHHI(80)                                          +
      QDPLOEV(DISABLED)                                     +
      QDEPTHLO(40)                                          +

```

```

        QSVCIEV(NONE) +
        QSVCINT(999999999) +
        NOTRIGGER +
        TRIGTYPE(NONE) +
        TRIGMPRI(0) +
        TRIGDPH(1) +
        TRIGDATA(' ') +
        PROCESS(' ') +
        INITQ(' ') +
        REPLACE +

*****
* Transmission Queue *
*****
* If required, modify the following parameters:
* QLOCAL('NTESAATT') <= Param. e15
* STGCLASS('REMOTE')
*
DELETE QLOCAL('NTESAATT') PURGE
DEFINE QLOCAL('NTESAATT') <= Param. e15 +
DESCR('Transmission Queue Named Like the Remote QMGR') +
PUT(ENABLED) +
DEFPRTY(5) +
DEFPSIST(YES) +
GET(ENABLED) +
SHARE +
DEFSOPT(EXCL) +
MSGDLVSQ(FIFO) +
RETINTVL(999999999) +
MAXDEPTH(10000) +
MAXMSGL(4194304) +
NOHARDENBO +
BOTHRESH(0) +
BOQNAME(' ') +
STGCLASS('REMOTE') +
USAGE(XMITQ) +
INDXTYPE(NONE) +
QDPMAXEV(ENABLED) +
QDPHIEV(DISABLED) +
QDEPTHHI(80) +
QDPLOEV(DISABLED) +
QDEPTHLO(40) +
QSVCIEV(NONE) +
QSVCINT(999999999) +
TRIGGER +
TRIGTYPE(FIRST) +
TRIGMPRI(0) +
TRIGDPH(1) +
TRIGDATA(' ') +
PROCESS('DSL.MQI.TRIGGER.PROCESS.NTESAATT') +
INITQ('DSL.MQI.INITQ.NTESAATT') <= Param. e13 +
REPLACE

*****
* Sender Channel *
*****
* If required, modify the following parameters:
* CHANNEL('CSQ1.NTESAATT') <= Param. e5
* CONNAME('9.164.171.19') <= Param. e4
* XMITQ('NTESAATT') <= Param. e15
* SEQRAP(999999999) <= Param. e23
*
DELETE CHANNEL('CSQ1.NTESAATT')
DEFINE CHANNEL('CSQ1.NTESAATT') <= Param. e5 +
DESCR('Channel for Sending Messages to NTESAATT') +
CHLTYPE(SDR) +
CONNAME('9.164.171.19') <= Param. e4 +

```

```

TRPTYPE(TCP)
XMITQ('NTESAATT') <= Param. e15
MCAUSER(' ')
BATCHSZ(50)
DISCINT(0)
SHORTRTY(10)
SHORTTMR(60)
LONGRTY(999999999)
LONGTMR(1200)
MSGEXIT('DSLKMEA') <= Param. f9
SCYEXIT(' ')
SCYDATA(' ')
SENDEXIT(' ')
SENDDATA(' ')
RCVEXIT(' ')
RCVDATA(' ')
SEQWRAP(999999999) <= Param. e23
MAXMSGL(4194304)
REPLACE

*****
* Receiver Channel *
*****
* If required, modify the following parameters:
* CHANNEL('NTESAATT.CSQ1') <= Param. e6
* SEQWRAP(999999999) <= Param. e23
*
DELETE CHANNEL('NTESAATT.CSQ1')
DEFINE CHANNEL('NTESAATT.CSQ1') <= Param. e6
DESCR('Channel for Receiving Messages from NTESAATT')
CHLTYPE(RCVR)
TRPTYPE(TCP)
MCAUSER(' ')
BATCHSZ(50)
MSGEXIT('DSLKMEA') <= Param. f9
SCYEXIT(' ')
SCYDATA(' ')
SENDEXIT(' ')
SENDDATA(' ')
RCVEXIT(' ')
RCVDATA(' ')
PUTAUT(DEF)
SEQWRAP(999999999) <= Param. e23
MAXMSGL(4194304)
REPLACE

*****
* Local Definition of the Remote Receive Queue *
*****
* If required, modify the following parameters:
* QREMOTE('DSL.MQI.SEND.USE') <= Param. e9
* RNAME('RCV.FROM.CSQ1') <= Param. e12
* RQMNAME('NTESAATT') <= Param. e2
* XMITQ('NTESAATT') <= Param. e15
*
DELETE QREMOTE('DSL.MQI.SEND.USE')
DEFINE QREMOTE('DSL.MQI.SEND.USE') <= Param. e9
DESCR('Queue for Accessing RCV.FROM.CSQ1 on NTESAATT')
RNAME('RCV.FROM.CSQ1') <= Param. e12
RQMNAME('NTESAATT') <= Param. e2
XMITQ('NTESAATT') <= Param. e15
PUT(ENABLED)
DEFPSIST(YES)
DEFPRTY(9)
REPLACE

```

Samples for Defining MERVA-MQI Attachment Resources

MERVA-MQI Attachment resources are defined in the process table DSLKPROC. A process table consists of one or more send process and receive process entries. For USE, two send processes and one receive process are required. Two sample process tables are provided:

- DSLKPUSE (for MVS)
- DSLKPUSV (for VSE)

In order to be used, a sample process table must be link-edited as DSLKPROC.

Each sample process table contains the following processes:

SPROC1

This process sends session key requests (MT 999) and SWIFT output messages (USE system messages and BKE messages) to the USE workstation as MQSeries datagrams. For each datagram, both a COA and COD report is requested (parameters **f5** and **f6**).

SPROC2

This process sends responses for the USE and BKE SWIFT input messages (that is, SWIFT acknowledgments or error information) to the USE workstation as MQSeries reply messages. For each reply message, both a COA and COD report is requested (parameters **f5** and **f6**).

RPROC1

This process receives session keys and USE or BKE SWIFT input messages from the USE workstation in queue DSL.MQI.RECEIVE (parameter **e10**). Session keys are MQSeries datagrams, USE or BKE SWIFT input messages are MQSeries request messages. COA and COD reports generated by MQSeries on the USE workstation are received in queue DSL.MQI.REPLY_TO_Q (parameter **e17**).

DSLKPUSE (MVS only)

The following example shows the send and receive process entries of table DSLKPUSE.

```

DSLKPROC TYPE=SEND,
    NAME=SPROC1,
    ALLSNDQ=((USESQ,DSL.MQI.SEND.USE)),
    COAWQ=DSLRCOA,
    CODWQ=DSLRCOD,
    ALTUID=remote,
    MQICTLQ=DSL.MQI.CONTROL,
    MRVCTLQ=(USECQS,CONTINUE),
    MRVSTAQ=DSLRCRSTS,
    REPLYTQ=DSL.MQI.REPLY_TO_Q,
    JRNDGRM=YES
DSLKPROC TYPE=SEND,
    NAME=SPROC2,
    ALLSNDQ=((USESQACK,DSL.MQI.SEND.USE)),
    COAWQ=DSLRCOA,
    CODWQ=DSLRCOD,
    ALTUID=remote,
    MQICTLQ=DSL.MQI.CONTROL,
    MRVCTLQ=(USECQS,CONTINUE),
    MRVSTAQ=DSLRCRSTS,
    REPLYTQ=DSL.MQI.REPLY_TO_Q,
    EXIT=8133,
    JRNRPLY=YES
DSLKPROC TYPE=RECEIVE,
    NAME=RPROC1,
    ALTUID=remote,
    MQIRCVQ=(DSL.MQI.RECEIVE,DSL.MQI.REPLY_TO_Q),
    MRVCTLQ=USECQR,
    MRVSTAQ=DSLRCRSTR,
    JRNDGRM=YES,
    JRNRCOA=YES,
    JRNRCOD=YES,
    JRNRQST=YES

```

*
*
* <= Param. f1,e9
* <= Param. f5
* <= Param. f6
* <= Param. e21
* <= Param. e18
* <= Param. f2
* <= Param. f8
* <= Param. e17
* <= Param. f17
*
*
* <= Param. f1,e9
* <= Param. f5
* <= Param. f6
* <= Param. e21
* <= Param. e18
* <= Param. f2
* <= Param. f8
* <= Param. e17
* <= Param. f14
* <= Param. f17
*
*
* <= Param. e21
* <= Param. e10,e17
* <= Param. f2
* <= Param. f8
* <= Param. f17
* <= Param. f17
* <= Param. f17
* <= Param. f17

DSLKPUSV (VSE only)

The following example shows the send and receive process entries of table DSLKPUSV.

```

DSLKPROC TYPE=SEND,
  NAME=SPROC1,
  ALLSNDQ=((USESQ,DSL.MQI.SEND.USE)),
  COAWQ=DSLMRCOA,
  CODWQ=DSLMRCOD,
  MQICTLQ=DSL.MQI.CONTROL,
  MRVCTLQ=(USECQS,CONTINUE),
  MRVSTAQ=DSLMRSTS,
  REPLYTQ=DSL.MQI.REPLY_TO_Q,
  SECURE=AUTHENCR,
  JRNDGRM=YES
DSLKPROC TYPE=SEND,
  NAME=SPROC2,
  ALLSNDQ=((USESQACK,DSL.MQI.SEND.USE)),
  COAWQ=DSLMRCOA,
  CODWQ=DSLMRCOD,
  MQICTLQ=DSL.MQI.CQ,
  MRVCTLQ=(USECQS,CONTINUE),
  MRVSTAQ=DSLMRSTS,
  REPLYTQ=DSL.MQI.REPLY_TO_Q,
  EXIT=8133,
  SECURE=AUTHENCR,
  JRNRPLY=YES
DSLKPROC TYPE=RECEIVE,
  NAME=RPROC1,
  MQIRCVQ=(DSL.MQI.RECEIVE,DSL.MQI.REPLY_TO_Q),
  MRVCTLQ=USECQR,
  MRVSTAQ=DSLMRSTR,
  SECURE=AUTHENCR,
  JRNDGRM=YES,
  JRNRCOA=YES,
  JRNRCOD=YES,
  JRNRQST=YES

```

*
*
* <= Param. f1,e9
* <= Param. f5
* <= Param. f6
* <= Param. e18
* <= Param. f2
* <= Param. f8
* <= Param. e17
* <= Param. f9
* <= Param. f17
*
*
* <= Param. f1,e9
* <= Param. f5
* <= Param. f6
* <= Param. e18
* <= Param. f2
* <= Param. f8
* <= Param. e17
* <= Param. f14
* <= Param. f9
* <= Param. f17
*
*
* <= Param. e10,e17
* <= Param. f2
* <= Param. f8
* <= Param. f9
* <= Param. f17
* <= Param. f17
* <= Param. f17
* <= Param. f17

Samples for Defining MERVA ESA Resources

MERVA ESA resources must be defined for MERVA USE & Branch. These resources are:

- MFS user exit
- Function table entries
- Logical terminal table
- Routing tables

Examples for definitions of these resources are shown in the following sections.

Message Format Service (MFS) Program Table (DSLMPPT)

The following example shows an MFS program table.

```

DSLMPPT DSLMPT TYPE=INITIAL
.
.
.
*-----
*      MERVA-MQI Attachment MFS PROGRAM DEFINITIONS
*-----
DSLMPPT NAME=DSLKQ133,TYPE=U,NUMBER=8133,LINK=NO,LANG=ASM <= Param. f14
.
.
.

```

This table is used to define, among other things, a MERVA-MQI Attachment user exit. All programs are identified by module numbers. By default, the MERVA-MQI Attachment user exits are loaded dynamically from the load library (LINK=NO). The table is used exclusively by the module DSLMMFS to set up the link to MFS user exits.

A sample for MERVA-MQI Attachment user exits is provided in the copy book DSLMPTTC.

The user exit DSLKQ133 is the only user exit that is called by MERVA-MQI Attachment for MERVA USE & Branch. Although written in Assembler, it adheres to the High Level Language (HLL) MFS user exit interface of MERVA ESA (LANG=ASM). That is, it uses MERVA ESA API functions rather than MERVA ESA macros to read from, and write to, the TOF.

Function Table for USE (DSLFNNTT)

All functions necessary for USE are defined in the copy books DSLFNNTTC and DWSFNNTTC.

DSLFNNTTC

The following example shows the functions that are defined in copy book DSLFNNTTC:

```

*          MERVA-TO-MQI START QUEUE
          DSLFNNT NAME=DSLNRSTS,                <= Param. f8  +
          DESCR='Start queue for MERVA-to-MQI send process',  +
          MQI=YES,                               +
          MSGLIM=10,                             +
          QUEUE=DUMMY,                           +
          TRAN=DSLS                               <= Param. g3=h3
*
*          MERVA-TO-MQI CONFIRM-ON-ARRIVAL WAIT QUEUE
          DSLFNNT NAME=DSLNRCOA,                <= Param. f5  +
          DESCR='COA wait queue for MERVA-to-MQI send process',  +
          KEY1=(DSLKMSID,24,,NOMOD),            +
          QUEUE=YES,                             +
          FRAME=(KTOP,0BOT),                    Only for FMT  +
          THRESH=0
*
*          MERVA-TO-MQI CONFIRM-ON-DELIVERY WAIT QUEUE
          DSLFNNT NAME=DSLNRCOD,                <= Param. f6  +
          DESCR='COD wait queue for MERVA-to-MQI send process',  +
          KEY1=(DSLKMSID,24,,NOMOD),            +
          QUEUE=YES,                             +
          FRAME=(KTOP,0BOT),                    Only for FMT  +
          THRESH=0

```

DWSFNNTTC

The following example shows the functions that are defined in copy book DWSFNNTTC:

```

*          MQI ATTACHMENT SEND QUEUES FOR USE MESSAGES
          DSLFNNT NAME=USESQ,                  <= Param. f1  +
          DESCR='Send Queue for USE',          +
          KEY1=(SWBHLT,12),                    +
          MQI=YES,                               +
          MSGLIM=10,                             +
          NOPR=NO,                               +
          PRINT=TX2PR0,                         +
          PROT=YES,                             +
          QUEUE=YES,                             +
          STORE=(SMALL,31900),                 No large messages  +

```

```

TRAN=DSLS                                     <= Param. g3=h3
*
DSL FNT NAME=USESQACK,                         <= Param. f1  +
  DESCR='Send Queue for USE Acknowledgments',  +
  KEY1=(SWBHLT,12),                            +
  MQI=YES,                                     +
  MSG LIM=10,                                  +
  NOPR=NO,                                     +
  PRINT=TX2PR0,                                +
  PROT=YES,                                    +
  QUEUE=YES,                                   +
  STORE=(SMALL,31900),                        No large messages +
  TRAN=DSLS                                     <= Param. g3=h3
*
*
MQI ATTACHMENT CONTROL QUEUES FOR USE MSG PROCESSING
DSL FNT NAME=USECQS,                           <= Param. f2  +
  DESCR='Control Queue for USE (Send process)', +
  DQFILL=NO,                                   +
  KEY1=(DSLKKEY1,16,,NOMOD),                  +
  KEY2=(DSLKMSID,24,,NOMOD),                 +
  QUEUE=YES,                                  +
  ROUTE=EKARTTXU                             <= Param. f16
*
DSL FNT NAME=USECQR,                           <= Param. f2  +
  DESCR='Control Queue for USE (Receive process)', +
  DQFILL=NO,                                   +
  KEY1=(DSLKKEY1,16,,NOMOD),                  +
  KEY2=(DSLKMSID,24,,NOMOD),                 +
  QUEUE=YES,                                  +
  ROUTE=EKARTTXU                             <= Param. f16
*
*
DELETE PROCESSED SWIFT SESSION KEY REQUESTS AND RESPONSES
DSL FNT NAME=USEDEL,                            +
  DESCR='Delete Processed SWIFT USE Messages', +
  QUEUE=DUMMY
*
*
SWIFT USE MESSAGES ERROR QUEUE
DSL FNT NAME=USEERROR,                          +
  DESCR='SWIFT USE Messages Errors',          +
  KEY1=(SWBHLT,12),                            +
  KEY2=(SWAHMT,3),                             +
  QUEUE=YES,                                   +
  SPCMND=DEL,                                  +
  THRESH=10
*
*
AUTHENTICATOR KEY FILE UPDATES RECEIVED FROM THE USE PC
DSL FNT NAME=BKEUPDT,                          +
  CHECK=NO,                                    +
  DESCR='BKE Authenticator Key File Updates', +
  NEXT=USEERROR,                              +
  PROT=YES,                                    +
  QUEUE=YES,                                   +
  STATUS=NOHOLD,                              +
  THRESH=10,                                  +
  TRAN=DWSA
*
*
QUEUE FOR CREATING AND UPDATING BKE PRE-AGREEMENTS
THIS QUEUE IS BOTH DATA ENTRY AND FORMS QUEUE
KEY1 = Transaction Reference Number
KEY2 = Own ID/Corr ID
*
DSL FNT NAME=BKEPRE,                            +
  CHECK=NO,                                    +
  COPY=BKEPRE,                                +
  DE=YES,                                     +
  DESCR='BKE Pre-Agreement Creation and Update', +
  KEY1=(SW20,16),                             +
  KEY2=(SW21,16),                             +

```

```

NEXT=USESQ,                                     <= Param. f1  +
NOPR=YES,                                       +
PRINT=L1PRO,                                    +
QUEUE=YES,                                     +
SPCMND=DEL
*
* RECEIVED SINGLE SWIFT SESSION KEYS
DSLNT NAME=SLSRECV,                             +
  DESCR='Received SWIFT Session Keys',          +
  DQFILL=NO,                                    +
  KEY1=(SWBHLT,12),                             +
  NEXT=USEERROR,                                +
  QUEUE=YES
*
* RECEIVED SWIFT PRELOAD SESSION KEYS
DSLNT NAME=SLSLOAD,                             +
  DESCR='Received SWIFT Preload Session Keys',  +
  DQFILL=NO,                                    +
  KEY1=(SWBHLT,12),                             +
  NEXT=USEERROR,                                +
  QUEUE=YES
*
* PRELOADED GPA SESSION KEYS
* KEY 1 = LTNAME + G, BLANK, ICCPARMS (6), BLANK, LSN/SSN (4)
* KEY 2 = LTNAME + G, BLANK, LSN/SSN (4), BLANK, ICCPARMS (6)
* DSLNT NAME=SLSGPA,                             <= Param. f12 +
  DESCR='Preloaded GPA Session Keys',          +
  DQFILL=NO,                                    +
  KEY1=(,22),                                   +
  KEY2=(,22),                                   +
  MSGSEL=LIST,                                  +
  QUEUE=YES,                                    +
  SPCMND=DEL,                                   Delete with QSN +
  STORE=(SMALL,20),                             +
  TOF=NO                                         No TOF is stored
*
* PRELOADED FIN SESSION KEYS
* KEY 1 = LTNAME + F, BLANK, ICCPARMS (6), BLANK, LSN/SSN (4)
* KEY 2 = LTNAME + F, BLANK, LSN/SSN (4), BLANK, ICCPARMS (6)
* DSLNT NAME=SLSFIN,                             <= Param. f13 +
  DESCR='Preloaded FIN Session Keys',          +
  DQFILL=NO,                                    +
  KEY1=(,22),                                   +
  KEY2=(,22),                                   +
  MSGSEL=LIST,                                  +
  QUEUE=YES,                                    +
  SPCMND=DEL,                                   Delete with QSN +
  STORE=(SMALL,20),                             +
  TOF=NO                                         No TOF is stored
*
* QUEUE TO KEEP LSN/SSN FROM TERMINATION TO NEXT STARTUP
* KEY 1 = LTNAME + G/F
* DSLNT NAME=SLSLTT,                             +
  DESCR='Keep SWIFT LSN/SSN',                  +
  DQFILL=NO,                                    +
  KEY1=(,10),                                   +
  QUEUE=YES,                                    +
  SPCMND=DEL,                                   +
  TOF=NO                                         No TOF is stored

```

Logical Terminal Table for USE (DWSLTT)

The following example shows entries for a logical terminal table for USE (DWSLTT). DWSLTT can be used both for MERVA-MQI Attachment and MERVA Link.

```

LTT      TITLE 'EXAMPLE FOR SWIFT II LOGICAL TERMINAL TABLE'
*
DWSLTT  DWSLT TYPE=MAS,NAME=VNDEBETO,LINE=8,          * <= Param. f18 or b23
        ROUTIN=DWSL1IN,ROUTOUT=DWSL1OUT, RT TABLES *
        READYQ=(L1RGAU),                             *
        TFLAG=SLS,          TECHNOLOGY FLAG          *
        USENAME=USEMERVA2,   USE NAME                * <= Param. f11 or b12
        ROUTSK=DWSRTSK,     SESS KEY REQUEST ROUT    * <= Param. f15 or b17
        SKEYQ=SLSGPA        PRELOADED SESSION KEYS   <= Param. f12 or b13
DWSLT   TYPE=AP,NAME=FIN,
        ROUTIN=DWSL1IN,ROUTOUT=DWSL1OUT, RT TABLES *
        READYQ=(L1RFINU,L1RFINN),   READY QUEUES *
        SKEYQ=SLSFIN,          PRELOADED SESSION KEYS* <= Param. f13 or b14
        DELSUB=(SYSTEM,URGENT,NORMAL)

```

Routing Table for USE (DWSRTSK)

DWSRTSK is a routing table that is used to obtain the LOGIN or SELECT session key when you perform a LOGIN or SELECT request. This routing table must be defined as a CICS program (for example, in the CICS processing program table DFHPPTxx). Its name is assigned to the parameter ROUTSK in the logical terminal table DWSLTT.

DWSRTSK can be used both for MERVA-MQI Attachment and MERVA Link. Parts specific to MERVA-MQI Attachment are marked by the characters @LEI0008.

If MERVA-MQI Attachment is to be used as a backup for MERVA Link, make the changes described in the sections titled **For backup:**. If MERVA Link is to be used as a backup for MERVA-MQI Attachment, no changes are required.

```

RTSK     TITLE 'DWSRTSK - EXAMPLE 1, ROUTING OF SESSION KEY REQUESTS'
*****
*        ROUTE SESSION KEY REQUESTS TO THE MERVA Link OR      @LEI0008
*                                                MQI ATTACHMENT SEND QUEUE @LEI0008
*****
*        DEFINE THE USE PC NAME FROM THE APPLICATION HEADER
DWSRTSK  DSLROUTE TYPE=DEFINE,FIELD=(USERNAME,SWAHILT,,,,VFIRST), *
        LENGTH=9,NOTFND=ERR
*
*        TEST FOR CORRECT USE PC NAME
DSLROUTE TYPE=TEST,COND=(USERNAME,'USEMERVA2',EQ),FALSE=ERR
*
*        ROUTE CORRECT USE PC NAMES
*        For backup: for unconditional MQI Attachment routing, @LEI0008
*        set FOUND=USESQ in the following DSLROUTE @LEI0008
DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,VFIRST), *
        FOUND=TX2USESQ for backup: FOUND=USESQ @LEI0008
USESQ    DSLROUTE TYPE=SET,TARGET=('USESQ'),GOTO=END @LEI0008
*
*        TX2USESQ DSLROUTE TYPE=SET,TARGET=('TX2USESQ'),GOTO=END @LEI0008
*
*        ROUTE INCORRECT USE PC NAMES
ERR      DSLROUTE TYPE=SET,TARGET=('USEERROR'),GOTO=END
*
*        IN CASE OF ROUTING ERRORS THE MESSAGE IS ROUTED TO USEERROR
END      DSLROUTE TYPE=FINAL,TARGET='USEERROR'
        END

```

Routing Table for Telex and USE (EKARTTXU)

Routing table EKARTTXU can be used both for MERVA-MQI Attachment and MERVA Link, and must be defined for CICS (for example, in the CICS processing

program table DFHPPTxx). It is important that the table name is assigned to the ROUTE parameter of the control queues TX2USECQ (for MERVA Link) or USECQR and USECQS (for MERVA-MQI Attachment). These control queues are defined in the copy book DWSFN TTC.

The following shows an example of EKARTTXU for USE (MERVA-MQI Attachment and MERVA Link) and telex (MERVA Link only). Parts specific to MERVA-MQI Attachment are flagged with the characters @LEI0008.

If MERVA-MQI Attachment is to be used as a backup for MERVA Link, make the changes described in the sections titled **For backup**:. If MERVA Link is to be used as a backup for MERVA-MQI Attachment, no changes are required.

```

          TITLE 'MERVA Link SAMPLE ROUTING TABLE EKARTTXU'
*****
*
* Name :          EKARTTXU
*
*****
* Licensed Materials - Property of IBM
* 5648-B29 IBM MERVA for ESA Version 4 Release 1
* (C) Copyright IBM Corp. 1986, 1999
*****
*
* Status = Version 4, Release 1, Modification Level 0
*
* Description : Routing table for MERVA Link Control Queue
*               o TX2USECQ - Workstation based Telex and SWIFT USE
*               o EKAIBZCQ - Workstation based Telex, USE, and IFT
*
*               Also used for the MQI Attachment Control Queue @LEI0008
*               o USECQS - Workstation based SWIFT USE (Send) @LEI0008
*               o USECQR - Workstation based SWIFT USE (Receive)@LEI0008
*
*               Note : The IFT Connection (named IFT Bridge before)
*                   was integrated into MERVA ESA with V3.3.
*                   Therefore EKARTTXU now handles also IFT mes-
*                   sages, which was done before by the IBZ own
*                   routing table IBZRTITU.
*
*****
          SPACE
*****
*
*           C O N T R O L
*
*****
*
*       This part of routing distinguishes between :
*
*       o TELEX           - Go to label TELEX
*       o uncorr. TELEX   - Go to label MCTL
*       o SWIFT USE       - Go to label USE
*       o IFT Connection  - Go to label IFT
*       o OTHER           - Go to label UNKNOWN
*
*****
          SPACE
*       Define the MERVA ESA TELEX message type field
EKARTTXU DSLROUTE TYPE=DEFINE, FIELD=(ENLEXIT,ENLEXIT,,,,VFIRST),
          FOUND=TELEX
*
*       Now it must be SWIFT or uncorrelated TELEX ACK,
*       if not go to UNKNOWN
*       Define the MERVA ESA SWIFT or MERVA Link message type field

```



```

DSLROUTE TYPE=DEFINE,FIELD=(MTYPE,DSLEXIT,,,,,VFIRST),      +
    NOTFND=UNKNOWN
*
*   If uncorrelated TELEX ACK go to MCTL
DSLROUTE TYPE=TEST,COND=(MTYPE,'MCTL',EQ),TRUE=MCTL
*
*   If IFT go to IFT, else go to USE
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZ',EQ,SHORT),TRUE=IFT,      +
    FALSE=USE
EJECT
*****
*
*   T E L E X
*
*****
*
*   This part of routing handles the MERVA Link application
*   for TELEX transmission via PC
*
*   o Outgoing TELEXES are routed to TX2WAIT to await an
*     ACK/NAK.
*   o Incoming TELEXES are routed to TX2RCV, unless poss. dup.
*     indicator is on when they are routed to TX2PDR.
*   o Commands for MERVA/2 TELEX Link are deleted.
*   o Responses to MERVA/2 commands which come back from the
*     PC are routed to TX2TLC. Responses can have type=7,
*     or can be system messages, type=8.
*   o An uncorrelated ACK is routed to TX2INVR - invalid receipt.
*   o Messages routed as a result of the MERVA Link commands
*     'RECOVER' or 'IPRECOV' or as a result of a closed ASP
*     are routed to the TX2HOLD queue and can be manually routed
*     from there to another TELEX application.
*
*****
SPACE
*   Define the MERVA Link application control queue
TELEX DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,,VFIRST),      +
    NOTFND=TXERR
*
*   Define the MERVA Link message class
DSLROUTE TYPE=DEFINE,FIELD=(CLASS,EKACCLASS,,,,,VFIRST),      +
    NOTFND=TXERR
*
*   Define the MERVA Link receipt return code
DSLROUTE TYPE=DEFINE,FIELD=(RECR, EKARECR, , , , , VFIRST)
*
*   Define the TELEX Link station message type
DSLROUTE TYPE=DEFINE,FIELD=(TYPE,ENLP2TYP,,,,,VFIRST)
*
*   Define the last processed queue
DSLROUTE TYPE=DEFINE,FIELD=(PREVQ,MSGTRACE,,,,,VFIRST),      +
    DISP=8,LENGTH=8,NOTFND=TXERR
*
*   Define the message type
DSLROUTE TYPE=DEFINE,FIELD=(MTYPE,DSLEXIT,,,,,VFIRST)
*
*   Check MERVA Link message classes
DSLROUTE TYPE=TEST,COND=(CLASS,'CF',EQ),TRUE=CF
DSLROUTE TYPE=TEST,COND=(CLASS,'LR',EQ),TRUE=LR
DSLROUTE TYPE=TEST,COND=(CLASS,'CA',EQ),TRUE=CHKRC
*   If IP force routing error to keep msg at same pos      @LEI0012
*     and in control queue                                  @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=SETCQ
DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=END
*   If LC route msg back to control queue                  @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=SETCQ      @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=END      @LEI0012

```

```

DSLROUTE TYPE=TEST,COND=(CLASS,'RC',EQ),TRUE=RC
DSLROUTE TYPE=TEST,COND=(CLASS,'RI',EQ),TRUE=RI
DSLROUTE TYPE=TEST,COND=(CLASS,'RS',EQ),TRUE=RS
SPACE
*
Route all incorrect messages
TXERR DSLROUTE TYPE=SET,TARGET='TX2ERROR',GOTO=END
SPACE
*-----
*
Process message class CF
*-----
CF DSLROUTE TYPE=TEST,COND=(PREVQ,'TX2TLC ',EQ),TRUE=SETDQ
*
.. outgoing commands are deleted
DSLROUTE TYPE=TEST,COND=(RECR,'0',EQ,SHORT),TRUE=SETDQ
*
.. ack'd messages are deleted
DSLROUTE TYPE=SET,TARGET='TX2WAIT',GOTO=END
SETDQ DSLROUTE TYPE=SET,TARGET='EKADMY',GOTO=END
SPACE
*-----
*
Process message classes LR, CA, IP, and LC
*-----
*
Route back to control queue after EOM command @LEI0012
LR DSLROUTE TYPE=DEFINE,DISP=32,LENGTH=8,EMPTY=LRR, @LEI0012*
FIELD=(TERM,MSGTRACE,,,,VFIRST,LASTDA) @LEI0012
DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END @LEI0012
*
.. msg type MCTL indicates an uncorrelated ack
LRR DSLROUTE TYPE=TEST,COND=(MTYPE,'MCTL ',EQ),TRUE=TXACKER
*
'LR': last received msg
*
RECR is application rc
*
= '00' remote application has indicated positive ACK
*
= '04' not final ACK: msg stays in ACK wait queue
*
= '08' remote application has indicated negative ACK
*
= ' ' this is not an ACK but an inbound message
CHKRC DSLROUTE TYPE=TEST,COND=(RECR,'00',EQ),TRUE=TXACK
DSLROUTE TYPE=TEST,COND=(RECR,'04',EQ),TRUE=TXWAIT
DSLROUTE TYPE=TEST,COND=(RECR,'08',EQ),TRUE=TXNAK
DSLROUTE TYPE=TEST,COND=(RECR,' ',EQ),TRUE=TXRCV, +
NOTFND=TXRCV
DSLROUTE TYPE=SET,TARGET='TX2ERROR',GOTO=SETCQ
SPACE
TXACKER DSLROUTE TYPE=SET,TARGET='TX2INVR',GOTO=SETCQ
SPACE
TXACK DSLROUTE TYPE=SET,TARGET='TX2ACK',GOTO=CHKCLS
SPACE
TXWAIT DSLROUTE TYPE=SET,TARGET='TX2WAIT',GOTO=CHKCLS
SPACE
TXNAK DSLROUTE TYPE=SET,TARGET='TX2NAK',GOTO=CHKCLS
SPACE
TXRCV DSLROUTE TYPE=TEST,COND=(TYPE,'5',EQ),TRUE=TXRCV2, +
NOTFND=TXRCV2
DSLROUTE TYPE=TEST,COND=(TYPE,'0',EQ),TRUE=TXRCV2, +
NOTFND=TXRCV2 0 is set by Telex Front-end @KPP9810
*
.. other types=command replies etc.
DSLROUTE TYPE=SET,TARGET='TX2TLC',GOTO=SETCQ
*
.. type 5 is a rcvd TELEX
TXRCV2 DSLROUTE TYPE=DEFINE,FIELD=(PDR,ENLP2DUP,,,,VFIRST)
DSLROUTE TYPE=TEST,COND=(PDR,'0',EQ),FALSE=TXPDR
DSLROUTE TYPE=SET,TARGET='TX2RCV',GOTO=SETCQ
TXPDR DSLROUTE TYPE=SET,TARGET='TX2PDR',GOTO=SETCQ
SPACE
CHKCLS DSLROUTE TYPE=TEST,COND=(CLASS,'CA',EQ),TRUE=END
*
FOR CLASS=LR, IP, or LC:
SETCQ DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END
SPACE
*-----
*
Process message classes RC, RI, and RS
*-----
RC DSLROUTE TYPE=SET,TARGET='TX2HOLD',GOTO=END

```

```

RI      DSLROUTE TYPE=SET,TARGET='TX2HOLD',GOTO=END
RS      DSLROUTE TYPE=SET,TARGET='TX2HOLD',GOTO=END
*-----*
EJECT
*****
*
*      M C T L
*
*****
*
*      This part of routing handles uncorrelated TELEX
*
*****
SPACE
*      Define the MERVA Link application control queue.
*      If found go to LR in TELEX, else go to TXERR in TELEX
MCTL   DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,VFIRST),
      FOUND=LR,NOTFND=TXERR
EJECT
*****
*
*      S W I F T   U S E
*
*****
*
*      This part of routing handles all SWIFT USE messages sent to
*      and received from the USE PC:
*
*      o All sent and acknowledged messages are routed
*        to the dummy queue USEDEL where they are deleted.
*        Changing USEDEL to a real queue keeps the messages there.
*        The messages are routed to the MERVA Link or      @LEI0008
*        MQI Attachment send queues by the Routing modules @LEI0008
*        DWSL1IN, DWSL1OUT, DWSL1VE0, and DWSR1TSK.      @LEI0008
*
*      The following messages are received from the USE PC
*      and routed:
*
*      - MT 999 session key response messages to the
*        session key receive queue (TRN is 'SESS KEY REQUEST')
*      - MT 999 pre-generated session key messages to the
*        load session keys queue (TRN is 'SESS KEY PRELOAD')
*      - MT 999 authenticator key file update messages to the
*        queue BKEUPDT (TRN is 'AUTH KEYS UPDATE')
*      - MT 999 pre-agreement messages to the queue BKEPREAG @LEI0008
*        or BKEPRE (TRN is 'BKE PREAGREEMENT') @LEI0008
*      - to the SWIFT link ready queues according to LT name,
*        application id, and priority:
*        -- MT 074 broadcast request for CV revocation
*        -- MT 075 certification request
*        -- MT 085 ICC delivery confirmation
*        -- MT 090 system messages for security administration
*        -- MT 96x user-to-user BKE messages
*      - all errors are routed to the queue USEERROR
*
*****
SPACE
*      If COA or COD report go to STATUS @LEI0008
USE    DSLROUTE TYPE=TEST,COND=(MTYPE,'KRPT',EQ), @LEI0008+
      TRUE=STATUS @LEI0008
*      If Reply message go to STATUS @LEI0008
      DSLROUTE TYPE=TEST,COND=(MTYPE,'KRPL',EQ), @LEI0008+
      TRUE=STATUS @LEI0008
*
*      Define the basic header to see it in the routing trace
*      DSLROUTE TYPE=DEFINE,FIELD=(SWBH,SWBH,,,,VFIRST),
      NOTFND=ERR

```

```

*      Define the application header to see it in the routing trace
DSLROUTE TYPE=DEFINE,FIELD=(SWAH,SWAH,,,,,VFIRST),      +
      NOTFND=ERR

*
*      Define the emitting LT from the basic header
DSLROUTE TYPE=DEFINE,FIELD=(LT,SWBHLT,,,,,VFIRST),      +
      NOTFND=ERR,LENGTH=9

*
*      Define the Merva Link application control queue
*      For backup: for unconditional MQI Attachment routing, @LEI0008
*      uncomment the following DSLROUTE: @LEI0008
***** DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,,VFIRST), +
      FOUND=STATUS @LEI0008
DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,,VFIRST), +
      FOUND=MLMCLS @LEI0008
* @LEI0008
*      Define the MQI Attachment message status @LEI0008
STATUS DSLROUTE TYPE=DEFINE,FIELD=(STATUS,DSLKSTAT,,,,,VFIRST), +
      NOTFND=ERR @LEI0008
*      Message status must be: @LEI0008
*      'RCVD ' for messages received from the USE PC @LEI0008
*      'SENT ' for messages sent to the USE PC @LEI0008
*      'COA ' for messages correlated with a COA report @LEI0008
*      'COAC ' for correlated COA report messages @LEI0008
*      'COD ' for messages correlated with a COD report @LEI0008
*      'CODC ' for correlated COD report messages @LEI0008
*      All other states are considered to be errors @LEI0008
DSLROUTE TYPE=TEST,COND=(STATUS,'RCVD ',EQ),TRUE=RCVD @LEI0008
DSLROUTE TYPE=TEST,COND=(STATUS,'SENT ',EQ),TRUE=CONFIRM @LEI0
DSLROUTE TYPE=TEST,COND=(STATUS,'COA ',EQ),TRUE=CONFIRM @LEI0
DSLROUTE TYPE=TEST,COND=(STATUS,'COAC ',EQ),TRUE=CONFIRM @LEI0
DSLROUTE TYPE=TEST,COND=(STATUS,'COD ',EQ),TRUE=CONFIRM @LEI0
DSLROUTE TYPE=TEST,COND=(STATUS,'CODC ',EQ),TRUE=CONFIRM, +
      FALSE=ERR @LEI0008

*
*      Define the Merva Link message class
MLMCLS DSLROUTE TYPE=DEFINE,FIELD=(CLASS,EKACLASS,,,,,VFIRST), @LEI08+
      NOTFND=ERR
*      Message class must be:
*      'IP' for in-process messages to be sent to the USE PC
*      'LC' for the last confirmed control message
*      'CF' for confirmed messages sent to the USE PC
*      'LR' for messages received from the USE PC
*      All other classes are considered to be errors
*
*      If IP force routing error to keep msg at same pos @LEI0012
*      and in control queue @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=CTLQUEUE
DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=END
*      If LC route msg back to control queue @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=CTLQUEUE @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=END @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'CF',EQ),TRUE=CONFIRM
DSLROUTE TYPE=TEST,COND=(CLASS,'LR',EQ),FALSE=ERR
SPACE

*-----
*      Process message class LR or status RCVD @LEI0008
*-----
*      Route back to control queue after EOM command @LEI0012
DSLROUTE TYPE=DEFINE,DISP=32,LENGTH=8,EMPTY=LRRR, @LEI0012*
      FIELD=(TERM,MSGTRACE,,,,,VFIRST,LASTDA) @LEI0012
DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END @LEI0012
* @LEI0012
*      Provide the last received message for Merva Link integrity
LRRR DSLROUTE TYPE=SET,TARGET=ACQNM @LEI0012
*
*      Define the message type from the application header

```

```

RCVD      DSLROUTE TYPE=DEFINE,FIELD=(MT,SWAHMT,,,,,VFIRST),    @LEI0008+
          NOTFND=ERR
*
*      Test the message type
*      MT 999 is used internally by MERVA USE
*      DSLROUTE TYPE=TEST,COND=(MT,'999',EQ),TRUE=MT999
*      MT 96x are user-to-user BKE messages
*      DSLROUTE TYPE=TEST,COND=(MT,'96',EQ,SHORT),TRUE=READY
*      MT 090 are system messages for security administration
*      DSLROUTE TYPE=TEST,COND=(MT,'090',EQ),TRUE=READY
*      MT 085 is ICC delivery confirmation
*      DSLROUTE TYPE=TEST,COND=(MT,'085',EQ),TRUE=READY
*      MT 075 is certification request
*      DSLROUTE TYPE=TEST,COND=(MT,'075',EQ),TRUE=READY
*      MT 074 is broadcast request for CV revocation
*      DSLROUTE TYPE=TEST,COND=(MT,'074',EQ),TRUE=READY
*
*      Insert test for other message types here
*
*      All messages unknown in this place goto USEERROR
*      DSLROUTE TYPE=SET,TARGET='USEERROR',GOTO=END
*
*      Define the TRN for MT 999
MT999     DSLROUTE TYPE=DEFINE,FIELD=(TRN,SW20,,,,,VFIRST),    +
          NOTFND=ERR
*
*      Test the possible values of the TRN
*      TRN 'SESS KEY REQUEST' indicates a single session key received
*      TRN 'SESS KEY PRELOAD' indicates pregenerated session keys
*      TRN 'BKE PREAGREEMENT' indicates a preagreement for update
*      TRN 'BKE MANUAL START' indicates a manual start of a BKE
*      TRN 'BKE DISCONTINUAT' indicates a BKE discontinuation
*      TRN 'BKE DELETE RECOR' indicates a BKE deletion          @LEI0008
*      TRN 'BKE DISTRIBUTE K' indicates a BKE distribution      @LEI0008
*      TRN 'AUTH KEYS UPDATE' indicates an auth key file update
*      DSLROUTE TYPE=TEST,COND=(TRN,'SESS KEY REQUEST',EQ),TRUE=RECV
*      DSLROUTE TYPE=TEST,COND=(TRN,'SESS KEY PRELOAD',EQ),TRUE=LOAD
*      DSLROUTE TYPE=TEST,COND=(TRN,'BKE PREAGREEMENT',EQ),TRUE=PREAG
*      DSLROUTE TYPE=TEST,COND=(TRN,'BKE MANUAL START',EQ),TRUE=PREAG
*      DSLROUTE TYPE=TEST,COND=(TRN,'BKE DISCONTINUAT',EQ),TRUE=PREAG
*      DSLROUTE TYPE=TEST,COND=(TRN,'BKE DELETE RECOR',EQ),    @LEI0008*
*      TRUE=PREAG                                             @LEI0008
*      DSLROUTE TYPE=TEST,COND=(TRN,'BKE DISTRIBUTE K',EQ),    @LEI0008*
*      TRUE=PREAG                                             @LEI0008
*      DSLROUTE TYPE=TEST,COND=(TRN,'AUTH KEYS UPDATE',EQ),    +
*      TRUE=AUTUP,FALSE=ERR
*
*      Route messages to the respective ready queue
*      Define Appl from the field SWBHAPI
*      (application identifier in the basic header)
READY     DSLROUTE TYPE=DEFINE,FIELD=(APPL,SWBHAPI,,,,,VFIRST),    +
          EMPTY=ERR,NOTFND=ERR
*
*      TEST for F or A. L is not possible here
*      DSLROUTE TYPE=TEST,COND=(APPL,'A',EQ),TRUE=GPA
*      DSLROUTE TYPE=TEST,COND=(APPL,'F',EQ),TRUE=FIN,FALSE=ERR
*      SPACE
*
*      Messages of the GPA application
GPA       DSLROUTE TYPE=TEST,COND=(LT,'VNDEBET',EQ,SHORT),TRUE=GPA1
*      DSLROUTE TYPE=TEST,COND=(LT,'VNDOBET',EQ,SHORT),TRUE=GPA2
*      DSLROUTE TYPE=TEST,COND=(LT,'VNDPBET',EQ,SHORT),TRUE=GPA3,    +
*      FALSE=ERR
*
*      Set the ready queues for GPA
GPA1     DSLROUTE TYPE=SET,TARGET='L1RGPAU',GOTO=END
GPA2     DSLROUTE TYPE=SET,TARGET='L2RGPAU',GOTO=END
GPA3     DSLROUTE TYPE=SET,TARGET='L3RGPAU',GOTO=END

```

```

SPACE
*
* Messages of the FIN application
* Define prty from the field SWAHIPIY (priority of Appl header)
FIN    DSLROUTE TYPE=DEFINE,FIELD=(PRTY,SWAHIPIY,,,,,VFIRST),      +
        EMPTY=ERR,NOTFND=ERR
*
* Test for normal priority N
        DSLROUTE TYPE=TEST,COND=(PRTY,'N',EQ),TRUE=FINN,FALSE=FINU
*
* Normal ready queue for FINancial application
FINN   DSLROUTE TYPE=TEST,COND=(LT,'VNDEBET',EQ,SHORT),TRUE=FINN1
        DSLROUTE TYPE=TEST,COND=(LT,'VNDOBET',EQ,SHORT),TRUE=FINN2
        DSLROUTE TYPE=TEST,COND=(LT,'VNDBPET',EQ,SHORT),TRUE=FINN3,  +
        FALSE=ERR
FINN1  DSLROUTE TYPE=SET,TARGET='L1RFINN',GOTO=END
FINN2  DSLROUTE TYPE=SET,TARGET='L2RFINN',GOTO=END
FINN3  DSLROUTE TYPE=SET,TARGET='L3RFINN',GOTO=END
*
* Urgent ready queue for FINancial application
FINU   DSLROUTE TYPE=TEST,COND=(LT,'VNDEBET',EQ,SHORT),TRUE=FINU1
        DSLROUTE TYPE=TEST,COND=(LT,'VNDOBET',EQ,SHORT),TRUE=FINU2
        DSLROUTE TYPE=TEST,COND=(LT,'VNDBPET',EQ,SHORT),TRUE=FINU3,  +
        FALSE=ERR
FINU1  DSLROUTE TYPE=SET,TARGET='L1RFINU',GOTO=END
FINU2  DSLROUTE TYPE=SET,TARGET='L2RFINU',GOTO=END
FINU3  DSLROUTE TYPE=SET,TARGET='L3RFINU',GOTO=END
SPACE
*
* Route single session keys received from the USE PC
RECV   DSLROUTE TYPE=SET,TARGET='SLSRECV',GOTO=END
*
* Route pregenerated session keys
LOAD   DSLROUTE TYPE=SET,TARGET='SLSLOAD',GOTO=END
*
* Route preagreements to BKEPREAG for display and/or update
* For backup: for unconditional MQI Attachment routing, @LEI0008
* set FOUND=BKEPRE in the following DSLROUTE@LEI0008
PREAG  DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,,VFIRST),      +
        FOUND=BKEPREAG for backup: FOUND=BKEPRE @LEI0008
BKEPRE DSLROUTE TYPE=SET,TARGET='BKEPRE',GOTO=END @LEI0008
BKEPREAG DSLROUTE TYPE=SET,TARGET='BKEPREAG',GOTO=END @LEI0008
*
* Route auth key file updates to the update transaction
AUTUP  DSLROUTE TYPE=SET,TARGET='BKEUPDT',GOTO=END
SPACE
*-----
* Process message class CF or status SENT, COA, COAC, @LEI0008
* COD, or CODC @LEI0008
*-----
* Delete confirmed messages sent to the USE PC
CONFIRM DSLROUTE TYPE=SET,TARGET='USEDEL',GOTO=END
SPACE
*-----
* Process message classes IP and LC
*
* Route in process messages and the LC control message to the
* Merva Link control queue.
* This is only to correct a handling error
*-----
CTLQUEUE DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END
EJECT
*****
*
* I F T
*
*****

```

```

*
* This part of routing handles all IFT messages sent to
* and received from the IFT PC:
*
* o All sent and confirmed messages (message class 'CF')
* are routed to the archive queue IBZARCH where they can
* be deleted.
*
* o The following messages are received from the IFT PC
* and routed:
*
* - MT 'IBZACKN' Acknowledgments
* - MT 'IBZASEQ' A-seq and C-seq of FTMT102
* - MT 'IBZBSEQ' B-seq of FTMT102
* - MT 'IBZDELN' Delivery notifications
* - MT 'IBZIFTE' IFT/2 error messages
* - MT 'IBZINFO' Info messages from the IBZ/2 part
* - MT 'IBZSTAT' Status messages referring to files processed
* - All errors are routed to the queue IBZRE4
*
*****
SPACE
* Define the MERVA Link application control queue.
* If found continue, else go to IBZRE4
IFT DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,VFIRST), +
NOTFND=IBZRE4
* Define the MERVA Link message class
DSLROUTE TYPE=DEFINE,FIELD=(CLASS,EKACCLASS,,,,VFIRST), +
NOTFND=IBZRE4
*
* Message class must be:
* 'IP' for in-process messages to be sent to the IFT PC
* 'LC' for the last confirmed control message
* 'CF' for confirmed messages sent to the IFT PC
* 'LR' for messages received from the IFT PC
* All other classes are considered to be errors
*
* If IP force routing error to keep msg at same pos @LEI0012
* and in control queue @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=CTLQUEUE
DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=END
* If LC route msg back to control queue @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=CTLQUEUE @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=END @LEI0012
**DEL DSLROUTE TYPE=TEST,COND=(CLASS,'CF',EQ),TRUE=CONFIRM
DSLROUTE TYPE=TEST,COND=(CLASS,'CF',EQ),TRUE=IBZARCH
DSLROUTE TYPE=TEST,COND=(CLASS,'LR',EQ),FALSE=IBZRE4
* @LEI0012
* Route back to control queue after EOM command @LEI0012
DSLROUTE TYPE=DEFINE,DISP=32,LENGTH=8,EMPTY=LRRRR, @LEI0012*
FIELD=(TERM,MSGTRACE,,,,VFIRST,LASTDA) @LEI0012
DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END @LEI0012
*
* Provide the last received message for MERVA Link integrity
*
* Test the message type
LRRRR DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZACKN',EQ,SHORT),TRUE=IBZRA3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZASEQ',EQ,SHORT),TRUE=IBZRA3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZBSEQ',EQ,SHORT),TRUE=IBZRB3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZDELN',EQ,SHORT),TRUE=IBZRI3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZIFTE',EQ,SHORT),TRUE=IBZRI3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZINFO',EQ,SHORT),TRUE=IBZRI3
DSLROUTE TYPE=TEST,COND=(MTYPE,'IBZSTAT',EQ,SHORT), +
TRUE=IBZRA3,FALSE=IBZERR
*
* Set the queues for received messages
IBZRA3 DSLROUTE TYPE=SET,TARGET='IBZRA3',GOTO=IBZEXT

```

```

IBZRB3  DSLROUTE TYPE=SET,TARGET='IBZRB3',GOTO=IBZEXT
IBZRI3  DSLROUTE TYPE=SET,TARGET='IBZRI3',GOTO=IBZEXT
IBZERR  DSLROUTE TYPE=SET,TARGET='IBZRE4',GOTO=IBZEXT
IBZEXT  DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END
*
*      Set the queue for sent messages
IBZARCH DSLROUTE TYPE=SET,TARGET='IBZARCH',GOTO=END
*
*      Set the queue for received IBZ messages in error
IBZRE4  DSLROUTE TYPE=SET,TARGET='IBZRE4',GOTO=END
*
      EJECT
*****
*
*      U N K N O W N
*
*****
      SPACE
*      Define the Merva Link application control queue
UNKNOWN DSLROUTE TYPE=DEFINE,FIELD=(ACQNM,EKAACQNM,,,,VFIRST),      +
      NOTFND=ERR
*      Define the Merva Link message class
DSLROUTE TYPE=DEFINE,FIELD=(CLASS,EKACCLASS,,,,VFIRST),      +
      NOTFND=ERR
**DEL   If message class 'LR' fill ACQNM
*
*      If IP force routing error to keep msg at same pos      @LEI0012
*      and in control queue                                  @LEI0012
**DEL   DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=CTLQUEUE
DSLROUTE TYPE=TEST,COND=(CLASS,'IP',EQ),TRUE=END
*      If LC route msg back to control queue                  @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=CTLQUEUE @LEI0012
**DEL   DSLROUTE TYPE=TEST,COND=(CLASS,'LC',EQ),TRUE=END      @LEI0012
DSLROUTE TYPE=TEST,COND=(CLASS,'CF',EQ),TRUE=ERR
DSLROUTE TYPE=TEST,COND=(CLASS,'LR',EQ),FALSE=ERR
*
*      Route back to control queue after EOM command          @LEI0012
DSLROUTE TYPE=DEFINE,DISP=32,LENGTH=8,EMPTY=LRRRRR, @LEI0012*
      FIELD=(TERM,MSGTRACE,,,,VFIRST,LASTDA) @LEI0012
DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=END @LEI0012
LRRRRR  DSLROUTE TYPE=SET,TARGET=ACQNM,GOTO=ERR @LEI0012
      EJECT
*****
*
*      E R R   and   E N D
*
*****
      SPACE
*      Route all incorrect messages
ERR     DSLROUTE TYPE=SET,TARGET='USEERROR',GOTO=END
*
*      Do NOT use the final target with Merva Link
END     DSLROUTE TYPE=FINAL
      END

```

USE Only Scenario

If you previously used only USE with Merva Link (instead of telex and USE together), do the following:

1. In process table DSLKPUSE (for MVS) or DSLKPUSV (for VSE), replace the following Merva ESA queues:
 - Replace send queue USESQ with USESND
 - Replace send queue USESQACK with USESNDACK
 - Replace control queue USECQR with USEMQCTR

- Replace control queue USECQS with USEMQCTS

Assemble and link-edit the process table as DSLKPROC.

2. Use the routing table DWSRTCT instead of EKARTTXU. DWSRTCT is associated to the MERVA ESA control queues USEMQCTR and USEMQCTS. Routing table DWSRTCT can be used both for MERVA-MQI Attachment and MERVA Link. Parts specific to MERVA-MQI Attachment are flagged with the characters @LEI0008.

If MERVA-MQI Attachment is to be used as a backup for MERVA Link, make the changes described in the sections titled **For backup**. If MERVA Link is to be used as a backup for MERVA-MQI Attachment, no changes are required.

3. Replace the MERVA ESA send queue USESQ with USESND in the following routing tables:

- DWSRTSK
- DWSL1OUT
- DWSL2OUT
- DWSL3GPO
- DWSL3FIO

Assemble and link-edit these routing tables.

4. Replace the MERVA ESA send queue USESQACK with USESND in the following routing tables:

- DWSL1IN
- DWSL2IN
- DWSL3GPI
- DWSL3FII

Assemble and link-edit these tables.

5. Replace the MERVA ESA send queue USESQ with USESND in the copy book DWSFNNTC. This applies to the parameter NEXT of the function BKEPRE.
6. Generate the MERVA ESA function table DSLFNNT, then assemble and link-edit it. Use the copy book DWSFNNTC. The appropriate entries are shown in the example below.

```

*      MQI ATTACHMENT SEND QUEUES FOR USE MESSAGES
      DSLFNT NAME=USESND,
      DESCR='Send USE Messages to the USE PC',
      KEY1=(SWBHLT,12),
      MQI=YES,
      MSGLIM=10,
      QUEUE=YES,
      STORE=(SMALL,31900),      No large messages
      TRAN=DSL5
*
      DSLFNT NAME=USESND,
      DESCR='Send USE Acknowledgments to the USE PC',
      KEY1=(SWBHLT,12),
      MQI=YES,
      MSGLIM=10,
      QUEUE=YES,
      STORE=(SMALL,31900),      No large messages
      TRAN=DSL5
*
*      MQI ATTACHMENT CONTROL QUEUES FOR USE MSG PROCESSING
      DSLFNT NAME=USEMQCTS,
      DESCR='SWIFT Session Key Control Queue (Send)',
      DQFILL=UC,
      KEY1=(DSLKKEY1,16,,NOMOD),

```

```

KEY2=(DSLKMSID,24,,NOMOD),      +
QUEUE=YES,                      +
ROUTE=DWSRTCT
*
DSL FNT NAME=USEMQCTR,          +
DESCR='SWIFT Session Key Control Queue (Receive)', +
DQFILL=UC,                     +
KEY1=(DSLKKEY1,16,,NOMOD),     +
KEY2=(DSLKMSID,24,,NOMOD),     +
QUEUE=YES,                      +
ROUTE=DWSRTCT
*
*   QUEUE FOR CREATING AND UPDATING BKE PRE-AGREEMENTS
*   THIS QUEUE IS BOTH DATA ENTRY AND FORMS QUEUE
*   KEY1 = Transaction Reference Number
*   KEY2 = Own ID/Corr ID
DSL FNT NAME=BKEPRE,           +
CHECK=NO,                     +
COPY=BKEPRE,                  +
DE=YES,                       +
DESCR='BKE Pre-Agreement Creation and Update', +
KEY1=(SW20,16),               +
KEY2=(SW21,16),               +
NEXT=USESND,                  +
NOPR=YES,                      +
PRINT=L1PRO,                  +
QUEUE=YES,                     +
SPCMND=DEL

```

Sample Startup Jobs for MERVA-MQI Attachment

The way that MERVA-MQI Attachment is started depends on whether it is running under CICS or IMS.

Sample CICS Startup Job

Under CICS, MERVA-MQI Attachment is started together with the rest of MERVA ESA. The following example shows excerpts from a sample startup job for CICS/ESA Version 4.1 for use with MERVA-MQI Attachment. See the MERVA ESA Program Directory for more information on how to modify CICS/ESA in order to set up a connection between CICS/ESA and MQSeries.

```

//CICSESA JOB (Account),Name,....
.
.
.
//CICS EXEC PGM=DFHSIP,REGION=8M,PARM=SYSIN
/*
//STEPLIB DD DSN=CICS410.SDFHAUTH,DISP=SHR
// DD DSN=MQSERIES.SCSQANLE,DISP=SHR
// DD DSN=MQSERIES.SCSQAUTH,DISP=SHR
//DFHRPL DD DSN=MERVA.SDSLLODC,DISP=SHR
// DD DSN=MERVA.SDSLLODB,DISP=SHR
// DD DSN=CICS410.SDFHLOAD,DISP=SHR
// DD DSN=MQSERIES.SCSQANLE,DISP=SHR
// DD DSN=MQSERIES.SCSQCICS,DISP=SHR
// DD DSN=MQSERIES.SCSQAUTH,DISP=SHR
.
.
.
.. more DD-statements ..
.
.
.
//*-----
//* CICS Startup Parameters
//*-----
//SYSIN DD *
SIT=6$,
GRPLIST=(DFHLIST,MLIST,MQILIST),
INITPARM=(CSQCPARM='SN=CSQ1,TN=001,IQ=CICS01.INITQ'), <= Param. e13
.
.
.
.END
//

```

Sample IMS MPP Job

Under IMS, MERVA-MQI Attachment is started using an MPP job. The following example shows excerpts from a sample MPP job for IMS Version 6.1. See the MERVA ESA Program Directory for more information on how to modify IMS in order to set up a connection between IMS and MQSeries.

```

//IMS      JOB (Account),Name,....
          .
          .
          .
//MERVAMPP PROC DOUT=X,SOUT='*',RGN=4096K,
//          CL1=038,CL2=039,CL3=000,CL4=000,
//          OPT=W,OVLA=0,SPIE=0,VALCK=1,TLIM=07,
//          PCB=002,PRLD=DS,STIMER=,SOD=X,DBLDL=10,
//          NBA=,OBA=,IMSID=I610,
//          SSM=DSN
//*
//REGION  EXEC PGM=DFSRR00,REGION=&RGN,
//          TIME=1440,DPRTY=(12,0),
//          PARM=(MSG,&CL1&CL2&CL3&CL4,
//          &OPT&OVLA&SPIE&VALCK&TLIM&PCB,
//          &PRLD,&STIMER,&SOD,&DBLDL,&NBA,
//          &OBA,&IMSID,&SSM)
//STEPLIB DD DSN=MERVA.SDSLLODI,DISP=SHR
//          DD DSN=MERVA.SDSLLODB,DISP=SHR
//          DD DSN=IMS610.RESLIB,DISP=SHR
//          DD DSN=MQSERIES.SCSQANLE,DISP=SHR
//          DD DSN=MQSERIES.SCSQAUTH,DISP=SHR
//DFSESL  DD DSN=IMS610.RESLIB,DISP=SHR
//          DD DSN=MQSERIES.SCSQANLE,DISP=SHR
//          DD DSN=MQSERIES.SCSQAUTH,DISP=SHR
//          DD DSN=MQSERIES.USERLIB(CSQQDEFV),DISP=SHR
//PROCLIB DD DSN=IMS610.PROCLIB,DISP=SHR
          .....
          .. more DD-statements ..
          .....
//          PEND
//S1      EXEC MERVAMPP

```

Notes:

1. MERVA-MQI Attachment does not require that the BMP job for the MERVA ESA nucleus be modified.
2. If the module CSQQDEFV is contained in the MQSeries library with the low level qualifier SCSQAUTH and you want to use it from this library, omit the DD statement for the USERLIB in the DFSESL concatenation.

Appendix C. Samples for Customizing USE Functions for MERVA-MQI Attachment on the Workstation

MERVA USE & Branch provides samples of files that you can use to define MQSeries resources and MERVA-MQI Attachment resources.

Samples for Defining MQSeries Resources under Windows NT

Under Windows NT, use the MQSeries batch utility RUNMQSC to define MQSeries resources such as queues and channels. The definitions can be made in a file, for example in NTMVSATT.MQ (for MVS) or NTVSEATT.MQ (for VSE). Examples of these files are shown in "Samples for Defining MQSeries Resources for a Connection from Windows NT to MVS" and "Samples for Defining MQSeries Resources for a Connection from Windows NT to VSE" on page 197.

Samples for Defining MQSeries Resources for a Connection from Windows NT to MVS

NTMVSATT.MQ is a sample file that defines all MQSeries objects needed by MERVA-MQI Attachment for use with MVS. For more information refer to the *MQSeries MQSC Command Reference*.

```
*-----*
* File name:   NTMVSATT.MQ
*
* Description: Configuration script to define all
*              MQSeries objects needed for USE over MQ (using MQIA) for MVS
*-----*
* Licensed materials - Property of IBM
*
* MERVA ESA Components V4 (5648-B30)
*
* (C) Copyright IBM Corp. 1999, 2001
*
* All rights reserved.
*
* US Government Users Restricted Rights - Use, duplication or disclosure
* restricted by GSA ADP Schedule Contract with IBM Corp.
*-----*

*****
* >>>> COMMON RESET SECTION
*
DELETE QLOCAL(DLQ)           PURGE
DELETE QLOCAL(DXQ)           PURGE
DELETE QLOCAL(REFQ)          PURGE                               <= Param. e19
DELETE QLOCAL(TRIGQ)         PURGE                               <= Param. e14
DELETE PROCESS(ENMMQAT)
*
* END COMMON RESET SECTION
*****

*****
* >>>> LINK SPECIFIC RESET SECTION   for Link No. 01 <<<<
*
STOP  CHANNEL(CSQ1.NTESAATT)      MODE(FORCE)           <= Param. e8
DELETE CHANNEL(CSQ1.NTESAATT)     <= Param. e8
STOP  CHANNEL(NTESAATT.CSQ1)      MODE(FORCE)           <= Param. e7
```

```

DELETE CHANNEL(NTESAATT.CSQ1)                <= Param. e7
DELETE QREMOTE(SND.TO.CSQ1)                  <= Param. e11
DELETE QLOCAL(CSQ1)                          PURGE    <= Param. e16
DELETE QLOCAL(RCV.FROM.CSQ1)                 PURGE    <= Param. e12
*
* END LINK SPECIFIC RESET SECTION
*****
*****
* >>>> COMMON DEFINITION SECTION <<<<
* common elements.
*
DEFINE QLOCAL(DLQ)                            +
  REPLACE                                     +
  DEFPRTY(0)                                  +
  DEFPSIST(YES)                               +
  DESCR('NTESAATT Dead-Letter queue')        +
  PUT(ENABLED)                                +
  DEFSOPT(SHARED)                             +
  GET(ENABLED)                                 +
  MAXDEPTH(99999)                             +
  MAXMSGL(31000)                              +
  MSGDLVSQ(PRIORITY)                          +
  SHARE                                         +
  NOTRIGGER                                    +
  QDEPTHHI(0)                                  +
  QDEPTHLO(100)                               +
  QDPHIEV(DISABLED)                           +
  QDPLOEV(DISABLED)                           +
  QDPMAXEV(DISABLED)                          +
  QSVCIEV(NONE)                               +
  SCOPE(QMGR)                                  +
  USAGE(NORMAL)                               +

DEFINE QLOCAL(REFQ)                          + <= Param. e19
  REPLACE                                     +
  DEFPRTY(0)                                  +
  DEFPSIST(YES)                               +
  DESCR('NTESAATT Reference queue')          +
  PUT(ENABLED)                                +
  DEFSOPT(SHARED)                             +
  GET(ENABLED)                                 +
  MAXDEPTH(99999)                             +
  MAXMSGL(31000)                              +
  MSGDLVSQ(PRIORITY)                          +
  SHARE                                         +
  NOTRIGGER                                    +
  QDEPTHHI(0)                                  +
  QDEPTHLO(100)                               +
  QDPHIEV(DISABLED)                           +
  QDPLOEV(DISABLED)                           +
  QDPMAXEV(DISABLED)                          +
  QSVCIEV(NONE)                               +
  SCOPE(QMGR)                                  +
  USAGE(NORMAL)                               +

DEFINE QLOCAL(TRIGQ)                          + <= Param. e14
  REPLACE                                     +
  DEFPRTY(0)                                  +
  DEFPSIST(NO)                                +
  DESCR('NTESAATT Initiation queue')        +
  PUT(ENABLED)                                +
  DEFSOPT(SHARED)                             +
  GET(ENABLED)                                 +
  MAXDEPTH(99999)                             +
  MAXMSGL(1000)                               +
  MSGDLVSQ(PRIORITY)                          +

```

```

SHARE +
NOTRIGGER +
QDEPTHHI(0) +
QDEPTHLO(100) +
QDPHIEV(DISABLED) +
QDPLOEV(DISABLED) +
QDPMAEV(DISABLED) +
QSVCI EV(NONE) +
SCOPE(QMGR) +
USAGE(NORMAL)

DEFINE QLOCAL(DXQ) +
REPLACE +
DEFPRTY(0) +
DEFPSIST(YES) +
DESCR('NTESAATT Default transmission queue') +
PUT(ENABLED) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(99999) +
MAXMSGL(31000) +
MSGDLVSQ(PRIORITY) +
SHARE +
NOTRIGGER +
QDEPTHHI(0) +
QDEPTHLO(100) +
QDPHIEV(DISABLED) +
QDPLOEV(DISABLED) +
QDPMAEV(DISABLED) +
QSVCI EV(NONE) +
SCOPE(QMGR) +
USAGE(XMITQ)

*
DEFINE PROCESS(ENMMQAT) +
REPLACE +
* +++ User definition follows : complete path to application ...
  APPLICID('c:\merva\USE_Branch\bin\cmemqat.exe') +
* --- End of User definition.
  APPLTYPE(WindowsNT) +
  DESCR('ENMMQAT application: CMEMQAT')

*
*
* END COMMON DEFINITION SECTION
*****

*****
* >>>> LINK SPECIFIC DEFINITION SECTION for Link No. 01 <<<<
* Used to connect NTESAATT with CSQ1.
*
DEFINE CHANNEL(CSQ1.NTESAATT) + <= Param. e8
  CHLTYPE(RCVR) +
  TRPTYPE(TCP) +
  DESCR('NTESAATT RCV Channel from CSQ1') +
  REPLACE +
  MAXMSGL(4194304) +
* +++ User definition follows : message channel exit and remote system type ...
  MSGEXIT('c:\merva\USE_Branch\bin\cmemqat.dll(MQAMsgExit)') + <= Param. f10
* --- End of User definition.
  SEQWRAP(999999999) + <= Param. e23
  MRRTY(3) +
  MRTMR(30)

DEFINE CHANNEL(NTESAATT.CSQ1) + <= Param. e7
  CHLTYPE(SDR) +
* +++ User definition follows : remote partners TCP/IP address ...
  CONNAME('1.123.123.0(1414)') + <= Param. e3
* --- End of User definition.

```

```

|         TRPTYPE(TCP)                                     +
| * +++ User definition follows :
| *   local transmit q ( named as remote queue manager ! )
|   XMITQ(CSQ1)                                           + <= Param. e16
| * --- End of User definition.
|   CONVERT(NO)                                           +
|   DESCR('NTESAATT send CHANNEL to CSQ1')              +
|   REPLACE                                               +
|   DISCINT(0)                                            +
|   LONGRTY(999999)                                       +
|   LONGTMR(30)                                           +
|   MAXMSGL(4194304)                                       +
| * +++ User definition follows : message channel exit and remote system type ...
|   MSGEXIT('c:\merva\USE_Branch\bin\cmenemqe.dll(MQAMsgExit)') + <= Param. f10
| * --- End of User definition.
|   SEQWRAP(999999999)                                     + <= Param. e23
|   SHORTRTY(3)                                           +
|   SHORTTMR(3)
|
| DEFINE QLOCAL(RCV.FROM.CSQ1)                             + <= Param. e12
|   REPLACE                                               +
|   DEFPRTY(0)                                           +
|   DEFPSIST(YES)                                         +
|   DESCR('NTESAATT Receive Queue from CSQ1')           +
|   PUT(ENABLED)                                         +
|   DEFSOPT(SHARED)                                       +
|   GET(ENABLED)                                         +
|   INITQ(TRIGQ)                                         +
|   MAXDEPTH(99999)                                       +
|   MAXMSGL(32000)                                       +
|   MSGDLVSQ(PRIORITY)                                    +
|   SHARE                                                 +
|   TRIGGER                                               +
|   PROCESS(ENMMQAT)                                      +
|   QDEPTHHI(0)                                          +
|   QDEPTHLO(100)                                        +
|   QDPHIEV(DISABLED)                                    +
|   QDPLOEV(DISABLED)                                    +
|   QDPMAEV(DISABLED)                                    +
|   QSVCI EV(NONE)                                       +
|   SCOPE(QMGR)                                          +
|   TRIGDATA('RCV from remote system')                  +
|   TRIGDPTH(1)                                          +
|   TRIGMPRI(0)                                          +
|   TRIGTYPE(EVERY)                                      +
|   USAGE(NORMAL)
|
| * +++ User definition follows :
| *   this local transmission queue must be named like
| *   the remote queue manager !
| DEFINE QLOCAL(CSQ1)                                     + <= Param. e16
| * --- End of User definition.
|   REPLACE                                               +
|   DEFPRTY(0)                                           +
|   DEFPSIST(YES)                                         +
|   DESCR('NTESAATT Local transmit queue to CSQ1')      +
|   PUT(ENABLED)                                         +
|   DEFSOPT(SHARED)                                       +
|   GET(ENABLED)                                         +
|   MAXDEPTH(99999)                                       +
|   MAXMSGL(31000)                                       +
|   MSGDLVSQ(PRIORITY)                                    +
|   SHARE                                                 +
|   NOTRIGGER                                             +
|   QDEPTHHI(0)                                          +
|   QDEPTHLO(100)                                        +
|   QDPHIEV(DISABLED)                                    +

```



```

        QDPL0EV(DISABLED)          +
        QDPMAXEV(DISABLED)        +
        QSVCI0V(NONE)              +
        SCOPE(QMGR)                +
        USAGE(XMITQ)

DEFINE QREMOTE(SND.TO.CSQ1)      + <= Param. e11
    REPLACE                       +
    DEFPRTY(0)                    +
    DEFPSIST(YES)                 +
    DESCR('NTESAATT Send queue to CSQ1')
    PUT(ENABLED)                  +
* +++ User definition follows :
*   remote receive queue name,
*   remote queue manager name,
*   local transmission queue name
*   (named like the remote queue manager!)
    RNAME(DSL.MQI.RECEIVE)        + <= Param. e10
    RQMNAME(CSQ1)                 + <= Param. e1
    XMITQ(CSQ1)                   + <= Param. e16
* --- End of User definition.
    SCOPE(QMGR)

*
* END LINK SPECIFIC DEFINITION SECTION   for Link No. 01
*****

*****
* >>>> STARTING LISTENER SECTION <<<<
*
STA LISTENER TRPTYPE(TCP)
*
* END LISTENER SECTION
*****

*****
* >>>> STARTING CHANNEL SECTION <<<<
*
STA CHL(NTESAATT.CSQ1)           <= Param. e7
STA CHL(CSQ1.NTESAATT)          <= Param. e8
*
* END STARTING CHANNEL SECTION
*****

```

Samples for Defining MQSeries Resources for a Connection from Windows NT to VSE

NTVSEATT.MQ is a sample file that defines all the MQSeries objects needed by MERVA-MQI Attachment for use with VSE. For more information refer to the *MQSeries MQSC Command Reference*.

```

*-----*
* File name:   NTVSEATT.MQ
*
* Description: Configuration script to define all
*              MQSeries objects needed for USE over MQ (using MQIA) for VSE
*-----*
* Licensed materials - Property of IBM
*
* MERVA ESA Components V4 (5648-B30)
*
* (C) Copyright IBM Corp. 1999, 2001
*
* All rights reserved.
*
* US Government Users Restricted Rights - Use, duplication or disclosure
* restricted by GSA ADP Schedule Contract with IBM Corp.
*

```

```

*-----*
*****
* >>>> COMMON RESET SECTION
*
DELETE QLOCAL(DLQ)          PURGE
DELETE QLOCAL(DXQ)          PURGE
DELETE QLOCAL(REFQ)         PURGE                    <= Param. e19
DELETE QLOCAL(TRIGQ)        PURGE                    <= Param. e14
DELETE PROCESS(ENMMQAT)
*
* END COMMON RESET SECTION
*****

*****
* >>>> LINK SPECIFIC RESET SECTION   for Link No. 01 <<<<
*
*
STOP  CHANNEL(DVSCQMGR.NTESAATT)          MODE(FORCE)          <= Param. e8
DELETE CHANNEL(DVSCQMGR.NTESAATT)         <= Param. e8
STOP  CHANNEL(NTESAATT.DVSCQMGR)          MODE(FORCE)          <= Param. e7
DELETE CHANNEL(NTESAATT.DVSCQMGR)         <= Param. e7
DELETE QREMOTE(SND.TO.DVSCQMGR)           <= Param. e11
DELETE QLOCAL(DVSCQMGR)                   PURGE                <= Param. e16
DELETE QLOCAL(RCV.FROM.DVSCQMGR)          PURGE                <= Param. e12
*
* END LINK SPECIFIC RESET SECTION
*****

*****
* >>>> COMMON DEFINITION SECTION <<<<
* common elements.
*
DEFINE QLOCAL(DLQ)          +
REPLACE                    +
DEFPRTY(0)                 +
DEFPSIST(YES)              +
DESCR('NTESAATT Dead-Letter queue')    +
PUT(ENABLED)               +
DEFSOPT(SHARED)            +
GET(ENABLED)               +
MAXDEPTH(99999)            +
MAXMSGL(31000)             +
MSGDLVSQ(PRIORITY)        +
SHARE                      +
NOTRIGGER                  +
QDEPTHHI(0)                +
QDEPTHLO(100)              +
QDPHIEV(DISABLED)         +
QDPLOEV(DISABLED)         +
QDPMAXEV(DISABLED)        +
QSVCI EV(NONE)             +
SCOPE(QMGR)                +
USAGE(NORMAL)              +

DEFINE QLOCAL(REFQ)        + <= Param. e19
REPLACE                    +
DEFPRTY(0)                 +
DEFPSIST(YES)              +
DESCR('NTESAATT Reference queue')      +
PUT(ENABLED)               +
DEFSOPT(SHARED)            +
GET(ENABLED)               +
MAXDEPTH(99999)            +
MAXMSGL(31000)             +
MSGDLVSQ(PRIORITY)        +
SHARE                      +

```

```

NOTRIGGER +
QDEPTHHI(0) +
QDEPTHLO(100) +
QDPHIEV(DISABLED) +
QDPLOEV(DISABLED) +
QDPMAXEV(DISABLED) +
QSVCIIEV(NONE) +
SCOPE(QMGR) +
USAGE(NORMAL)

DEFINE QLOCAL(TRIGQ) + <= Param. e14
REPLACE +
DEFPRTY(0) +
DEFPSIST(NO) +
DESCR('NTESAATT Initiation queue') +
PUT(ENABLED) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(99999) +
MAXMSGL(1000) +
MSGDLVSQ(PRIORITY) +
SHARE +
NOTRIGGER +
QDEPTHHI(0) +
QDEPTHLO(100) +
QDPHIEV(DISABLED) +
QDPLOEV(DISABLED) +
QDPMAXEV(DISABLED) +
QSVCIIEV(NONE) +
SCOPE(QMGR) +
USAGE(NORMAL)

DEFINE QLOCAL(DXQ) +
REPLACE +
DEFPRTY(0) +
DEFPSIST(YES) +
DESCR('NTESAATT Default transmission queue') +
PUT(ENABLED) +
DEFSOPT(SHARED) +
GET(ENABLED) +
MAXDEPTH(99999) +
MAXMSGL(31000) +
MSGDLVSQ(PRIORITY) +
SHARE +
NOTRIGGER +
QDEPTHHI(0) +
QDEPTHLO(100) +
QDPHIEV(DISABLED) +
QDPLOEV(DISABLED) +
QDPMAXEV(DISABLED) +
QSVCIIEV(NONE) +
SCOPE(QMGR) +
USAGE(XMITQ)

*
DEFINE PROCESS(ENMMQAT) +
REPLACE +
* +++ User definition follows : complete path to application ...
  APPLICID('c:\merva\USE_Branch\bin\cmemqat.exe') +
* --- End of User definition.
  APPLTYPE(WindowsNT) +
  DESCR('ENMMQAT application: CMEMQAT')
*
*
* END COMMON DEFINITION SECTION
*****
*****

```

```

* >>>> LINK SPECIFIC DEFINITION SECTION   for Link No. 01 <<<<
* Used to connect NTESAATT with DVSCQMGR.
*
DEFINE CHANNEL(DVSCQMGR.NTESAATT)          + <= Param. e8
      CHLTYPE(RCVR)                        +
      TRPTYPE(TCP)                         +
      DESCR('NTESAATT RCV Channel from DVSCQMGR') +
      REPLACE                              +
      MAXMSGL(4194304)                    +
* +++ User definition follows : message channel exit and remote system type ...
      MSGEXIT('c:\merva\USE_Branch\bin\cmenemqe.dll(MQAMsgExit)') + <= Param. f10
* --- End of User definition.
      SEQWRAP(999999)                     + <= Param. e23
      MRRTY(3)                             +
      MRTMR(30)
*
DEFINE CHANNEL(NTESAATT.DVSCQMGR)          + <= Param. e7
      CHLTYPE(SDR)                        +
* +++ User definition follows : remote partners TCP/IP address ...
      CONNAME('1.123.123.0(1414)')        + <= Param. e3
* --- End of User definition.
      TRPTYPE(TCP)                        +
* +++ User definition follows :
*   local transmit q ( named as remote queue manager ! )
      XMITQ(DVSCQMGR)                     + <= Param. e16
* --- End of User definition.
      CONVERT(NO)                         +
      DESCR('NTESAATT send CHANNEL to DVSCQMGR') +
      REPLACE                              +
      DISCINT(0)                          +
      LONGRTY(999999)                     +
      LONGTMR(30)                         +
      MAXMSGL(4194304)                    +
* +++ User definition follows : message channel exit and remote system type ...
      MSGEXIT('c:\merva\USE_Branch\bin\cmenemqe.dll(MQAMsgExit)') +
* --- End of User definition.
      SEQWRAP(999999)                     + <= Param. e23
      SHORTRTY(3)                         +
      SHORTTMR(3)
*
DEFINE QLOCAL(RCV.FROM.DVSCQMGR)          + <= Param. e12
      REPLACE                              +
      DEFPRTY(0)                          +
      DEFPSIST(YES)                       +
      DESCR('NTESAATT Receive Queue from DVSCQMGR') +
      PUT(ENABLED)                        +
      DEFSOPT(SHARED)                     +
      GET(ENABLED)                        +
      INITQ(TRIGQ)                        +
      MAXDEPTH(99999)                     +
      MAXMSGL(32000)                      +
      MSGDLVSQ(PRIORITY)                  +
      SHARE                                +
      TRIGGER                              +
      PROCESS(ENMMQAT)                    +
      QDEPTHHI(0)                         +
      QDEPTHLO(100)                       +
      QDPHIEV(DISABLED)                   +
      QDPLOEV(DISABLED)                   +
      QDPMAXEV(DISABLED)                  +
      QSVCIEV(NONE)                       +
      SCOPE(QMGR)                         +
      TRIGDATA('RCV from remote system')  +
      TRIGDPH(1)                          +
      TRIGMPRI(0)                         +
      TRIGTYPE(EVERY)                     +
      USAGE(NORMAL)

```

```

* +++ User definition follows :
*   this local transmission queue must be named like
*   the remote queue manager !
DEFINE QLOCAL(DVSCQMGR)                                + <= Param. e16
* --- End of User definition.
    REPLACE                                           +
    DEFPRTY(0)                                         +
    DEFPSIST(YES)                                       +
    DESCR('NTESAATT Local transmit queue to DVSCQMGR') +
    PUT(ENABLED)                                       +
    DEFSOPT(SHARED)                                    +
    GET(ENABLED)                                       +
    MAXDEPTH(99999)                                    +
    MAXMSGL(31000)                                    +
    MSGDLVSQ(PRIORITY)                                +
    SHARE                                              +
    NOTRIGGER                                          +
    QDEPTHHI(0)                                       +
    QDEPTHLO(100)                                     +
    QDPHIEV(DISABLED)                                 +
    QDPLOEV(DISABLED)                                 +
    QDPMAXEV(DISABLED)                               +
    QSVCI EV(NONE)                                     +
    SCOPE(QMGR)                                       +
    USAGE(XMITQ)

DEFINE QREMOTE(SND.TO.DVSCQMGR)                       + <= Param. e11
    REPLACE                                           +
    DEFPRTY(0)                                         +
    DEFPSIST(YES)                                       +
    DESCR('NTESAATT Send queue to DVSCQMGR')         +
    PUT(ENABLED)                                       +

* +++ User definition follows :
*   remote receive queue name,
*   remote queue manager name,
*   local transmission queue name
*   (named like the remote queue manager!)
    RNAME(DSL.MQI.RECEIVE)                             + <= Param. e10
    RQMNAME(DVSCQMGR)                                  + <= Param. e1
    XMITQ(DVSCQMGR)                                    + <= Param. e16
* --- End of User definition.
    SCOPE(QMGR)

*
* END LINK SPECIFIC DEFINITION SECTION   for Link No. 01
*****

*****
* >>>> STARTING LISTENER SECTION <<<<
*
STA LISTENER TRPTYPE(TCP)
*
* END LISTENER SECTION
*****

*****
* >>>> STARTING CHANNEL SECTION <<<<
*
STA CHL(NTESAATT.DVSCQMGR)                            <= Param. e7
STA CHL(DVSCQMGR.NTESAATT)                            <= Param. e8
*
* END STARTING CHANNEL SECTION
*****

```

Samples for Defining MERVA-MQI Attachment Resources

Which sample configuration file you use depends on whether you are connecting to MVS or VSE.

Sample Configuration File for a Connection to MVS

The sample configuration file NTMVSATT.ATN contains the common section (MVQAttachment) and one link-specific section (MVQLink1) for a link to an MVS system.

```
#-----
# File Name:   NTMVSATT.ATN
#
# Description: MQI Attachment configuration file
#              (1) => to be customized in MERVA
#              (2) => to be configured in MQSeries
#-----
# Licensed materials - Property of IBM
#
# MERVA ESA Components V4 (5648-B30)
#
# (C) Copyright IBM Corp. 1999, 2001
#
# All rights reserved.
#
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#-----

# Common Section
[MVQAttachment]
  MVQName       = remote           # Parameter e21
  MRVName       = MervaMQ         #
  MRVUser       = merva1         # Parameter e22
# MRVPwd       =                  #
  MQQMgrName    = NTESAATT       # Parameter e2
  MQRefQName    = REFQ           # Parameter e19
# MQTActQName  = TACTQ          # Parameter e20
  MQTrigQName   = TRIGQ         #
  ScanInterval  = 60             #
  LogDir        = c:\temp        #
# MQPIIRecovery = Yes           #
# MVPIIRecovery = Yes           #
# APITrace     = No             #
# MQClient     = No             #
# LogLevel     = 3              #

# First Link Section
[MVQLink1]
# LkName       = MVQLK1         #
  RmtUser      = remote         # Parameter e21
  MQRcvQName   = RCV.FROM.CSQ1  # Parameter e12
  MQSendQName  = SND.TO.CSQ1   # Parameter e11
  MVRcvQName   = MQRCV         # Parameter f4
  MVWaitQName  = MQWAIT        # Parameter f7
  MVSendQName  = MQSND1,MQSND1,7,0,0 # Parameter f3
  MVSendQName  = MQSND2,MQSND2,3,1,0 # Parameter f3
# ConvExitName = CNV2NT        #
# MQPIIRecovery = No          #
# MVPIIRecovery = No          #
```

Sample Configuration File for a Connection to VSE

The sample configuration file NTVSEATT.ATN contains the common section (MVQAttachment), and one link-specific section (MVQLink1) for a link to a VSE system.

```
#-----
# File Name:   NTVSEATT.ATN
#
# Description: MQI Attachment configuration file
#              (1) => to be customized  in MERVA
#              (2) => to be configured  in MQSeries
#-----
# Licensed materials - Property of IBM
#
# MERVA ESA Components V4 (5648-B30)
#
# (C) Copyright IBM Corp. 1999, 2001
#
# All rights reserved.
#
# US Government Users Restricted Rights - Use, duplication or disclosure
# restricted by GSA ADP Schedule Contract with IBM Corp.
#-----

# Common Section
[MVQAttachment]
MVQName      = (CICS User ID)      # Parameter e21
MRVName      = MervamQ             #
MRVUser      = merva1              # Parameter e22
MRVPwd       = merva9              #
MQQMgrName   = NTESAATT            # Parameter e2
MQRefQName   = REFQ                # Parameter e19
# MQTActQName = TACTQ              # Parameter e20
MQTrigQName  = TRIGQ              #
ScanInterval = 60                  #
LogDir       = c:\temp             #
# MQPIIRecovery = Yes              #
# MVPIIRecovery = Yes              #
# APITrace     = No                 #
# MQClient    = No                 #
# LogLevel    = 3                   #
# First Link Section
[MVQLink1]
#
# LkName      = MVQLK1              #
RmtUser      = (CICS User ID)      # Parameter e21
MQRcvQName   = RCV.FROM.DVSCQMGR   # Parameter e12
MQSendQName  = SND.TO.DVSCQMGR    # Parameter e11
MVRcvQName   = MQRCV               # Parameter f4
MVWaitQName  = MQWAIT              # Parameter f7
MVSndQName   = MQSND1,MQSND1,7,0,0 # Parameter f3
MVSndQName   = MQSND2,MQSND2,3,1,0 # Parameter f3
# ConvExitName = CNV2NT            #
# MQPIIRecovery = No               #
# MVPIIRecovery = No               #
```

Appendix D. 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 in the product(s) and/or the program(s) described in this publication at any time without notice.

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 Deutschland
Informationssysteme GmbH
Department 3982
Pascalstrasse 100

70569 Stuttgart
Germany

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

The following paragraph does apply to the US only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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 IBM Corporation in the United States, other countries, or both:

- Advanced Peer-to-Peer Networking
- AIX
- APPN
- C/370
- CICS
- CICS/ESA
- CICS/MVS
- CICS/VSE
- DB2
- DB2 Universal Database
- Distributed Relational Database Architecture
- DRDA
- IBM
- IMS/ESA
- Language Environment
- MQSeries

- MVS
- MVS/ESA
- MVS/XA
- OS/2
- OS/390
- RACF
- VisualAge
- VSE/ESA
- VTAM

Workstation (AWS) and Directory Services Application (DSA) are trademarks of S.W.I.F.T., La Hulpe in Belgium.

Pentium is a trademark of Intel Corporation.

PC Direct is a trademark of Ziff Communications Company in the United States, other countries, or both, and is used by IBM Corporation under license.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Glossary of Terms and Abbreviations

This glossary defines terms as they are used in this book. If you do not find the terms you are looking for, refer to the *IBM Dictionary of Computing*, New York: McGraw-Hill, and the *S.W.I.F.T. User Handbook*.

A

ACB. Access method control block.

ACC. MERVA Link USS application control command application. It provides a means of operating MERVA Link USS in USS shell and MVS batch environments.

Access method control block (ACB). A control block that links an application program to VSAM or VTAM.

ACD. MERVA Link USS application control daemon.

ACT. MERVA Link USS application control table.

address. See *SWIFT address*.

address expansion. The process by which the full name of a financial institution is obtained using the SWIFT address, telex correspondent's address, or a nickname.

AMPDU. Application message protocol data unit, which is defined in the MERVA Link P1 protocol, and consists of an envelope and its content.

answerback. In telex, the response from the dialed correspondent to the WHO R U signal.

answerback code. A group of up to 6 letters following or contained in the answerback. It is used to check the answerback.

APC. Application control.

API. Application programming interface.

APPC. Advanced Program-to-Program Communication based on SNA LU 6.2 protocols.

APPL. A VTAM definition statement used to define a VTAM application program.

application programming interface (API). An interface that programs can use to exchange data.

application support filter (ASF). In MERVA Link, a user-written program that can control and modify any data exchanged between the Application Support Layer and the Message Transfer Layer.

application support process (ASP). An executing instance of an application support program. Each application support process is associated with an ASP entry in the partner table. An ASP that handles outgoing messages is a *sending ASP*; one that handles incoming messages is a *receiving ASP*.

application support program (ASP). In MERVA Link, a program that exchanges messages and reports with a specific remote partner ASP. These two programs must agree on which conversation protocol they are to use.

ASCII. American Standard Code for Information Interchange. The standard code, using a coded 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.

ASF. Application support filter.

ASF. (1) Application support process. (2) Application support program.

ASPDU. Application support protocol data unit, which is defined in the MERVA Link P2 protocol.

authentication. The SWIFT security check used to ensure that a message has not changed during transmission, and that it was sent by an authorized sender.

authenticator key. A set of alphanumeric characters used for the authentication of a message sent via the SWIFT network.

authenticator-key file. The file that stores the keys used during the authentication of a message. The file contains a record for each of your financial institution's correspondents.

B

Back-to-Back (BTB). A MERVA Link function that enables ASPs to exchange messages in the local MERVA Link node without using data communication services.

bank identifier code. A 12-character code used to identify a bank within the SWIFT network. Also called a SWIFT address. The code consists of the following subcodes:

- The bank code (4 characters)
- The ISO country code (2 characters)
- The location code (2 characters)
- The address extension (1 character)

- The branch code (3 characters) for a SWIFT user institution, or the letters "BIC" for institutions that are not SWIFT users.

Basic Security Manager (BSM). A component of VSE/ESA Version 2.4 that is invoked by the System Authorization Facility, and used to ensure signon and transaction security.

BIC. Bank identifier code.

BIC Bankfile. A tape of bank identifier codes supplied by S.W.I.F.T.

BIC Database Plus Tape. A tape of financial institutions and currency codes, supplied by S.W.I.F.T. The information is compiled from various sources and includes national, international, and cross-border identifiers.

BIC Directory Update Tape. A tape of bank identifier codes and currency codes, supplied by S.W.I.F.T., with extended information as published in the printed BIC Directory.

body. The second part of an IM-ASPDU. It contains the actual application data or the message text that the IM-AMPDU transfers.

BSC. Binary synchronous control.

BSM. Basic Security Manager.

BTB. Back-to-back.

buffer. A storage area used by MERVA programs to store a message in its internal format. A buffer has an 8-byte prefix that indicates its length.

C

CBT. SWIFT computer-based terminal.

CCSID. Coded character set identifier.

CDS. Control data set.

central service. In MERVA, a service that uses resources that either require serialization of access, or are only available in the MERVA nucleus.

CF message. Confirmed message. When a sending MERVA Link system is informed of the successful delivery of a message to the receiving application, it routes the delivered application messages as CF messages, that is, messages of class CF, to an ACK wait queue or to a complete message queue.

COA. Confirm on arrival.

COD. Confirm on delivery.

coded character set identifier (CCSID). The name of a coded set of characters and their code point assignments.

commit. In MQSeries, to commit operations is to make the changes on MQSeries queues permanent. After putting one or more messages to a queue, a commit makes them visible to other programs. After getting one or more messages from a queue, a commit permanently deletes them from the queue.

confirm-on-arrival (COA) report. An MQSeries report message type created when a message is placed on that queue. It is created by the queue manager that owns the destination queue.

confirm-on-delivery (COD) report. An MQSeries report message type created when an application retrieves a message from the queue in a way that causes the message to be deleted from the queue. It is created by the queue manager.

control fields. In MERVA Link, fields that are part of a MERVA message on the queue data set and of the message in the TOF. Control fields are written to the TOF at nesting identifier 0. Messages in SWIFT format do not contain control fields.

correspondent. An institution to which your institution sends and from which it receives messages.

correspondent identifier. The 11-character identifier of the receiver of a telex message. Used as a key to retrieve information from the Telex correspondents file.

cross-system coupling facility. See XCF.

coupling services. In a sysplex, the functions of XCF that transfer data and status information among the members of a group that reside in one or more of the MVS systems in the sysplex.

couple data set. See XCF *couple data set*.

CTP. MERVA Link command transfer processor.

currency code file. A file containing the currency codes, together with the name, fraction length, country code, and country names.

D

daemon. A long-lived process that runs unattended to perform continuous or periodic systemwide functions.

DASD. Direct access storage device.

data area. An area of a predefined length and format on a panel in which data can be entered or displayed. A field can consist of one or more data areas.

data element. A unit of data that, in a certain context, is considered indivisible. In MERVA Link, a data

element consists of a 2-byte data element length field, a 2-byte data-element identifier field, and a field of variable length containing the data element data.

datagram. In TCP/IP, the basic unit of information passed across the Internet environment. This type of message does not require a reply, and is the simplest type of message that MQSeries supports.

data terminal equipment. That part of a data station that serves as a data source, data link, or both, and provides for the data communication control function according to protocols.

DB2. A family of IBM licensed programs for relational database management.

dead-letter queue. A queue to which a queue manager or application sends messages that it cannot deliver. Also called *undelivered-message queue*.

dial-up number. A series of digits required to establish a connection with a remote correspondent via the public telex network.

direct service. In MERVA, a service that uses resources that are always available and that can be used by several requesters at the same time.

display mode. The mode (PROMPT or NOPROMPT) in which SWIFT messages are displayed. See *PROMPT mode* and *NOPROMPT mode*.

distributed queue management (DQM). In MQSeries message queuing, the setup and control of message channels to queue managers on other systems.

DQM. Distributed queue management.

DTE. Data terminal equipment.

E

EBCDIC. Extended Binary Coded Decimal Interchange Code. A coded character set consisting of 8-bit coded characters.

ECB. Event control block.

EDIFACT. Electronic Data Interchange for Administration, Commerce and Transport (a United Nations standard).

ESM. External security manager.

EUD. End-user driver.

exception report. An MQSeries report message type that is created by a message channel agent when a message is sent to another queue manager, but that message cannot be delivered to the specified destination queue.

external line format (ELF) messages. Messages that are not fully tokenized, but are stored in a single field in the TOF. Storing messages in ELF improves performance, because no mapping is needed, and checking is not performed.

external security manager (ESM). A security product that is invoked by the System Authorization Facility. RACF is an example of an ESM.

F

FDT. Field definition table.

field. In MERVA, a portion of a message used to enter or display a particular type of data in a predefined format. A field is located by its position in a message and by its tag. A field is made up of one or more data areas. See also *data area*.

field definition table (FDT). The field definition table describes the characteristics of a field; for example, its length and number of its data areas, and whether it is mandatory. If the characteristics of a field change depending on its use in a particular message, the definition of the field in the FDT can be overridden by the MCB specifications.

field group. One or several fields that are defined as being a group. Because a field can occur more than once in a message, field groups are used to distinguish them. A name can be assigned to the field group during message definition.

field group number. In the TOF, a number is assigned to each field group in a message in ascending order from 1 to 255. A particular field group can be accessed using its field group number.

field tag. A character string used by MERVA to identify a field in a network buffer. For example, for SWIFT field 30, the field tag is :30.

FIN. Financial application.

FIN-Copy. The MERVA component used for SWIFT FIN-Copy support.

finite state machine. The theoretical base describing the rules of a service request's state and the conditions to state transitions.

FMT/ESA. MERVA-to-MERVA Financial Message Transfer/ESA.

form. A partially-filled message containing data that can be copied for a new message of the same message type.

G

GPA. General purpose application.

H

HFS. Hierarchical file system.

hierarchical file system (HFS). A system for organizing files in a hierarchy, as in a UNIX system. OS/390 UNIX System Services files are organized in an HFS. All files are members of a directory, and each directory is in turn a member of a directory at a higher level in the HFS. The highest level in the hierarchy is the root directory.

I

IAM. Interapplication messaging (a MERVA Link message exchange protocol).

IM-ASPDU. Interapplication messaging application support protocol data unit. It contains an application message and consists of a heading and a body.

incore request queue. Another name for the request queue to emphasize that the request queue is held in memory instead of on a DASD.

InetD. Internet Daemon. It provides TCP/IP communication services in the OS/390 USS environment.

initiation queue. In MQSeries, a local queue on which the queue manager puts trigger messages.

input message. A message that is input into the SWIFT network. An input message has an input header.

INTERCOPE TelexBox. This telex box supports various national conventions for telex procedures and protocols.

interservice communication. In MERVA ESA, a facility that enables communication among services if MERVA ESA is running in a multisystem environment.

intertask communication. A facility that enables application programs to communicate with the MERVA nucleus and so request a central service.

IP. Internet Protocol.

IP message. In-process message. A message that is in the process of being transferred to another application.

ISC. Intersystem communication.

ISN. Input sequence number.

ISN acknowledgment. A collective term for the various kinds of acknowledgments sent by the SWIFT network.

ISO. International Organization for Standardization.

ITC. Intertask communication.

J

JCL. Job control language.

journal. A chronological list of records detailing MERVA actions.

journal key. A key used to identify a record in the journal.

journal service. A MERVA central service that maintains the journal.

K

KB. Kilobyte (1024 bytes).

key. A character or set of characters used to identify an item or group of items. For example, the user ID is the key to identify a user file record.

key-sequenced data set (KSDS). A VSAM data set whose records are loaded in key sequence and controlled by an index.

keyword parameter. A parameter that consists of a keyword, followed by one or more values.

KSDS. Key-sequenced data set.

L

LAK. Login acknowledgment message. This message informs you that you have successfully logged in to the SWIFT network.

large message. A message that is stored in the large message cluster (LMC). The maximum length of a message to be stored in the VSAM QDS is 31900 bytes. Messages up to 2MB can be stored in the LMC. For queue management using DB2 no distinction is made between messages and large messages.

large queue element. A queue element that is larger than the smaller of:

- The limiting value specified during the customization of MERVA
- 32KB

LC message. Last confirmed control message. It contains the message-sequence number of the application or acknowledgment message that was last confirmed; that is, for which the sending MERVA Link system most recently received confirmation of a successful delivery.

LDS. Logical data stream.

LMC. Large message cluster.

LNK. Login negative acknowledgment message. This message indicates that the login to the SWIFT network has failed.

local queue. In MQSeries, a queue that belongs to a local queue manager. A local queue can contain a list of messages waiting to be processed. Contrast with *remote queue*.

local queue manager. In MQSeries, the queue manager to which the program is connected, and that provides message queuing services to that program. Queue managers to which a program is not connected are remote queue managers, even if they are running on the same system as the program.

login. To start the connection to the SWIFT network.

LR message. Last received control message, which contains the message-sequence number of the application or acknowledgment message that was last received from the partner application.

LSN. Login sequence number.

LT. See *LTERM*.

LTC. Logical terminal control.

LTERM. Logical terminal. Logical terminal names have 4 characters in CICS and up to 8 characters in IMS.

LU. A VTAM logical unit.

M

maintain system history program (MSHP). A program used for automating and controlling various installation, tailoring, and service activities for a VSE system.

MCA. Message channel agent.

MCB. Message control block.

MERVA ESA. The IBM licensed program Message Entry and Routing with Interfaces to Various Applications for ESA.

MERVA Link. A MERVA component that can be used to interconnect several MERVA systems.

message. A string of fields in a predefined form used to provide or request information. See also *SWIFT financial message*.

message body. The part of the message that contains the message text.

message category. A group of messages that are logically related within an application.

message channel. In MQSeries distributed message queuing, a mechanism for moving messages from one queue manager to another. A message channel comprises two message channel agents (a sender and a receiver) and a communication link.

message channel agent (MCA). In MQSeries, a program that transmits prepared messages from a transmission queue to a communication link, or from a communication link to a destination queue.

message control block (MCB). The definition of a message, screen panel, net format, or printer layout made during customization of MERVA.

Message Format Service (MFS). A MERVA direct service that formats a message according to the medium to be used, and checks it for formal correctness.

message header. The leading part of a message that contains the sender and receiver of the message, the message priority, and the type of message.

Message Integrity Protocol (MIP). In MERVA Link, the protocol that controls the exchange of messages between partner ASPs. This protocol ensures that any loss of a message is detected and reported, and that no message is duplicated despite system failures at any point during the transfer process.

message-processing function. The various parts of MERVA used to handle a step in the message-processing route, together with any necessary equipment.

message queue. See *queue*.

Message Queue Interface (MQI). The programming interface provided by the MQSeries queue managers. It provides a set of calls that let application programs access message queuing services such as sending messages, receiving messages, and manipulating MQSeries objects.

Message Queue Manager (MQM). An IBM licensed program that provides message queuing services. It is part of the MQSeries set of products.

message reference number (MRN). A unique 16-digit number assigned to each message for identification purposes. The message reference number consists of an 8-digit domain identifier that is followed by an 8-digit sequence number.

message sequence number (MSN). A sequence number for messages transferred by MERVA Link.

message type (MT). A number, up to 7 digits long, that identifies a message. SWIFT messages are identified by a 3-digit number; for example SWIFT message type MT S100.

MFS. Message Format Service.

MIP. Message Integrity Protocol.

MPDU. Message protocol data unit, which is defined in P1.

MPP. In IMS, message-processing program.

MQA. MQ Attachment.

MQ Attachment (MQA). A MERVA feature that provides message transfer between MERVA and a user-written MQI application.

MQH. MQSeries queue handler.

MQI. Message queue interface.

MQM. Message queue manager.

MQS. MQSeries nucleus server.

MQSeries. A family of IBM licensed programs that provides message queuing services.

MQSeries nucleus server (MQS). A MERVA component that listens for messages on an MQI queue, receives them, extracts a service request, and passes it via the request queue handler to another MERVA ESA instance for processing.

MQSeries queue handler (MQH). A MERVA component that performs service calls to the Message Queue Manager via the provided Message Queue Interface.

MRN. Message reference number.

MSC. MERVA system control facility.

MSHP. Maintain system history program.

MSN. Message sequence number.

MT. Message type.

MTP. (1) Message transfer program. (2) Message transfer process.

MTS. Message Transfer System.

MTSP. Message Transfer Service Processor.

MTT. Message type table.

multisystem application. (1) An application program that has various functions distributed across MVS systems in a multisystem environment. (2) In XCF, an authorized application that uses XCF coupling services. (3) In MERVA ESA, multiple instances of MERVA ESA that are distributed among different MVS systems in a multisystem environment.

multisystem environment. An environment in which two or more MVS systems reside on one or more processors, and programs on one system can communicate with programs on the other systems. With XCF, the environment in which XCF services are available in a defined sysplex.

multisystem sysplex. A sysplex in which one or more MVS systems can be initialized as part of the sysplex. In a multisystem sysplex, XCF provides coupling services on all systems in the sysplex and requires an XCF couple data set that is shared by all systems. See also *single-system sysplex*.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture.

N

namelist. An MQSeries for MVS/ESA object that contains a list of queue names.

nested message. A message that is composed of one or more message types.

nested message type. A message type that is contained in another message type. In some cases, only part of a message type (for example, only the mandatory fields) is nested, but this "partial" nested message type is also considered to be nested. For example, SWIFT MT 195 could be used to request information about a SWIFT MT 100 (customer transfer). The SWIFT MT 100 (or at least its mandatory fields) is then nested in SWIFT MT 195.

nesting identifier. An identifier (a number from 2 to 255) that is used to access a nested message type.

network identifier. A single character that is placed before a message type to indicate which network is to be used to send the message; for example, **S** for SWIFT

network service access point (NSAP). The endpoint of a network connection used by the SWIFT transport layer.

NOPROMPT mode. One of two ways to display a message panel. NOPROMPT mode is only intended for experienced SWIFT Link users who are familiar with the structure of SWIFT messages. With NOPROMPT mode, only the SWIFT header, trailer, and pre-filled fields and their tags are displayed. Contrast with *PROMPT mode*.

NSAP. Network service access point.

nucleus server. A MERVA component that processes a service request as selected by the request queue handler. The service a nucleus server provides and the way it provides it is defined in the nucleus server table (DSLNSVT).

O

object. In MQSeries, objects define the properties of queue managers, queues, process definitions, and namelists.

occurrence. See *repeatable sequence*.

option. One or more characters added to a SWIFT field number to distinguish among different layouts for and meanings of the same field. For example, SWIFT field 60 can have an option F to identify a first opening balance, or M for an intermediate opening balance.

origin identifier (origin ID). A 34-byte field of the MERVA user file record. It indicates, in a MERVA and SWIFT Link installation that is shared by several banks, to which of these banks the user belongs. This lets the user work for that bank only.

OSN. Output sequence number.

OSN acknowledgment. A collective term for the various kinds of acknowledgments sent to the SWIFT network.

output message. A message that has been received from the SWIFT network. An output message has an output header.

P

P1. In MERVA Link, a peer-to-peer protocol used by cooperating message transfer processes (MTPs).

P2. In MERVA Link, a peer-to-peer protocol used by cooperating application support processes (ASPs).

P3. In MERVA Link, a peer-to-peer protocol used by cooperating command transfer processors (CTPs).

packet switched public data network (PSPDN). A public data network established and operated by network common carriers or telecommunication administrations for providing packet-switched data transmission.

panel. A formatted display on a display terminal. Each page of a message is displayed on a separate panel.

parallel processing. The simultaneous processing of units of work by several servers. The units of work can be either transactions or subdivisions of larger units of work.

parallel sysplex. A sysplex that uses one or more coupling facilities.

partner table (PT). In MERVA Link, the table that defines how messages are processed. It consists of a

header and different entries, such as entries to specify the message-processing parameters of an ASP or MTP.

PCT. Program Control Table (of CICS).

PDE. Possible duplicate emission.

PDU. Protocol data unit.

PF key. Program-function key.

positional parameter. A parameter that must appear in a specified location relative to other parameters.

PREMIUM. The MERVA component used for SWIFT PREMIUM support.

process definition object. An MQSeries object that contains the definition of an MQSeries application. A queue manager uses the definitions contained in a process definition object when it works with trigger messages.

program-function key. A key on a display terminal keyboard to which a function (for example, a command) can be assigned. This lets you execute the function (enter the command) with a single keystroke.

PROMPT mode. One of two ways to display a message panel. PROMPT mode is intended for SWIFT Link users who are unfamiliar with the structure of SWIFT messages. With PROMPT mode, all the fields and tags are displayed for the SWIFT message. Contrast with *NOPROMPT mode*.

protocol data unit (PDU). In MERVA Link a PDU consists of a structured sequence of implicit and explicit data elements:

- Implicit data elements contain other data elements.
- Explicit data elements cannot contain any other data elements.

PSN. Public switched network.

PSPDN. Packet switched public data network.

PSTN. Public switched telephone network.

PT. Partner table.

PTT. A national post and telecommunication authority (post, telegraph, telephone).

Q

QDS. Queue data set.

QSN. Queue sequence number.

queue. (1) In MERVA, a logical subdivision of the MERVA queue data set used to store the messages associated with a MERVA message-processing function. A queue has the same name as the message-processing function with which it is associated. (2) In MQSeries, an

object onto which message queuing applications can put messages, and from which they can get messages. A queue is owned and maintained by a queue manager. See also *request queue*.

queue element. A message and its related control information stored in a data record in the MERVA ESA Queue Data Set.

queue management. A MERVA service function that handles the storing of messages in, and the retrieval of messages from, the queues of message-processing functions.

queue manager. (1) An MQSeries system program that provides queueing services to applications. It provides an application programming interface so that programs can access messages on the queues that the queue manager owns. See also *local queue manager* and *remote queue manager*. (2) The MQSeries object that defines the attributes of a particular queue manager.

queue sequence number (QSN). A sequence number that is assigned to the messages stored in a logical queue by MERVA ESA queue management in ascending order. The QSN is always unique in a queue. It is reset to zero when the queue data set is formatted, or when a queue management restart is carried out and the queue is empty.

R

RACF. Resource Access Control Facility.

RBA. Relative byte address.

RC message. Recovered message; that is, an IP message that was copied from the control queue of an inoperable or closed ASP via the **recover** command.

ready queue. A MERVA queue used by SWIFT Link to collect SWIFT messages that are ready for sending to the SWIFT network.

remote queue. In MQSeries, a queue that belongs to a remote queue manager. Programs can put messages on remote queues, but they cannot get messages from remote queues. Contrast with *local queue*.

remote queue manager. In MQSeries, a queue manager is remote to a program if it is not the queue manager to which the program is connected.

repeatable sequence. A field or a group of fields that is contained more than once in a message. For example, if the SWIFT fields 20, 32, and 72 form a sequence, and if this sequence can be repeated up to 10 times in a message, each sequence of the fields 20, 32, and 72 would be an occurrence of the repeatable sequence.

In the TOF, the occurrences of a repeatable sequence are numbered in ascending order from 1 to 32767 and can be referred to using the occurrence number.

A repeatable sequence in a message may itself contain another repeatable sequence. To identify an occurrence within such a nested repeatable sequence, more than one occurrence number is necessary.

reply message. In MQSeries, a type of message used for replies to request messages.

reply-to queue. In MQSeries, the name of a queue to which the program that issued an MQPUT call wants a reply message or report message sent.

report message. In MQSeries, a type of message that gives information about another message. A report message usually indicates that the original message cannot be processed for some reason.

request message. In MQSeries, a type of message used for requesting a reply from another program.

request queue. The queue in which a service request is stored. It resides in main storage and consists of a set of request queue elements that are chained in different queues:

- Requests waiting to be processed
- Requests currently being processed
- Requests for which processing has finished

request queue handler (RQH). A MERVA ESA component that handles the queueing and scheduling of service requests. It controls the request processing of a nucleus server according to rules defined in the finite state machine.

Resource Access Control Facility (RACF). An IBM licensed program that provides for access control by identifying and verifying users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

retype verification. See *verification*.

routing. In MERVA, the passing of messages from one stage in a predefined processing path to the next stage.

RP. Regional processor.

RQH. Request queue handler.

RRDS. Relative record data set.

S

SAF. System Authorization Facility.

SCS. SNA character string

SCP. System control process.

SDI. Sequential data set input. A batch utility used to import messages from a sequential data set or a tape into MERVA ESA queues.

SDO. Sequential data set output. A batch utility used to export messages from a MERVA ESA queue to a sequential data set or a tape.

SDY. Sequential data set system printer. A batch utility used to print messages from a MERVA ESA queue.

service request. A type of request that is created and passed to the request queue handler whenever a nucleus server requires a service that is not currently available.

sequence number. A number assigned to each message exchanged between two nodes. The number is increased by one for each successive message. It starts from zero each time a new session is established.

sign off. To end a session with MERVA.

sign on. To start a session with MERVA.

single-system sysplex. A sysplex in which only one MVS system can be initialized as part of the sysplex. In a single-system sysplex, XCF provides XCF services on the system, but does not provide signalling services between MVS systems. A single-system sysplex requires an XCF couple data set. See also *multisystem sysplex*.

small queue element. A queue element that is smaller than the smaller of:

- The limiting value specified during the customization of MERVA
- 32KB

SMP/E. System Modification Program Extended.

SN. Session number.

SNA. Systems network architecture.

SNA character string. In SNA, a character string composed of EBCDIC controls, optionally mixed with user data, that is carried within a request or response unit.

SPA. Scratch pad area.

SQL. Structured Query Language.

SR-ASPDU. The status report application support PDU, which is used by MERVA Link for acknowledgment messages.

SSN. Select sequence number.

subfield. A subdivision of a field with a specific meaning. For example, the SWIFT field 32 has the subfields date, currency code, and amount. A field can

have several subfield layouts depending on the way the field is used in a particular message.

SVC. (1) Switched Virtual Circuit. (2) Supervisor call instruction.

S.W.I.F.T. (1) Society for Worldwide Interbank Financial Telecommunication s.c. (2) The network provided and managed by the Society for Worldwide Interbank Financial Telecommunication s.c.

SWIFT address. Synonym for *bank identifier code*.

SWIFT Correspondents File. The file containing the bank identifier code (BIC), together with the name, postal address, and zip code of each financial institution in the BIC Directory.

SWIFT financial message. A message in one of the SWIFT categories 1 to 9 that you can send or receive via the SWIFT network. See *SWIFT input message* and *SWIFT output message*.

SWIFT header. The leading part of a message that contains the sender and receiver of the message, the message priority, and the type of message.

SWIFT input message. A SWIFT message with an input header to be sent to the SWIFT network.

SWIFT link. The MERVA ESA component used to link to the SWIFT network.

SWIFT network. Refers to the SWIFT network of the Society for Worldwide Interbank Financial Telecommunication (S.W.I.F.T.).

SWIFT output message. A SWIFT message with an output header coming from the SWIFT network.

SWIFT system message. A SWIFT general purpose application (GPA) message or a financial application (FIN) message in SWIFT category 0.

switched virtual circuit (SVC). An X.25 circuit that is dynamically established when needed. It is the X.25 equivalent of a switched line.

sysplex. One or more MVS systems that communicate and cooperate via special multisystem hardware components and software services.

System Authorization Facility (SAF). An MVS or VSE facility through which MERVA ESA communicates with an external security manager such as RACF (for MVS) or the basic security manager (for VSE).

System Control Process (SCP). A MERVA Link component that handles the transfer of MERVA ESA commands to a partner MERVA ESA system, and the receipt of the command response. It is associated with a system control process entry in the partner table.

System Modification Program Extended (SMP/E). A licensed program used to install software and software changes on MVS systems.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operating sequences for transmitting information units through, and for controlling the configuration and operation of, networks.

T

tag. A field identifier.

TCP/IP. Transmission Control Protocol/Internet Protocol.

Telex Correspondents File. A file that stores data about correspondents. When the user enters the corresponding nickname in a Telex message, the corresponding information in this file is automatically retrieved and entered into the Telex header area.

telex header area. The first part of the telex message. It contains control information for the telex network.

telex interface program (TXIP). A program that runs on a Telex front-end computer and provides a communication facility to connect MERVA ESA with the Telex network.

Telex Link. The MERVA ESA component used to link to the public telex network via a Telex substation.

Telex substation. A unit comprised of the following:

- Telex Interface Program
- A Telex front-end computer
- A Telex box

Terminal User Control Block (TUCB). A control block containing terminal-specific and user-specific information used for processing messages for display devices such as screen and printers.

test key. A key added to a telex message to ensure message integrity and authorized delivery. The test key is an integer value of up to 16 digits, calculated manually or by a test-key processing program using the significant information in the message, such as amounts, currency codes, and the message date.

test-key processing program. A program that automatically calculates and verifies a test key. The Telex Link supports panels for input of test-key-related data and an interface for a test-key processing program.

TFD. Terminal feature definitions table.

TID. Terminal identification. The first 9 characters of a bank identifier code (BIC).

TOF. Originally the abbreviation of *tokenized form*, the TOF is a storage area where messages are stored so that their fields can be accessed directly by their field names and other index information.

TP. Transaction program.

transaction. A specific set of input data that triggers the running of a specific process or job; for example, a message destined for an application program.

transaction code. In IMS and CICS, an alphanumeric code that calls an IMS message processing program or a CICS transaction. Transaction codes have 4 characters in CICS and up to 8 characters in IMS.

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

transmission queue. In MQSeries, a local queue on which prepared messages destined for a remote queue manager are temporarily stored.

trigger event. In MQSeries, an event (such as a message arriving on a queue) that causes a queue manager to create a trigger message on an initiation queue.

trigger message. In MQSeries, a message that contains information about the program that a trigger monitor is to start.

trigger monitor. In MQSeries, a continuously-running application that serves one or more initiation queues. When a trigger message arrives on an initiation queue, the trigger monitor retrieves the message. It uses the information in the trigger message to start a process that serves the queue on which a trigger event occurred.

triggering. In MQSeries, a facility that allows a queue manager to start an application automatically when predetermined conditions are satisfied.

TUCB. Terminal User Control Block.

TXIP. Telex interface program.

U

UMR. Unique message reference.

unique message reference (UMR). An optional feature of MERVA ESA that provides each message with a unique identifier the first time it is placed in a queue. It is composed of a MERVA ESA installation name, a sequence number, and a date and time stamp.

UNIT. A group of related literals or fields of an MCB definition, or both, enclosed by a DSLUNIT and DSLUEND macroinstruction.

UNIX System Services (USS). A component of OS/390, formerly called OpenEdition (OE), that creates a UNIX environment that conforms to the XPG4 UNIX 1995 specifications, and provides two open systems interfaces on the OS/390 operating system:

- An application program interface (API)
- An interactive shell interface

UN/EDIFACT. United Nations Standard for Electronic Data Interchange for Administration, Commerce and Transport.

USE. S.W.I.F.T. User Security Enhancements.

user file. A file containing information about all MERVAs ESA users; for example, which functions each user is allowed to access. The user file is encrypted and can only be accessed by authorized persons.

user identification and verification. The acts of identifying and verifying a RACF-defined user to the system during logon or batch job processing. RACF identifies the user by the user ID and verifies the user by the password or operator identification card supplied during logon processing or the password supplied on a batch JOB statement.

USS. UNIX System Services.

V

verification. Checking to ensure that the contents of a message are correct. Two kinds of verification are:

- Visual verification: you read the message and confirm that you have done so
- Retype verification: you reenter the data to be verified

Virtual LU. An LU defined in MERVAs Extended Connectivity for communication between MERVAs and MERVAs Extended Connectivity.

Virtual Storage Access Method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry sequence), or by relative-record number.

Virtual Telecommunications Access Method (VTAM). An IBM licensed program that controls communication and the flow of data in an SNA network. It provides single-domain, multiple-domain, and interconnected network capability.

VSAM. Virtual Storage Access Method.

VTAM. Virtual Telecommunications Access Method (IBM licensed program).

W

Windows NT service. A type of Windows NT application that can run in the background of the Windows NT operating system even when no user is logged on. Typically, such a service has no user interaction and writes its output messages to the Windows NT event log.

X

X.25. An ISO standard for interface to packet switched communications services.

XCF. Abbreviation for *cross-system coupling facility*, which is a special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex. XCF provides the MVS coupling services that allow authorized programs on MVS systems in a multisystem environment to communicate with (send data to and receive data from) authorized programs on other MVS systems.

XCF couple data sets. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the MVS systems in a sysplex. It is accessed only by XCF and contains XCF-related data about the sysplex, systems, applications, groups, and members.

XCF group. The set of related members defined to SCF by a multisystem application in which members of the group can communicate with (send data to and receive data from) other members of the same group. All MERVAs systems working together in a sysplex must pertain to the same XCF group.

XCF member. A specific function of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in a sysplex and can use XCF services to communicate with other members of the same group.

Bibliography

MERVA ESA Publications

- *MERVA for ESA Version 4: Application Programming Interface Guide*, SH12-6374
- *MERVA for ESA Version 4: Advanced MERVA Link*, SH12-6390
- *MERVA for ESA Version 4: Concepts and Components*, SH12-6381
- *MERVA for ESA Version 4: Customization Guide*, SH12-6380
- *MERVA for ESA Version 4: Diagnosis Guide*, SH12-6382
- *MERVA for ESA Version 4: Installation Guide*, SH12-6378
- *MERVA for ESA Version 4: Licensed Program Specifications*, GH12-6373
- *MERVA for ESA Version 4: Macro Reference*, SH12-6377
- *MERVA for ESA Version 4: Messages and Codes*, SH12-6379
- *MERVA for ESA Version 4: Operations Guide*, SH12-6375
- *MERVA for ESA Version 4: System Programming Guide*, SH12-6366
- *MERVA for ESA Version 4: User's Guide*, SH12-6376

MERVA ESA Components Publications

- *MERVA Automatic Message Import/Export Facility: User's Guide*, SH12-6389
- *MERVA Connection/NT*, SH12-6339
- *MERVA Connection/400*, SH12-6340
- *MERVA Directory Services*, SH12-6367
- *MERVA Extended Connectivity: Installation and User's Guide*, SH12-6157
- *MERVA Message Processing Client for Windows NT: User's Guide*, SH12-6341
- *MERVA-MQI Attachment User's Guide*, SH12-6714
- *MERVA Traffic Reconciliation*, SH12-6392
- *MERVA USE: Administration Guide*, SH12-6338
- *MERVA USE & Branch for Windows NT: User's Guide*, SH12-6334

- *MERVA USE & Branch for Windows NT: Installation and Customization Guide*, SH12-6335
- *MERVA USE & Branch for Windows NT: Application Programming Guide*, SH12-6336
- *MERVA USE & Branch for Windows NT: Diagnosis Guide*, SH12-6337
- *MERVA USE & Branch for Windows NT: Migration Guide*, SH12-6393
- *MERVA USE & Branch for Windows NT: Installation and Customization Guide*, SH12-6335
- *MERVA Workstation Based Functions*, SH12-6383

Other IBM Publications

- *APPC Security: MVS/ESA, CICS/ESA, and OS/2*, GG24-3960
- *CICS/ESA V4.1 Resource Definition Guide*, SC33-1166
- *CICS Transaction Server for OS/390 Resource Definition Guide*, SC33-1684
- *CICS Transaction Server for VSE/ESA Resource Definition Guide*, SC33-1653
- *CICS/VSE 2.3 Resource Definition (Macro)*, SC33-0709
- *CICS/VSE 2.3 Resource Definition (Online)*, SC33-0708
- *IMS/ESA Version 6 Administration Guide: Transaction Manager*, SC26-8731
- *IMS/ESA Version 6 Installation Volume 2: System Definition and Tailoring*, GC26-8737
- *MQSeries Application Programming Reference*, SC33-1673
- *MQSeries for OS/390 System Administration Guide*, SC34-5652
- *MQSeries for OS/390 Messages and Codes*, GC34-5891
- *MQSeries for VSE/ESA System Management Guide*, GC34-5364
- *MQSeries MQSC Command Reference*, SC33-1369
- *Multi-Protocol Adapter Installation Guide*, 95X2560-9

S.W.I.F.T. Publications

The following are published by the Society for Worldwide Interbank Financial Telecommunication, s.c., in La Hulpe, Belgium:

- *S.W.I.F.T. User Handbook*
- *S.W.I.F.T. Dictionary*
- *S.W.I.F.T. FIN Security Guide*
- *S.W.I.F.T. Card Readers User Guide*

Index

A

- APPC
 - configuring connection 28
 - configuring with Communications Server for Windows NT 28
 - establishing connection to host 28
 - LU name (on Windows NT) 38
 - LU name (on workstation) 11, 87
- APPC/IMS
 - base LU 18
 - customizing 18
 - definitions in SYS1.PARMLIB 154
 - examples 154
 - sample LU definition 146
 - TP profile data 154
- APPC/MVS
 - base LU 14, 15, 18, 19, 72, 89
 - customizing 18
 - definitions in SYS1.PARMLIB 153
 - examples 152
 - sample LU definition 145
 - TP profile data 153
- APPCLU62 146
- application support process name 12, 19, 38, 71, 72, 88
- APPLID 15, 16
- ASP example
 - telex 155
 - USE 155
- ASP name of MERVA AIX 12, 19, 88
- ASP name of MERVA ESA 12, 19, 38, 71, 88
- ASP name of MERVA OS/2 12, 19, 72, 88
- ASP name of MERVA USE & Branch 38
- ATBSDFMU program 18
- authenticated, whether outgoing messages are 19, 72
- authenticated and encrypted, whether outgoing messages are 12, 88

B

- background process 70
- Base Functions of MERVA ESA 3, 4
- bilateral key exchange 6
 - database update 6
 - description of 6
 - function table
 - MERVA Link 158
 - routing of messages relating to 13
- BKE 6

C

- CICS
 - customizing 15, 90
 - defining APPC connection 151
 - MERVA Link 150

- CICS (*continued*)
 - defining programs at macro level
 - MERVA Link 149
 - MERVA-MQI Attachment 165
 - defining transactions at macro level
 - MERVA Link 149
 - MERVA-MQI Attachment 165
 - processing program table (PPT)
 - MERVA Link 149
 - MERVA-MQI Attachment 165
 - program control table (PCT) 165
 - MERVA Link 149
 - sample LU definition 145
 - terminal control table (TCT) 17
 - MERVA Link 150
 - terminal definition 17
 - transaction ID 13, 16, 89
 - what to customize on 9, 85
- CICS/ESA startup job
 - MERVA Link 164
 - MERVA-MQI Attachment 190
- CICS examples
 - CSD 150
- CICS resources
 - MERVA Link 148
 - MERVA-MQI Attachment 165
- communications controller, Token Ring
 - address of 11, 86
- Communications Server for OS/2
 - parameters
 - parameter LU name 15
 - partner LU name 14, 89
- Communications Server for Windows NT
 - configuring 28
- configuration
 - example 145, 165
 - operating environment of sample 9, 85
- control queue name 12, 19, 72, 88
- conversation security 14, 39
- copy book
 - DSLCSDDxx 91
 - DSLPPPT21 91
 - DWSCSDxx 17, 91
 - DWSFNNTTC 64, 73, 134
 - DWSPPT21 17, 91
 - EKACSDxx 16, 17
 - EKAPPT21 17
 - ENLCSDDxx 17
 - ENLFNNT2 73, 76
 - ENLPPT21 17
- CSD, examples 150
- CSD utility examples
 - MERVA Link 150
- customization
 - of APPC/IMS 18
 - of APPC/MVS 18
 - of CICS 9, 15, 90
 - of CICS or IMS 85
 - of host 14, 89
 - of IMS 10, 17, 91

- customization (*continued*)
 - of MERVA ESA for telex 71
 - of MERVA ESA for USE 19, 103
 - of VTAM 15
 - of workstation based components 28
 - overview of 9, 85
 - Windows NT based resources 106
- Customization program
 - local ASP name 12
 - local MTP name 12
 - partner ASP name 12
 - partner LU name 11
 - partner MTN name 12
 - partner MTP name 12
 - transaction ID 14
 - using to change standard routing 38
- customizing MERVA 106
- customizing MQSeries 92

D

- decryption 111
- defining MERVA-MQI Attachment
 - resources under NT
 - for MVS 202
 - for VSE 203
- defining MQSeries resources under NT
 - for MVS 193
 - for VSE 197
- destination address (of LAN) 11, 15, 86
- destination name, symbolic 11, 87
- DFHCSDUP 17, 91
- DFHCSDUP, offline utility program 16
- DSLCSDDxx copy book 91
- DSLFNNTTC 176
- DSLKCSQC (example) 168
- DSLKCSQI (example) 169
- DSLKCSQU (example) 170
- DSLKPUSE (sample entries) 173
- DSLKPUSV (sample entries) 174
- DSLKQR 91
- DSLKQS 91
- DSLPPPT21 copy book 91
- DWSCSDxx copy book 17, 91
- DWSFNNTTC 76, 176
- DWSFNNTTC copy book 64, 73, 134
- DWSL1IN routing table 68, 138
- DWSL1OUT routing table 68, 138
- DWSL2IN routing table 68, 138
- DWSL2OUT routing table 68, 138
- DWSL3FII routing table 68, 138
- DWSL3FIO routing table 68, 138
- DWSL3GPI routing table 68, 138
- DWSL3GPO routing table 68, 138
- DWSLTT example for USE 178
 - MERVA Link 160
- DWSPPT21 copy book 17, 91
- DWSRTSK 179
- DWSRTSK routing table 67, 137

E

EKAAS10 17
EKACSDxx copy book 17
EKAMU133 user exit
 exit number 7133 72
 exit number or name 12, 88
 formats supported by 5
 overview of 5
EKAOSVR 11, 87
EKAPPT21 copy book 17
EKAPTTXU 155
EKARTTXU 4
EKARTTXU example 179
EKARTTXU routing table 65, 67, 68, 78,
 135, 137, 138
EKATPO1 14, 19, 72, 89
encryption 111
ENLCSDDxx copy book 17
ENLFNTT2
 DWSFNNTC 76
ENLFNTT2 copy book 73
ENLPPT21 copy book 17
ENLR2AI0 routing table 76
ENLR2DE0 routing table 76
ENLR2TKC routing table 76
ENLR2TKV routing table 78
ENLR2TLC example 160
ENLR2VE0 routing table 76
exchange identification 11, 86
external message transfer process (MTP)
 name 12, 19, 72, 88

F

format specification 13, 19, 72, 89
function table
 DWSFNNTC 21, 105
 example for USE and telex
 MERVA Link 158
function table for USE 176
functions, incoming telexes 78

H

host
 customizing 14, 89
 establishing connection to
 workstation 28
Host Receiving TP Name 38

I

IMS
 customizing 17, 91
 what to customize on 10
IMS examples
 MERVA Link 152
 MERVA-MQI Attachment 167
IMS MPP job
 MERVA-MQI Attachment 191
incoming report routing parameters 13,
 72
integrity, message 5
internal message transfer process
 (MTP) 12, 19, 72, 88

ISC=YES parameter definition 13, 16, 89

J

journaling parameters 13, 72

L

LAN destination address 11, 15, 86
local ASP name 38
local MTP name 12, 19, 38, 72, 88
local node ID 11, 15, 86
local node name 11, 15, 86
logical terminal table (DWSLTT) 21, 105
 assembling and link-editing 21, 106
 reference to routing tables in 21, 105
logical terminal table example for
 USE 178
 MERVA Link 160
Logmode 146
logon mode entry 11, 15, 87
LU 6.2, establishing connection 28
LU name 11, 87
 MERVA USE & Branch 38
 of CICS system 11, 15, 38, 86
 of PS/2 71
 of RS/6000 15, 19
 of workstation 14, 15, 19, 89

M

macro level
 CICS transaction definitions
 MERVA Link 149
 MERVA-MQI Attachment 165
 defining APPC connection at
 MERVA Link 150
 program definition
 MERVA Link 149
 MERVA-MQI Attachment 165
macro level interface examples
 MERVA Link 149
 MERVA-MQI Attachment 165
master logical terminal name 19
MERVA Base component of MERVA USE
& Branch
 overview 6
 services provided by 6
MERVA Link 157
 control queue 12, 19, 72, 88
 conversation security 39
 EKAPT header 155
 examples 155
 partner table header 155
 send queue 12, 19, 88
 send queue (PC) 12, 88
 send queue cluster 13, 72
MERVA Link Definitions (of
Customization program)
 local ASP name 12
 local MTP name 12
 partner ASP name 12
 partner LU name 11
 partner MTN name 12
 partner MTP name 12
 transaction ID 14

MERVA Link of MERVA AIX
 send queue name 12, 88
MERVA Link of MERVA ESA
 control queue name 12, 19, 72, 88
 security parameter 12, 19, 72, 88
 send queue name 12, 19, 88
 specifying journaling options 13, 19,
 72
MERVA Link of MERVA OS/2 12, 88
MERVA Link of MERVA USE & Branch
 customizing 38
MERVA-MQI Attachment 175
 examples 173
MERVA-MQI Attachment resources,
 defining under NT
 for MVS 202
 for VSE 203
MERVA AIX
 MLSNDNRM queue 68, 138
 UFROMSWF queue 70
MERVA ESA
 additional information 73
 BKEUPDT queue 68, 138
 components located on 4
 components of 3
 L1ACK queue 68, 138
 L1DO0 queue 68, 138
 L1SDO queue 68, 138
 overview 4
 SLSLOAD queue 65, 135
 SLSRECV queue 67, 137
 TX2USESQ queue 67, 68, 137, 138
MERVA ESA components
 Base Functions of MERVA ESA 4
 MERVA Link 4
 MERVA-MQI Attachment 4, 5, 6
 SWIFT Link 5
 Telex Message Processing 6
MERVA ESA resources 157, 175
MERVA OS/2
 MLSNDNRM queue 68, 138
MERVA USE & Branch
 components located on 6
 customizing MERVA Link of 38
 overview 6
MERVA USE & Branch components
 MERVA Base 6
 MERVA Link 6
message format service
 MERVA Link 157
 MERVA-MQI Attachment 175
message integrity 5
message queues 70, 141
message queues (MERVA AIX)
 MLSNDNRM 68, 138
message queues (MERVA ESA)
 BKEUPDT 68, 138
 L1ACK 68, 138
 L1DO0 68, 138
 L1SDO 68, 138
 L1VE0 68, 138
 SLSLOAD 65, 135
 SLSLRECV 67, 137
 TX2USESQ 67, 68, 137, 138
message queues (MERVA OS/2)
 MLSNDNRM 68, 138
message security 4, 5, 111

message transfer node name 12, 19, 71, 87
 message transfer process (MTP)
 name 12, 88
 MFS
 program table definition
 MERV Link 157
 MERV-MQI Attachment 175
 MFS program table
 MERV Link 149, 157
 MERV-MQI Attachment 166, 175
 MLRECEIV 70
 MLSNDLOW 70
 MLSNDNRM 70
 MLSNDURG 70
 mode name 11, 15, 87
 MPP job, IMS
 MERV-MQI Attachment 191
 MQSeries
 examples 167
 MQSeries, customizing 92
 MQSeries receiver channel
 examples 101
 MQSeries remote queue
 examples 98
 MQSeries resources, defining under NT
 for MVS 193
 for VSE 197
 MQSeries sender channel
 examples 99
 MQSND1 141
 MQSND2 141
 MQWAIT 141
 MT 999 65, 134
 MT node name 12, 87
 MTN name 19, 71
 MTP 12, 19, 72, 88
 MTP, local 19, 38, 72
 MTP, partner 19, 38
 MTP example
 APPC/IMS 156
 APPC/MVS 156
 CICS 156

N

network ID 11, 28, 86
 network ID of MERV ESA 15
 Notices 205
 NTMVSATT.ATN 202
 NTMVSATT.MQ 193
 NTVSEATT.ATN 203
 NTVSEATT.MQ 197

O

offline utility program (DFHCSDUP) 16, 91
 outgoing messages, whether
 authenticated 19, 72
 outgoing messages, whether
 authenticated and encrypted 12, 88

P

parameter LU name 15

parameters
 APPC LU name (on workstation) 11, 87
 APPC/MVS base LU 14, 89
 application support process name 12, 88
 ASP name of MERV AIX 12, 88
 ASP name of MERV ESA 12, 88
 ASP name of MERV OS/2 12, 88
 control queue name 12, 88
 conversation security 14
 destination address (of LAN) 11, 86
 destination name, symbolic 11, 87
 EKATPO1 14, 89
 external message transfer process
 (MTP) name 12, 88
 format specification 13, 89
 host queue manager 86
 incoming report routing 13
 internal message transfer process
 (MTP) 12, 88
 ISC=YES 13, 89
 journaling 13
 LAN destination address 11, 86
 local ASP name 12, 88
 local MTP name 12, 88
 local node ID 11, 86
 local node name 11, 13, 86
 LU name of CICS system 11, 86
 LU name of workstation 11, 14, 87, 89
 master logical terminal name 13
 MERV Link control queue 12, 88
 MERV Link receive queue (PC) 12, 88
 MERV Link send queue
 (MERV ESA) 12, 88
 MERV Link send queue (PC) 12, 88
 MERV Link send queue
 (RS/6000) 12, 88
 MERV Link send queue cluster 13
 message transfer node name 12, 87
 message transfer process (MTP)
 name 12, 88
 mode name 11, 14, 87, 89
 MT node name 12, 87
 MTP name of MERV AIX 12, 88
 MTP name of MERV ESA 12, 88
 MTP name of MERV OS/2 12, 88
 name of the workstation 12
 network ID 11, 86
 node name 11, 86
 partner ASP name 12, 88
 partner LU name (on
 workstation) 11, 86
 partner MTN name 12, 87
 partner MTP name 12, 88
 partner network ID 11, 86
 PC network ID 11
 physical unit name 11, 86
 pregenerated session key queue
 (FIN) 12, 88
 pregenerated session key queue
 (GPA) 12, 88
 queue manager, host 86
 receive queue name (PC) 12, 88

parameters (*continued*)

received messages routing table 13, 89
 receiving MTP name of
 MERV ESA 13, 89
 receiving TP file name 11, 87
 receiving TP name 11, 14, 87
 receiving transaction ID 13, 89
 relationship between different
 (parameters) 43, 112
 routing table (received messages) 13, 89
 routing table (session key
 requests) 13, 89
 routing table (Telex Link station
 command) 13
 security 12, 88
 send queue name (MERV ESA) 12, 88
 send queue name (PC) 12, 88
 send queue name (RS/6000) 12, 88
 sending MTP name of
 MERV ESA 13
 sending TP name 14
 sending transaction ID 14
 sending transaction ID of
 MERV ESA 13
 session key routing table 13, 89
 SNA network ID 11
 SWIFT format specification 13, 89
 symbolic destination name 11, 87
 SYSIDNT 13, 89
 system identification of CICS 13, 89
 Telex Link station command routing
 table 13
 TP profile name 14
 transaction program profile name 14
 transmission format 13, 89
 user-exit program number or
 name 12, 88
 VTAM logon mode entry 14, 89
 workstation name 12
 partner ASP name 38, 71
 partner LU
 name (on WIndows NT) 38
 partner MTN name 12, 71, 87
 partner MTP name 12, 38, 88
 partner network ID 11, 15, 86
 partner table
 EKAPT header 155
 partner table example
 APPC/IMS 156
 APPC/MVS 156
 CICS 156
 telex 155
 USE 155
 PC network ID 11, 86
 PCT (program control table) 16, 91
 physical unit name 11, 15, 86
 PPT (processing program table) 17, 91
 pregenerate (session keys) 65, 134
 pregenerated session key queue
 (FIN) 12, 19, 88
 pregenerated session key queue
 (GPA) 12, 19, 88
 processing program table (PPT) 17, 91
 MERV Link 149

processing program table (PPT)
(continued)
 MERVA-MQI Attachment 165
 program control table (PCT) 16, 91
 MERVA Link 149
 MERVA-MQI Attachment 165
 program number, user-exit 19, 72
 program number or name, user-exit 12, 88
 protocol, message security 4

R

receive queue name (PC) 12, 88
 received messages routing table 13, 16, 19, 72, 89
 receiver channel, MQSeries
 examples 101
 receiving MTP name of
 MERVA ESA 13, 16, 89
 receiving TP file name 11, 87
 receiving TP name 11, 14, 18, 19, 71, 87
 receiving transaction ID 13, 16, 89
 relationship between parameters 43, 112
 remote link address 15
 remote queue, MQSeries
 examples 98
 Resource Definition Online (RDO) 17
 routing table example for USE 179
 routing table
 DWSL1IN 68, 138
 DWSL1OUT 68, 138
 DWSL2IN 68, 138
 DWSL2OUT 68, 138
 DWSL3FIH 68, 138
 DWSL3FIO 68, 138
 DWSL3GPI 68, 138
 DWSL3GPO 68, 138
 DWSRTSK 67, 137
 EKARTTXU 65, 67, 68, 73, 78, 135, 137, 138
 ENLR2AI0 76
 ENLR2DE0 76
 ENLR2TKC 76
 ENLR2TKV 78
 ENLR2TLC 73
 ENLR2VE0 76
 routing table example for telex 160
 routing table example for telex and USE 179

S

SDLC
 sample configuration 148
 Secure Login and Select 5
 security 39
 security, message 4, 5, 111
 security parameter 12, 19, 72, 88
 send queue cluster, MERVA Link 13, 72
 send queue name 12, 19, 88
 send queue name (PC)
 send queue (RS/6000) 12, 88
 send queue name (RS/6000) 12, 88
 sender channel, MQSeries
 examples 99

sending MTP name of MERVA ESA 13, 16
 sending TP name 14, 17
 sending transaction ID 14, 17
 sending transaction ID of
 MERVA ESA 13, 16, 19, 72
 session key queue (FIN) 12, 19, 88
 session key queue (GPA) 12, 19, 88
 session key routing table 13, 15, 19, 89
 session keys (pregenerated) 65, 134
 session keys, sending with MT 999 65, 134
 SLS 5, 6
 description of service 5
 example function table
 MERVA Link 158
 PC name 19
 RS/6000 name 19
 sending pregenerated keys to
 MERVA ESA 65, 134
 value of TFLAG parameter 21, 105
 workstation name for master LT 12
 SNA parameters
 exchange identification 11, 86
 local node name 11, 86
 network ID 11, 15, 28, 86
 PC network ID 11, 86
 physical unit name 11, 86
 token ring address 11, 86
 XID 11, 86
 SNASVCMG 146
 standard routing, changing with
 Customization program 38
 startup job, CICS/ESA
 MERVA Link 164
 MERVA-MQI Attachment 190
 startup job, IMS
 MERVA-MQI Attachment 191
 startup jobs
 MERVA Link 164
 MERVA-MQI Attachment 190
 SWIFT format specification 13, 89
 SWIFT Link 3
 SWIFT Link of MERVA ESA
 USE services supported by 5
 SWIFT messages
 format specification for 5
 symbolic destination name 11, 87
 SYSIDNT 13, 16, 19, 72, 89
 system identification of CICS 13, 16, 19, 72, 89

T

TCT (terminal control table) 17
 telex functions
 TX2ACK 77
 TX2AI0 76
 TX2CTL 80
 TX2DE0 76
 TX2DISTR 79
 TX2ERROR 80
 TX2HOLD 80
 TX2INVR 79
 TX2NAK 77
 TX2NOTX 77
 TX2NRM 80

telex functions *(continued)*

TX2PDR 78
 TX2PR0 77
 TX2PR1 79
 TX2RCV 78
 TX2SDI 77
 TX2SDO 79
 TX2SDY 79
 TX2TKC 77
 TX2TKCA 77
 TX2TKCER 77
 TX2TKV 79
 TX2TKVER 79
 TX2URG 80
 TX2USECQ 80
 TX2USESQ 80
 TX2VE0 76
 TX2WAIT 80
 Telex Link Connector 3
 Telex Link Station 3
 Telex Link station command routing
 table 13, 16, 72
 Telex Message Processing 3, 6
 telex messages
 format specification for 5
 Telex scenario 161
 terminal control table (TCT) 17
 MERVA Link 150
 terminal definition, CICS 17
 token ring
 sample configuration 147
 Token Ring
 communications controller
 address 11, 86
 TP Profile, partner 38
 TP profile name 14, 18
 transaction ID 14
 transaction program, specifying 38
 transaction program definition 11, 19, 71, 87
 transaction program profile name 14, 18
 transmission format 13, 19, 72, 89
 TX2ACK 77
 TX2AI0 76
 TX2CTL 80
 TX2DE0 76
 TX2DISTR 79
 TX2ERROR 80
 TX2HOLD 80
 TX2INVR 79
 TX2NAK 77
 TX2NOTX 77
 TX2NRM 80
 TX2PDR 78
 TX2PR0 77
 TX2PR1 79
 TX2RCV 78
 TX2SDI 77
 TX2SDO 79
 TX2SDY 79
 TX2TKC 77
 TX2TKCA 77
 TX2TKCER 77
 TX2TKV 79
 TX2TKVER 79
 TX2URG 80
 TX2USECQ telex function 80

TX2USESQ 80
TX2VE0 76
TX2WAIT 80

U

UFROMSWF 70
UNAKED 70
USE background process 70
USE functions 3, 5
USE only scenario
 MERVA Link 163
 MERVA-MQI Attachment 188
USEINCMD 70
user-exit program number 19, 72
user-exit program number or name 12,
 15, 88
user exits 5
User Security Enhancements 5
utility program DFHCSDUP 17, 91
UWOPREAG 70

V

VTAM
 application name 11, 86
 customizing 15
 logon mode entry 11, 14, 15, 19, 72,
 87, 89
 sample definitions 145

W

workstation based functions 3
 MERVA ESA components used 3

X

XID 11, 86

Readers' Comments — We'd Like to Hear from You

MERVA ESA Components
Workstation Based Functions
Version 4 Release 1

Publication No. SH12-6383-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Information Development, Dept. 0446
Schoenaicher Strasse 220
71032 Boeblingen
Germany

Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5648-B30

SH12-6383-01



Spine information:



MERVA ESA Components

Workstation Based Functions

Version 4
Release 1