BASEstar Open Device Connectivity

Device Access Software for Siemens 3964 User Guide

Order Number: AA-QT1RB-TE

April 1997

This document describes the Device Access Software (DAS) for Siemens 3964 which allows BASEstar Open users and DEComni API applications to exchange data with Siemens devices supporting the Siemens 3964 protocols.

Revision/Update Information:	This is a new document for the current release.
Operating System and Version:	BASEstar Open is available on a broad range of both hardware and software platforms. Refer to your *Software Product Description* for precise information.
Software Version:	3.1

Digital Equipment Corporation
Maynard, Massachusetts

First printing, February 1996
Revised, July 1996
Revised, April 1997

© Digital Equipment Corporation 1997.

All Rights Reserved.

Possession, use, or copying of the software described in this documentation is authorized only pursuant to a valid written license from DIGITAL or an authorized sublicensor.

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation: Alpha AXP, BASEstar Open, DEC, DECmessageQ, DECnet, DECnet-DOS, DECosap, DEComni, DIGITAL, DIGITAL UNIX, FMS, LN03, MicroVAX, NAS, OpenVMS, OpenVMS Alpha, PATHWORKS, PDAS, Rdb/VMS, ReGIS, ThinWire, TK, ULTRIX, VAX, VAXcluster, VAX COBOL, VAX FORTRAN, VAX Pascal, VAX RMS, VMS/ULTRIX Connection, VT, and the DIGITAL logo.

The following are third-party trademarks:

MS, Microsoft, and MS-DOS are registered trademarks of Microsoft Corporation.
UNIX is a registered trademark licensed exclusively by X/Open Company Ltd.
Windows and Windows NT are trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective holders.

Table of Contents

Preface..... vii

1 Introduction 1–1

Communicating With Siemens 3964 Devices..... 1–1
Features of the DIGITAL DAS for Siemens 3964 1–2
What's Next?..... 1–3

2 Installing the DAS for Siemens 3964 on Your Platform 2–1

Installation on DIGITAL UNIX Systems 2–2
Preparing for DAS for Siemens 3964 Installation..... 2–2
Installing the DAS for Siemens 3964 2–4
Getting Help and Reporting Problems 2–4
Installation on Windows NT Systems..... 2–6
Preparing for DAS for Siemens 3964 Installation..... 2–6
Installing the DAS for Siemens 3964 2–6
Deinstalling the DAS for Siemens 3964..... 2–7
Getting Help and Reporting Problems 2–7
Installation on OpenVMS Systems..... 2–8
Preparing for DAS for Siemens 3964 Installation..... 2–8
Installing the DAS for Siemens 3964 2–9
Getting Help and Reporting Problems 2–10

PART I Using BASEstar Open to Communicate With Siemens 3964 Devices

3 Overview..... 3–1

BASEstar Open Environment Components..... 3–2
Exchanging Data Messages With a Siemens 3964 Device (3964[R] Protocol)..... 3–4
Sending Data 3–4
Receiving Data 3–4
Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol) 3–6
Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol) 3–7
Datatypes and Memory Block Sizes 3–7

4 Configuring BASEstar Open 4–1

Registering ODS Entries..... 4–2

Creating Application Services Objects 4–2
Creating Device Services Objects 4–2
Creating Protocol_Profiles 4–2
Creating VMDs 4–3
Creating Variables 4–5
Creating Data Services Objects 4–6
Creating Device_Data_Points for Communication Via the 3964[R] Protocol 4–6
Creating Device_Data_Points for Communication Via the 3964[R]_512 Protocol 4–7

5 Examples 5–1

Example 1: Using the 3964[R] Protocol to Exchange Data 5–2
Example Overview 5–2
Before Running the Example 5–2
CLI Source 5–2
Example 2: Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol) 5–5
Example Overview 5–5
Before Running the Example 5–5
CLI Source 5–5
Example 3: Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol) 5–7
Example Overview 5–7
Before Running the Example 5–7
CLI Source 5–7

PART II Using the DEComni API to Communicate With Siemens 3964 Devices

6 Overview 6–1

DEComni API Environment Components 6–2
Exchanging Data Messages With a Siemens 3964 Device (3964[R] Protocol) 6–3
Receiving Data 6–3
Sending Data Messages 6–4
Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol) 6–4
Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol) 6–4
MMS Types and Siemens 3964 Memory Block Sizes 6–5

7 Configuring DEComni API 7–1

Registering ODS Entries 7–2
Setting the Application Profile 7–2
Creating VMD Definitions 7–2
Using the 3964[R] Protocol to Create a VMD for Communication 7–2
Using the 3964[R]_RK512 Protocol to Create a VMD for Communication 7–3
Creating Variable Definitions 7–3
Using the 3964[R] Protocol to Create Variables for Communication 7–3
Using the 3964[R]_RK512 Protocol to Create Variables for Communication 7–3

8 Programming 8-1

Programming With the 3964[R] Protocol 8-2
Opening a Connection 8-2
Sending Data Messages 8-2
Receiving Data Messages 8-2

Programming With the 3964[R]_RK512 Protocol 8-3
Opening and Closing a Connection 8-4
Reading and Writing Memory Blocks of a Siemens Device 8-4
Fulfilling Requests for an Emulated Device 8-4

PART III Appendices

A ODS Entry Attributes for the Siemens 3964 Protocol A–1

Creating ODS Entries for the Siemens 3964 Protocol A–2
Configuring a LAT Port A–3
OpenVMS Platform A–3
DIGITAL UNIX Platform A–3

B BASEstar Open Datatype Sizes B–1

C DEComni MMS Type Sizes C–1

E DEComni API Network IOSB Messages E-1

Figures

Figure 1-1: Software Components for Siemens 3964 Communication 1–1
Figure 1-2: Communicating With Siemens 3964 Devices..... 1–2
Figure 3-1: BASEstar Open Environment Components..... 3–2
Figure 3-2: BASEstar Open Objects and Operations (3964[R] Protocol) 3–4
Figure 3-3: Using BASEstar Open to Access the Memory of a Siemens 3964 Device 3–6
Figure 3-4: Using BASEstar Open to Emulate a Siemens 3964 Device..... 3–7
Figure 6-1: DEComni API Environment Components..... 6–2
Figure 6-2: DEComni Definitions and API Procedures (3964[R] Protocol)..... 6–3
Figure 6-3: Using the DEComni API to Access the Memory of a Siemens 3964 Device 6–4
Figure 6-4: Using the DEComni API to Emulate a Siemens 3964 Device..... 6–5

Tables

Table 1-1: Supported Siemens 3964 Protocols..... 1–2
Table 2-1: Pre-requisite Software for DIGITAL UNIX (BASEstar Open Environment)..... 2–3
Table 2-2: Pre-requisite Software for DIGITAL UNIX (DEComni API Environment)..... 2–3
Table 2-3: Pre-requisite Software for Windows NT 2–6
Table 2-4: Pre-requisite Software for OpenVMS (BASEstar Open Environment)..... 2–8
Table 2-5: Pre-requisite Software for OpenVMS (DEComni API Environment)..... 2–9
Table 2-6: Transport Network Software 2–9
Table 4-1: Protocol_Profile Attributes..... 4–2
Table 4-2: VMD Attributes..... 4–3
Table 4-3: Reception Variable Attributes (3964[R] Protocol) 4–5
Table 4-4: Transmission Variable Attributes (3964[R] Protocol)..... 4–5
Table 4-5: Variable Attributes (3964[R]_RK512 Protocol)..... 4–6
Table 4-6: Device_Data_Point Attributes for Receive Operations (3964[R] Protocol)..... 4–6
Table 4-7: Device_Data_Point Attributes for Send Operations (3964[R] Protocol) 4–7
Table 4-8: Device_Data_Point for Send and Receive Operations (3964[R]_512 Protocol) 4–7
Table 7-1: DEComni Application Profiles..... 7–2
Table 7-2: DEComni VMD Attributes 7–2
Table 7-3: DEComni Variable Attributes 7–3
Table 8-1: DEComni API Procedures for Data Exchange (3964[R] Protocol)..... 8-2
Table 8-2: DEComni API Procedures for Data Exchange (3964[R]_512 Protocol) 8-3

Examples

Example 4–1: Creating Device Services Protocol_Profiles 4–3
Example 4–2: Creating a Device Services VMD (3964[R] Protocol) 4–4
Example 4–3: Creating Device Services VMDs (3964[R]_RK512 Protocol)..... 4–4
Example 5–1: Creating BASEstar Open Objects (3964[R] Protocol) 5–2
Example 5–2: Exchanging Data with a Siemens 3964 Device (3964[R] Protocol)..... 5–4
Example 5–3: Creating BASEstar Open Objects (3964[R]_RK512 Protocol) 5–5
Example 5–4: Reading and Writing the Device Memory (3964[R]_RK512 Protocol)..... 5–6
Example 5–5: Parameter File (3964[R]_RK512 Protocol) 5–7
Example 5–6: Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol)..... 5–7
Example 5–7: Reading and Writing the Emulated Device Memory (3964[R]_RK512 Protocol) 5–9
Example A–1: ODS Entry for LAT Port A–2

Preface

This document describes how to install and use the DIGITAL Device Access Software (DAS) for Siemens 3964, the software that allows BASEstar Open users and DEComni API applications to exchange data with Siemens 3964 devices that support the Siemens 3964 protocols.

Multiplatform Applicability

Most of the information in this manual applies regardless of the operating system platform on which the DAS for Siemens 3964 software is installed. Information relating to installation is typically platform-dependent.

Intended Audience

This manual is addressed to:

- System managers who have to install and maintain the DAS for Siemens 3964 on any of the supported platforms.
- Application developers who have to configure and use BASEstar Open, or the DEComni API (or both) to exchange data with Siemens 3964 devices.

Structure of this Document

This document is organized as follows:

- Chapter 1 introduces you to the DAS for Siemens 3964; Chapter 2 contains the installation procedures.
- Part I describes how Siemens 3964 devices are accessed via BASEstar Open services, how you must configure BASEstar Open for use with the DAS for Siemens 3964, and finally provides code examples that you can run in the BASEstar Open environment.
- Part II describes how to access Siemens 3964 devices via the DEComni API, how to configure DEComni API for use with the DAS for Siemens 3964, and how to exchange data with the remote devices.
- Part III contains the appendices.

For More Information

If you are using BASEstar Open services you will find related information in the following documents:

- *BASEstar Open Introduction*
- *BASEstar Open Reference Guide*
- *BASEstar Open Command Language Interface*
- *BASEstar Open Application Programming Interface*
- *BASEstar Open Messages*
- Platform-specific management guides

If you are using DEComni API you will find related information in the following document:

- *DEComni API and DEComni MMS User Guide*
- *DEComni API Guide to Using Omni Directory Services* (for supported platforms)
- *DEComni API Omni Definition Facility* User Guide (for supported platforms)
- *DEComni API Guide to Using OmniView* (for supported platforms)

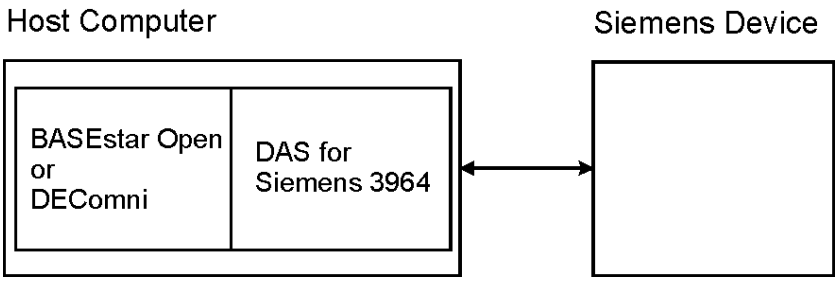
1
Introduction

The DIGITAL Device Access Software (DAS) for Siemens 3964 allows both BASEstar Open users and DEComni API applications to access Siemens 3964 devices that support the 3964 protocols.

Communicating With Siemens 3964 Devices

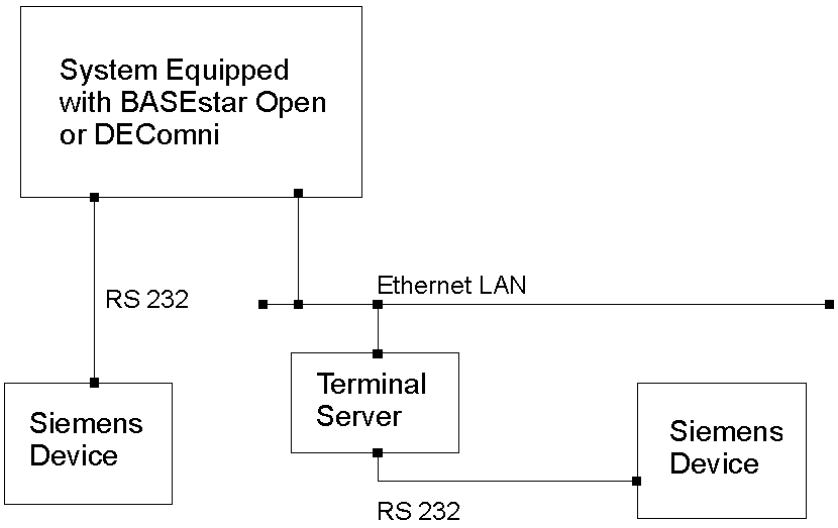
Figure 1-1 shows how a system equipped with the BASEstar Open or the DEComni API product can communicate with a Siemens 3964 device using the Siemens 3964 protocols. The system equipped with BASEstar Open or DEComni API is an OpenVMS or DIGITAL UNIX system.

Figure 1-1: Software Components for Siemens 3964 Communication



As illustrated in Figure 1-2, a system (and thus BASEstar Open users or DEComni API applications) can communicate with Siemens 3964 devices connected to a RS-232 port or to a terminal server via an Ethernet LAN. The system can connect several Siemens 3964 devices by means of a point-to-point connection.

Figure 1-2: Communicating With Siemens 3964 Devices



Features of the DIGITAL DAS for Siemens 3964

The DIGITAL DAS for Siemens 3964 allows BASEstar Open users and DEComni API applications to:

- Exchange data messages with a Siemens 3964 device using the Siemens 3964[R] protocol.
- Access memory blocks on a Siemens 3964 device using the Siemens 3964[R]_RK512 protocol.

The main features of the supported protocols are listed in Table 1-1.

In addition, a system equipped with BASEstar Open or DEComni API can emulate Siemens 3964 devices that accept and fulfill requests received from remote devices via the Siemens 3964[R]_RK512 protocol.

Table 1-1: Supported Siemens 3964 Protocols

Protocol Version	Features
Siemens 3964	The Siemens 3964 protocol allows the exchange of data messages between your BASEstar Open or DEComni API system and a Siemens device.
Siemens 3964R	The Siemens 3964R protocol is like the Siemens 3964 protocol but the Block Check Character (BCC) is also supported.
Siemens 3964_RK512	The RK512 interpreter is layered above the Siemens 3964 protocol. This allows direct and named access to device memory blocks.
Siemens 3964R_RK512	Same as Siemens 3964_RK512 but the RK512 interpreter is layered above the Siemens 3964R protocol.

What's Next?

In this chapter you have learnt about the features of the DIGITAL DAS for Siemens 3964.

If you intend to use the DAS for Siemens 3964 via BASEstar Open services, first read Chapter 2 to know how to install the DAS for Siemens 3964 on your system, and then read Part I of this manual to know how you must configure BASEstar Open and use its services to exchange data with Siemens 3964 devices.

If you intend to use the DAS for Siemens 3964 via the DEComni API, read Chapter 2 for information about how to install the DAS for Siemens 3964 on your system. Then read Part II of this manual for details of how to configure the DEComni API and use the API procedures to exchange data with Siemens 3964 devices.

2

Installing the DAS for Siemens 3964 on Your Platform

This chapter specifies the requirements and the procedures for installing the DAS for Siemens 3964 on the following platforms:

- DIGITAL UNIX
- Windows NT
- OpenVMS

The chapter provides the instructions for installing the DAS for Siemens 3964 in the BASEstar Open and DEComni API environments.

Installation on DIGITAL UNIX Systems

The installation requirements vary depending on whether you are installing the DAS for Siemens 3964 for use via BASEstar Open services or via the DEComni API.

Preparing for DAS for Siemens 3964 Installation

Before starting the installation procedure, complete the preparation requirements outlined in this section.

Release Notes

Your documentation includes the Device Access Software for Siemens 3964 Release Notes, which you should read before installing and using the product. The release notes may contain information about changes to the application.

After you install the Device Access Software for Siemens 3964 software, you can also access the online release notes in the form of an ASCII text file by entering the following command:

```
# more /usr/opt/DASS3964_310/doc/relnotes.txt
```

License Registration

The Device Access Software for Siemens 3964 includes support for the DIGITAL UNIX License Management Facility (LMF). A License Product Authorization Key (License PAK) must be registered in the License Database (LDB) in order to use the Device Access Software for Siemens 3964 on a newly-licensed node. The License PAK may be shipped along with the kit if you ordered the license and media together; otherwise, it is shipped separately to a location based on your license order.

If you are installing the Device Access Software for Siemens 3964 as an update on a node already licensed for this software, you have already completed the License PAK registration requirements.

If you are installing pre-requisite or optional software along with the Device Access Software for Siemens 3964, review the PAK status and install the PAKs for any pre-requisite or optional software before you install the Device Access Software for Siemens 3964.

To register a license under the DIGITAL UNIX system, first log in as `superuser`.

You then have a choice of two ways to perform the PAK registration in the License Database (LDB):

- Before installing the Device Access Software for Siemens 3964

  At the superuser prompt, edit an empty PAK template with the `lmf register` command and include all the information on your License PAK as follows:

  ```
  # lmf register
  ```

- After installing the Device Access Software for Siemens 3964

  At the superuser prompt, edit the partially completed PAK template in `/usr/var/adm/lmf/`*`template`* with the `lmf register` command to add your unique License PAK information as follows:

  ```
  # lmf register - < /usr/var/adm/lmf/DAS3964
  ```

  After you register your license, use the following `lmf reset` command to copy the license details from the License Database (LDB) to the kernel cache:

  ```
  # lmf reset
  ```

  For complete information on using the DIGITAL UNIX License Management Facility, see the *DIGITAL UNIX Guide to Software Licensing* or the `lmf(8)` reference page.

Software Requirements (BASEstar Open Environment)

If you are installing the DAS for Siemens 3964 in the BASEstar Open environment, check that the software products listed in Table 2-1 have been installed appropriately on your system.

Table 2-1: Pre-requisite Software for DIGITAL UNIX (BASEstar Open Environment)

Subset Name and Description	Comments
OSFLAT200 DEC TCP/IP Services for DIGITAL UNIX Version 3.1	Install this product to make LAT support available to DAS for Siemens 3964.
BSTR310 BASEstar Open Server for DIGITAL UNIX Version 3.1	Ensure that the BASEstar Open Device Connectivity (DEComni) has been installed. In particular, check for the presence of the following subsets: • DOUBASE310 • DASTK232310 • DASTKRUN310 See the *BASEstar Open Server Inst. and Management Guide* for your platform for details.

Software Requirements (DEComni API Environment)

If you are installing the DAS for Siemens 3964 in the DEComni API environment, check that the software products listed in Table 2-2 have been installed on your system.

Table 2-2: Pre-requisite Software for DIGITAL UNIX (DEComni API Environment)

Subset Name and Description	Comments
DOUBASE310 DEComni API for DIGITAL UNIX Version 3.1	In particular, check for the presence of the following subsets: • DASTK232310 • DASTKRUN310 See the *DEComni API Installation Guide* for your platform for details.
OSFLAT200 DEC TCP/IP Services for DIGITAL UNIX Version 3.1	Install this product to make LAT support available to DAS for Siemens 3964.

Disk Space Requirements

The following table lists the disk space required to install the DAS for Siemens 3964.

Usage (Kbytes)	(root)	/usr/opt	/var/opt
Installation	0	1103	381
Permanent	0	1103	339

Backing Up Your System Disk

DIGITAL recommends that you back up your system disk before installing any software. For information about how to perform a system disk backup, refer to your DIGITAL UNIX documentation.

Installing the DAS for Siemens 3964

This section describes the installation, deinstallation and IVP procedures for the Device Access Software for Siemens 3964 for DIGITAL UNIX. Installing the Device Access Software for Siemens 3964 and running the Installation and Verification Procedure (IVP) on your DIGITAL UNIX system takes approximately 5 to 10 minutes.

Installation Procedure

Install the DAS for Siemens 3964 as follows:

1. Log onto the system as superuser (root).
2. Mount the distribution media on the desired location (for example, mount `-dr /dev/rz4c /CDROM`).
3. Change your working directory to the kit location (for example, `cd /CDROM/DASS3964310`).
4. Issue the *`setld -l`* command to load the product onto the system, and follow the installation dialogs (for example, `setld -l DASS3964310`).

Running the IVP

During the installation procedure you can run the IVP as part of the installation. If you did not run the IVP at that time or the product does not work correctly, you can run the IVP at any time by issuing the following command:

```
setld -v DASS3964310
```

Getting Help and Reporting Problems

If an error occurs while you are using the DAS for Siemens 3964, and you believe the error is the result of a problem associated with the product, take one of the following actions:

- If you have a basic or DECsupport Software Agreement, call your Customer Support Center (CSC). The CSC provides telephone support for high-level advisory and remedial assistance.
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).
- If you purchased the product within the last 90 days and you think the problem is casued by a software error, you can submit an SPR.

If you submit an SPR, please take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include the description and version number of the product. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, include files, relevant data files and so forth.
4. Provide a list of the program.
5. If the program is longer than 50 lines, submit a copy of it on machine-readable media (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application.
6. Report only one program per SPR. This will facilitate a faster response.
7. Mail the SPR package to DIGITAL.

Experience shows that many SPRs do not contain enough information to duplicate or identify the problem. Concise, complete information helps DIGITAL give timely and accurate service to software problems.

If you find an error in the documentation, send an electronic mail message to *manufacturing@digital.com*.

Installation on Windows NT Systems

Device Access Software features for Siemens 3964 can only be accessed via BASEstar Open (the DEComni API is not currently available).

Preparing for DAS for Siemens 3964 Installation

Before starting the installation procedure, you should complete the preparation requirements outlined in this section.

License Registration

The installation requests a software key. You must enter this code in order to make the DAS for Siemens 3964 Version 3.1 available on your system.

For further information about the software key, refer to the *DAS for Siemens 3964 Cover Letter*.

Software Requirements (BASEstar Open Environment)

Check that the software products listed in Table 2-3 have been installed on your system.

Table 2-3: Pre-requisite Software for Windows NT

Subset Name and Description	Comments
BASEstar Open Server for Windows NT Version 3.1	Ensure that the BASEstar Open Device Connectivity has been installed. See the *BASEstar Open Server Inst. & Management Guide* for your platform for details.

Disk Space Requirements

The files copied under the installation directory by the installation procedure occupy 48 Kbytes on a Windows NT Intel system and 51 Kbytes on a Windows NT Alpha system.

Backing UpYour System Disk

DIGITAL recommends that you back up your system disk before installing any software. For information about how to perform a system disk backup, refer to your Windows NT documentation.

Installing the DAS for Siemens 3964

To install Device Access Software for Siemens 3964 on a Windows NT system, you must be a member of the Administrators group.

To initiate the installation procedure, run the SETUP.EXE file from diskette 1 of the distribution set. The Device Access Software for Siemens 3964 installation procedure allows you to specify a the installation directory. Installing the Device Access Software for Siemens 3964 on your Windows NT system takes about one minute.

Files Installed on Your System

The directories and files created by the Device Access Software for Siemens 3964 installation proccedure are as follows:

<install_dir>\3964\BIN\das3964.dll

<install_dir>\3964\INCLUDE\omnii4.h

<install_dir>\3964\LIB\3964.fsm

<install_dir>\3964\LIB\3964r.fsm

<install_dir>\3964\LIB\das3964.lib

<install_dir>\3964\relnotes.txt

Deinstalling the DAS for Siemens 3964

To deinstall the Device Access Software for Siemens 3964, double click on the "uninstaller DAS for Siemens 3964" icon in the BASEstar Open Program Manager.

Getting Help and Reporting Problems

If an error occurs while you are using the DAS for Siemens 3964, and you believe the error is the result of a problem associated with the product, take one of the following actions:

- If you have a basic or DECsupport Software Agreement, call your Customer Support Center (CSC). The CSC provides telephone support for high-level advisory and remedial assistance.
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).
- If you purchased the product within the last 90 days and you think the problem is casued by a software error, you can submit an SPR.

If you submit an SPR, please take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include the description and version number of the product. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, include files, relevant data files and so forth.
4. Provide a list of the program.
5. If the program is longer than 50 lines, submit a copy of it on machine-readable media (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application.
6. Report only one program per SPR. This will facilitate a faster response.
7. Mail the SPR package to DIGITAL.

   Experience shows that many SPRs do not contain enough information to duplicate or identify the problem. Concise, complete information helps DIGITAL give timely and accurate service to software problems.

   If you find an error in the documentation, send an electronic mail message to *manufacturing@digital.com*.

Installation on OpenVMS Systems

The information that follows is valid for both Alpha and VAX architectures unless otherwise stated.

The installation requirements vary depending on whether you are installing the DAS for Siemens 3964 for use via BASEstar Open services or via the DEComni API.

Preparing for DAS for Siemens 3964 Installation

Installing the Device Access Software for Siemens 3964 and running the Installation Verification Procedure (IVP) on your OpenVMS system takes approximately 3 minutes.

Before starting the installation procedure you should complete the preparation requirements outlined in this section.

License Registration

You must use the License Management Facility (LMF) to register the Device Access Software for Siemens 3964 in accordance with the license agreement signed for your site before you can run either the Installation Verification Procedure (IVP) or the software. License registration information is included in the Product Authorization Key (PAK) that is shipped with the Device Access Software for Siemens 3964. The PAK contains licensing information that should be registered *before* you start the installation.

To register the license, log into the SYSTEM account and set your default to SYS$UPDATE. You can use either of the following two methods to register the license under LMF:

- Invoke the following procedure and enter the data supplied by the PAK:

  ```
  $ SYS$UPDATE:VMSLICENSE.COM
  ```

- Enter the following command, using the qualifiers that specify the data supplied by the PAK:

  ```
  $ LICENSE REGISTER DAS3964 /qualifier,...
  ```

If you plan to use the Device Access Software for Siemens 3964 on more than one node of a VAXcluster system, you must load the license on each of the other nodes after you have completed this one.

For complete information about using the OpenVMS License Management Facility, refer to the *OpenVMS License Management Utility Manual*.

Software Requirements (BASEstar Open Environment)

Before starting the installation procedure, you should complete the preparation requirements outlined in this section.

If you are installing the DAS for Siemens 3964 in the BASEstar Open environment, check that the software products listed in Table 2-4 and Table 2-5 have been installed and are functioning on your system.

Table 2-4: Pre-requisite Software for OpenVMS (BASEstar Open Environment)

Subset Name and Description	Comments
BASEstar Open Server for OpenVMS Version 3.1	Ensure that the BASEstar Open Device Connectivity (DEComni, including the DASware RS232 facility) has been installed. See the *BASEstar Open Server Inst. & Management Guide* for your platform for details.

Table 2-5: Pre-requisite Software for OpenVMS (DEComni API Environment)

Subset Name and Description	Comments
BASEstar Open Server for OpenVMS Version 3.1	Ensure that the BASEstar Open Device Connectivity (DEComni, including the DASware RS232 facility) has been installed. See the *BASEstar Open Server Inst. & Management Guide* for your platform for details.

If you do not wish to use a direct connection to the device, make sure that the appropriate transport network software indicated in Table 2-6 has been installed and configured.

Table 2-6: Transport Network Software

Subset Name and Description	Comments
RS232	LAT (supplied with VMS)

Disk Space Requirements

The DAS for Siemens 3964 on OpenVMS occupies approximately 500 blocks.

Backing UpYour System Disk

DIGITAL recommends that you back up your system disk before installing any software. For information about how to perform a system disk backup, refer to your OpenVMS documentation.

Installing the DAS for Siemens 3964

This section describes the installation, deinstallation and IVP procedures for the Device Access Software for Siemens 3964 for OpenVMS.

Use the following command to install the DAS for Siemens 3964:

```
$ @SYS$UPDATE:VMSINSTALL DAS3964A310 ddcu
```

where *ddcu* is the installation directory.

Files Installed on Your System

This section lists directories and files created by the Device Access Software for Siemens 3964.

SYS$COMMON:[SYSLIB]

AS3964_SHR.EXE

OMNI_INTEGRATOR4_DEFS_INCLUDE.H

OMNI_INTEGRATOR4_DEFS_INCLUDE.H

OMNI_INTEGRATOR4_DEFS_INCLUDE.ADA

OMNI_INTEGRATOR4_DEFS_INCLUDE.PAS

OMNI_INTEGRATOR4_DEFS_INCLUDE.BAS

OMNI_INTEGRATOR4_DEFS_INCLUDE.FOR

OMNI_INTEGRATOR4_DEFS_INCLUDE.MAR

OMNI_INTEGRATOR4_DEFS_INCLUDE.PLI

OMNI_INTEGRATOR4_DEFS_INCLUDE.R32

DAS3964_STARTUP.COM

Getting Help and Reporting Problems

If an error occurs while you are using the DAS for Siemens 3964, and you believe the error is the result of a problem associated with the product, take one of the following actions:

- If you have a basic or DECsupport Software Agreement, call your Customer Support Center (CSC). The CSC provides telephone support for high-level advisory and remedial assistance.
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).
- If you purchased the product within the last 90 days and you think the problem is casued by a software error, you can submit an SPR.

If you submit an SPR, please take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include the description and version number of the product. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, include files, relevant data files and so forth.
4. Provide a list of the program.
5. If the program is longer than 50 lines, submit a copy of it on machine-readable media (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application.
6. Report only one program per SPR. This will facilitate a faster response.
7. Mail the SPR package to DIGITAL.

   Experience shows that many SPRs do not contain enough information to duplicate or identify the problem. Concise, complete information helps DIGITAL give timely and accurate service to software problems.

   If you find an error in the documentation, send an electronic mail message to *manufacturing@digital.com*.

PART I

Using BASEstar Open to Communicate With Siemens 3964 Devices

This part describes how to access Siemens 3964 devices via BASEstar Open services, and how to configure BASEstar Open for use with the DAS for Siemens 3964. It also provides code examples that you can run in the BASEstar Open environment.

The Command Language Interface (CLI) is used to create and operate the required BASEstar Open objects. However, you could use the Application Programming Interface (API), or the Graphical Configuration Utility (GCU) to perform the same tasks.

3
Overview

This chapter describes the BASEstar Open environment components involved when you access Siemens 3964 devices via BASEstar Open services. It also provides an overview of the operations that you can perform using the features provided by the DAS for Siemens 3964:

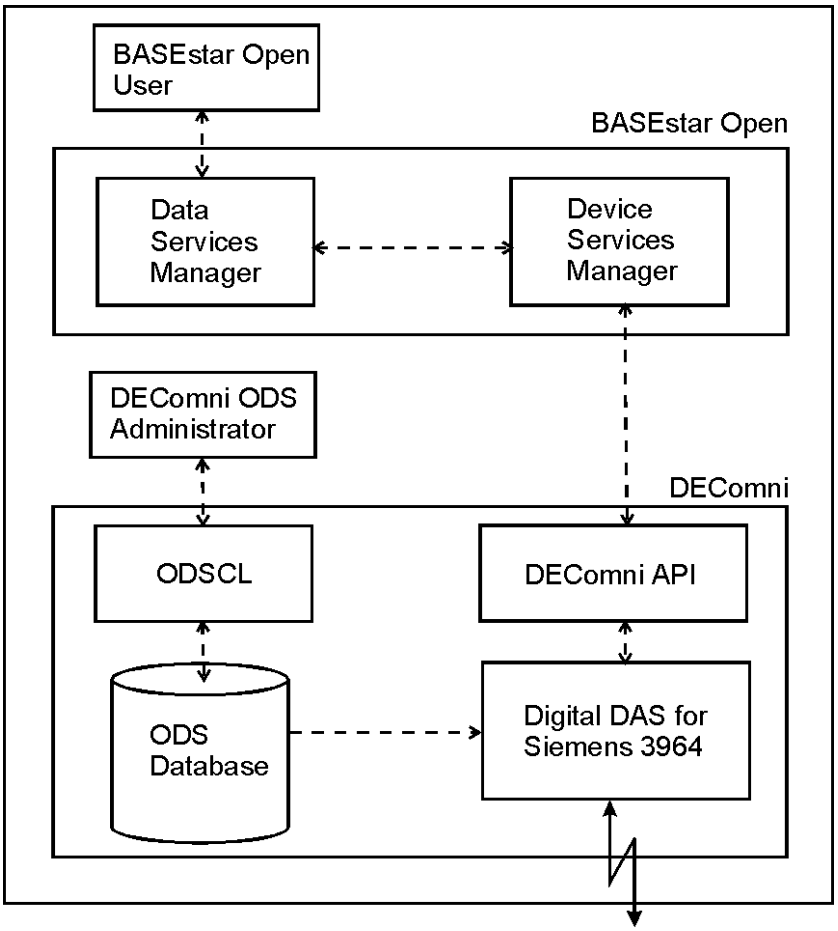
- Exchanging data messages with a Siemens 3964 device using the 3964[R] protocol.
- Reading and writing the memory of a Siemens 3964 device using the 3964[R]_RK512 protocol.

The chapter also discusses how a system equipped with BASEstar Open can emulate Siemens 3964 devices that accept and fulfill requests received from remote devices via the Siemens 3964[R]_RK512 protocol.

BASEstar Open Environment Components

Figure 3-1 shows the environment components involved when you exchange data with Siemens 3964 devices.

Figure 3-1: BASEstar Open Environment Components



Data Services

BASEstar Open users use Device_Data_Points to read and write memory locations of a Siemens 3964 device.

When a user writes the value of a Device_Data_Point, BASEstar Open Device Services sends a message to the appropriate Siemens 3964 device. Vice versa, BASEstar Open updates the value of a Device_Data_Point each time it receives an unsolicited message from a Siemens 3964 device.

Each Device_Data_Point is linked to a Device Services variable which, in turn, is associated with the VMD definition that models the target device.

Device Services

BASEstar Open Device Services provides a unique set of operations to BASEstar Open users for device-independent access to any type of device or network.

This device-independent approach to device connectivity is possible because Device Services models a device as a MMS Virtual Manufacturing Device (VMD).

Each Device Services VMD corresponds to a Siemens 3964 device. A variable associated with this VMD corresponds to a memory location on the same device.

DEComni API

The DEComni API is the engine that provides BASEstar Open Device Services with MMS functions. DEComni API also provides a standardized means of adding new DAS modules to the BASEstar Open environment.

The Omni Directory Services (ODS) database is the DEComni API component that allows you to define the information required for addressing Siemens 3964 devices connected via an RS232 port or terminal server.

DIGITAL DAS for Siemens 3964

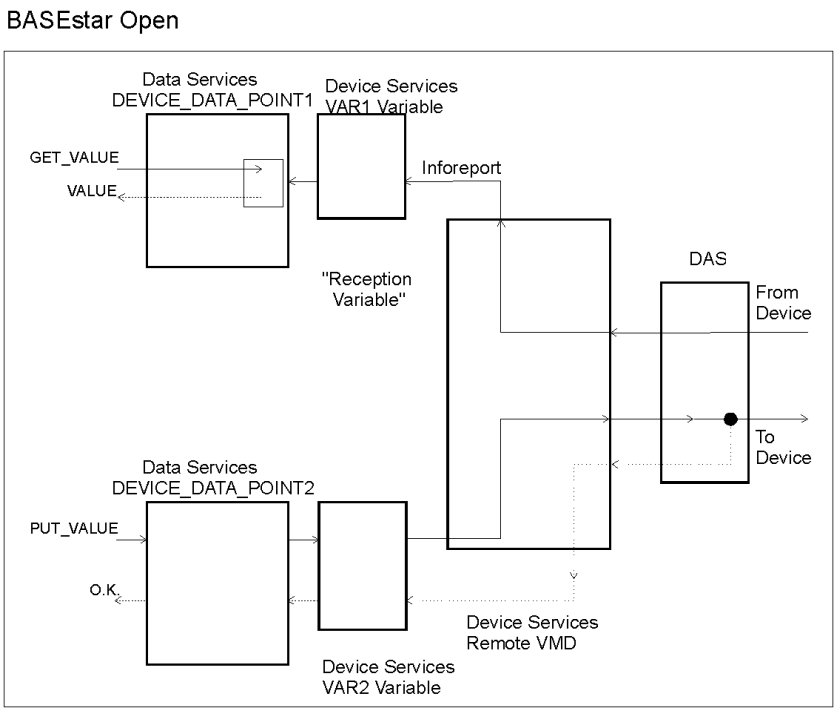
The DIGITAL DAS for Siemens 3964 is a BASEstar Open device connectivity module that implements the Siemens 3964[R] and 3964[R]_RK512 protocols.

Exchanging Data Messages With a Siemens 3964 Device (3964[R] Protocol)

The DAS for Siemens 3964 allows a BASEstar Open user to communicate with a Siemens 3964 device according to the 3964[R] protocol.

Figure 3-2 shows the BASEstar Open objects and the commands a BASEstar Open user must use to receive data messages from (and send data messages to) a remote Siemens 3964 device.

Figure 3-2: BASEstar Open Objects and Operations (3964[R] Protocol)



Sending Data

A BASEstar Open user must issue a PUT_VALUE command on a Device_Data_Point to send a data message to a remote Siemens 3964 device. The Device_Data_Point must be linked to a Device Services variable which is, in turn, associated with a VMD defined for that device.

Several variables can be associated with the same VMD, and a user can use each of the Device_Data_Points linked to these variables to send data messages of different lengths. BASEstar Open sends the remote device a number of characters from the user buffer equal to the size in bytes of the datatype that has been associated with the variable used for the send operation.

The PUT_VALUE command returns control to the application as soon as BASEstar Open has sent the data through the local port successfully.

Receiving Data

A BASEstar Open user must issue a GET_VALUE command on a Device_Data_Point to obtain from BASEstar Open the value of a data message coming from a remote Siemens 3964 device. The Device_Data_Point must have been configured for receive operations, and must be linked to the "reception variable" associated with the VMD that models the device. BASEstar Open always uses the reception variable to return the data received from the Siemens 3964 device.

The "reception variable" is the *first* Device Services Named_Variable that has been created for the VMD. If no Named_Variable has been defined, the first Unnamed_Variable is used. If no variable is associated with the remote VMD, any data received from the device are lost.

The GET VALUE command returns to the user the cached value of the Device_Data_Point itself; that is, there is no interaction with the DAS nor with the device.

BASEstar Open updates the value of the Device_Data_Point regardless of the receive operations performed by the user. When receiving a message from the device, BASEstar Open generates an internal Inforeport that causes the unsolicited updating of the cached value of the Device_Data_Point that is linked to the Device Services "reception variable".

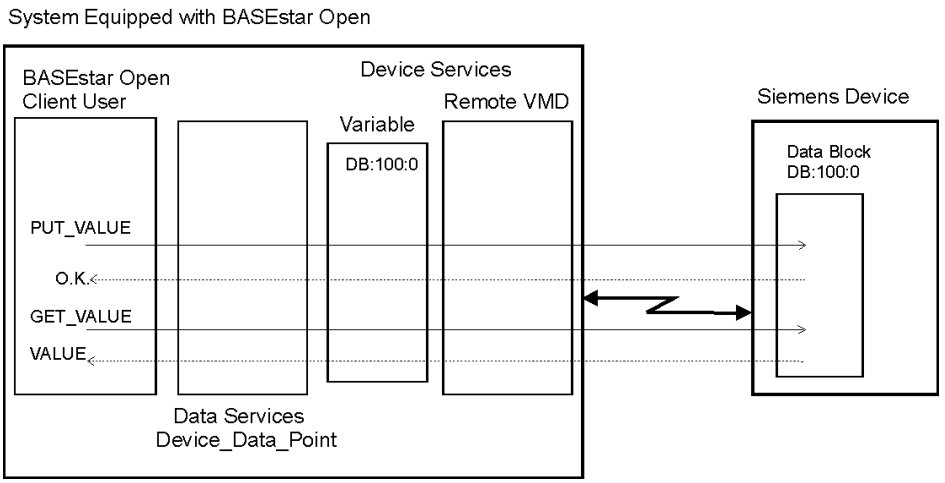
BASEstar Open interprets the data received from the device in one data message according to the datatype associated with the "reception variable". Note that if an insufficient number of bytes is received, the value returned in the user buffer is unpredictable; if, instead, the number of the received bytes exceeds that required to meet the datatype, the exceeding bytes are lost.

Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol)

A BASEstar Open user gets (or puts) the value of a Device_Data_Point to read data from (or write data to) a physical memory location of a Siemens 3964 device. As shown in Figure 3-3, this Device_Data_Point must be linked to an Unnamed_Variable associated with a VMD that models the device.

A GET_VALUE command reads data from the device memory, and returns in the user buffer a number of characters that is equal to the size of the BASEstar Open DataType associated with the variable used for read operations. A similar consideration applies to a PUT_VALUE command.

Figure 3-3: Using BASEstar Open to Access the Memory of a Siemens 3964 Device



To obtain a BASEstar Open configuration that is appropriate for your needs, you must consider the requirements of your overall application. Figure 3-3 illustrates an example in which BASEstar Open operations are synchronous with user's actions; in fact:

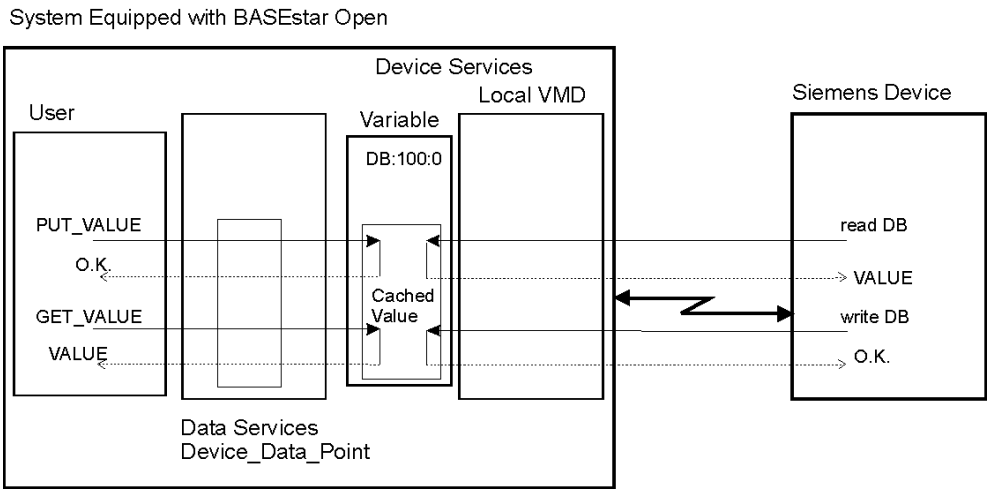
- Each time the user issues a PUT_VALUE command, BASEstar Open immediately updates the device memory.
- Each time the user issues a GET_VALUE command, BASEstar Open immediately reads the device memory.

However, you can configure BASEstar Open for asynchronous operations by means of a Polling_Set. In this case, BASEstar Open automatically reads the value of the device memory, and updates the value of the Device_Data_Point asynchronously with respect to the user's actions.

Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol)

Figure 3-4 shows how BASEstar Open emulates a Siemens 3964 device via a local VMD. When emulating a Siemens 3964 device, BASEstar Open acts as a server; that is, it accepts and fulfills write and read requests received from a remote Siemens 3964 device or a BASEstar Open user.

Figure 3-4: Using BASEstar Open to Emulate a Siemens 3964 Device



To direct BASEstar Open to act as server you must create a local VMD object for each Siemens 3964 device you want to emulate, and you must associate a Device Services Unnamed_Variable to the VMD for each memory block to be emulated.

BASEstar Open users can exchange data messages with emulated Siemens 3964 devices as described in the section of this chapter entitled *Exchanging Data Messages With a Siemens 3964 Device (3964[R] Protocol)*.

Datatypes and Memory Block Sizes

A GET_VALUE command returns in the user buffer a number of characters that is equal to the size of the BASEstar Open datatype associated with the variable used in the read operation. The size (in bytes) of this data type must therefore be equal to the size (in bytes) of the data that are contained in a data block (represented by the variable).

A similar consideration applies to the PUT_VALUE command.

In order to simplify the calculation of the sizes of BASEstar Open datatypes, Appendix B provides the sizes of the basic BASEstar Open datatypes.

4
Configuring BASEstar Open

This chapter describes how you must configure BASEstar Open in order to allow users to access a Siemens 3964 device.

To configure BASEstar Open you must:

- Register the appropriate entries in the ODS database.
- Create the required Application Services Activity and Program objects.
- Create the required Device Services Protocol_Profile, VMD and Variable objects.
- Create the required Device_Data_Point objects.

Note

For an explanation of the BASEstar Open object attributes whose values are not specified in this chapter, refer to the *BASEstar Open Command Language Interface* and the *BASEstar Open Reference Guide*.

Registering ODS Entries

For each Siemens 3964 device you must create an ODS entry which specifies Siemens 3964 protocol-specific information. See Appendix A for details.

Creating Application Services Objects

You can use the DATADEV server or separate Device and Data servers. The DATADEV server makes both Data and Device Services servers available in a single process, thus simplifying configuration operations, optimizing performance, and allowing the same Device_Data_Point to be used for both send and receive operations.

Note

The DATADEV server is used throughout this manual, therefore you should modify the information and examples provided in this chapter if you want to use two separate Device and Data Services servers (for example, if your BASEstar Open version does not support DATADEV servers).

See also Chapter 5 for examples showing the commands used to create the required Program and Activity objects. For further details, refer to the *BASEstar Open Command Language Interface* and to the *BASEstar Open Reference Guide*.

Creating Device Services Objects

You must create the following Device Services objects in the order listed below:

1. Protocol_Profile
2. VMD
3. Variables

Creating Protocol_Profiles

Device Services uses Protocol_Profiles to associate a VMD with the appropriate DAS. You must create one Protocol_Profile for each Siemens 3964 protocol, and you can use it for all the VMDs that are associated with that protocol. Table 4-1 lists the Protocol_Profile attributes whose values are specific to the Siemens 3964 protocols.

Table 4-1: Protocol_Profile Attributes

Attribute Name	Description and Values
-APPLPROFID	The identifier of the DAS to be used. Possible values are:
	"31" for the Siemens 3964 protocol
	"32" for the Siemens 3964R protocol
	"33" for the Siemens 3964_RK512 protocol
	"34" for the Siemens 3964R_RK512 protocol
-MAXPDUSIZE	The maximum length of the data exchanged with the Siemens 3964 device.
	For the 3964[R] protocol, there is no limit in the size of the message exchanged. You can specify any value, provided it is supported by DEComni API.
	For the 3964[R] _RK512 protocol, there is no limit in the size of the message exchanged between DEComni API applications and Siemens 3964 devices, but the message is segmented for transmission over the communication line. This attribute must specify the segment size, which must be less than, or equal to 138 (including 10 bytes for the header).

Example 4–1shows how to create the Protocol_Profiles for all the supported protocols.

Example 4–1: Creating Device Services Protocol_Profiles

```
BSTR> CREATE PROTOCOL_PROFILE PP_3964 –APPLPROFID 31\
      –MAXPDUSIZE 1024 –LOG

BSTR> CREATE PROTOCOL_PROFILE PP_3964R –APPLPROFID 32\
      –MAXPDUSIZE 1024 –LOG

BSTR> CREATE PROTOCOL_PROFILE PP_3964_RK512 –APPLPROFID 33\
      –MAXPDUSIZE 138 –LOG

BSTR> CREATE PROTOCOL_PROFILE PP_3964R_RK512 –APPLPROFID 34\
      –MAXPDUSIZE 138 –LOG
```

Creating VMDs

A Device Services VMD models a Siemens 3964 device. Table 4-2 lists the mandatory attributes that you must specify for a correct definition of a Siemens 3964 VMD for any of the supported protocols.

Table 4-2: VMD Attributes

Attribute Name	Description and Values
-DVM_ACCESS_POINT	When creating a remote VMD (that is, a VMD that models a real Siemens 3964 device), a list of one or more access point descriptors, for each access point users use to access the remote device. An access point descriptor has the following format: `application_simple_name:protocol_profile_name` where: *application_simple_name* must specify an ODS entry valid for the Siemens 3964 protocol in use (see Appendix A). *protocol_profile_name* must specify a Protocol_Profile created according to the explanations provided in this chapter.
-CLIENT_ACCESS_POINT	When creating a local VMD (that is, a VMD that models an emulated Siemens 3964 device), a list of one or more descriptors, one for each access point users use to access the remote device. An access point descriptor has the following format: `application_simple_name:protocol_profile_name` where: *application_simple_name* must specify an ODS entry valid for the used Siemens 3964 protocol (see Appendix A). *protocol_profile_name* must specify a Protocol_Profile created according to the explanations provided in this chapter.
-MODEL	Used for documentation purposes.
-VENDOR	Used for documentation purposes.

Creating a VMD for Communication Via the 3964[R] Protocol

To exchange data messages with a Siemens 3964 device using the 3964[R] protocol you must define a remote VMD. Example 4–2 shows how to create a valid remote VMD.

Example 4–2: Creating a Device Services VMD (3964[R] Protocol)

```
BSTR> CREATE VMD VMD_REMOTE_3964R -MODEL "SIEMENS 3964R"\
      -VENDOR "BASEstar Open"\
      -DVM_ACCESS_POINTS ("/cn=SIEMENS_3964":PP_3964R)\
      -DESCRIPTION "For sending data to a Siemens 3964 device using\
       the Siemens 3964R protocol."\
      -LOG
```

Creating a VMD for Communication Via the 3964[R]_RK512 Protocol

Depending on the operation you want to perform, you must define a remote VMD, a local VMD, or both:

- A remote VMD models a real Siemens 3964 device and associated variables. For a remote VMD, a BASEstar Open user can act as a client that writes and reads the value of memory blocks physically located on the associated Siemens 3964 device.
- A local VMD models a Siemens 3964 device emulated on the host system. For a local VMD, BASEstar Open acts as a server that fulfills write and read requests coming from both the Siemens 3964 device and BASEstar Open users.

Note

A DATADEV or DEVICE server currently supports one local VMD. When needed, its name must be specified by the BSTR_LOCAL_VMD_NAME BASEstar Open global variable.

Example 4–3 shows how to create the VMDs for communicating with a Siemens 3964 device using the Siemens 3964R_RK512 protocol.

Example 4–3: Creating Device Services VMDs (3964[R]_RK512 Protocol)

```
BSTR> CREATE VMD VMD_REMOTE_3964R_RK512 -MODEL "SIEMENS 3964R_RK512"\
      -VENDOR "BASEstar Open"\
      -DVM_ACCESS_POINTS ("/cn=REMOTE_3964_512":PP_3964R_RK512)\
      -DESCRIPTION "Allows BASEstar Open users to write and read
       memory blocks of a Siemens 3964 device using the Siemens
       3964[R]_RK512 protocol.
      -LOG

BSTR> CREATE VMD VMD_LOCAL_3964R_RK512 -MODEL "SIEMENS 3964R_RK512"\
      -VENDOR "BASEstar Open"\
      -DVM_ACCESS_POINTS ("/cn=LOCAL_3964_512":PP_3964R_RK512)\
      -DESCRIPTION "Emulates a Siemens 3964 device using the Siemens\
       3964[R]_RK512 protocol.\
      -LOG
```

Creating Variables

Device Services variables model variables at a remote or a local VMD.

Creating Variables for Communication Via the 3964[R] Protocol

Because the data type of the received data message often differs from the data type of the sent data message, a separate description is provided for the configuration characteristics of a "reception variable" and for those of a variable used to send data messages.

To receive data from Siemens 3964 devices you can use Named or Unnamed_Variables. The first Named_Variable you define at a given remote VMD is assumed to be the "reception variable", and will be used for read operations only. If no Named_Variables are defined, the first Unnamed_Variable is used.

Table 4-3 lists the attributes that you must specify for a correct definition of a reception variable.

Table 4-3: Reception Variable Attributes (3964[R] Protocol)

Attribute Name	Description and Values
-DATATYPE *datatype_name*	Name of the datatype BASEstar Open associates with the data received from the device. Appendix B specifies the sizes of the basic BASEstar Open datatypes.
-SEND *notification*	This attribute must be set to ALWAYS or to ON_CHANGE. Such settings direct BASEstar Open to update in unsolicited mode the value of the linked Device_Data_Point on reception of data from the device.
-ACCESS *access_type*	Must be set to READ.
-ADDRTYPE *address_type*	Significant for Unnamed_Variables only; it must be set to UNCONSTRAINED.
-ADDRESS *address*	Significant for Unnamed_Variables only; it can be set to any value.

You can define as many Named or Unnamed_Variables associated with a local VMD as you need to send data to a Siemens 3964 device. Each of these variables can have a different datatype, which allows you to send different types of data to the Siemens 3964 device.

Table 4-4 lists the attributes that you must specify for a correct definition of a transmission variable.

Table 4-4: Transmission Variable Attributes (3964[R] Protocol)

Attribute Name	Description and Values
-DATATYPE *datatype_name*	Name of the datatype used by BASEstar Open to know the number of bytes in the user buffer. Appendix B specifies the sizes of the basic BASEstar Open datatypes.
.-ACCESS *access_type*	Must be set to WRITE.
-ADDRTYPE *address_type*	Significant for Unnamed_Variables only; it must be set to UNCONSTRAINED.
-ADDRESS *address*	Significant for Unnamed_Variables only; it can be set to any value.

Creating Variables for Communication Via the 3964[R]_RK512 Protocol

You must use Unnamed_Variables to exchange data with a Siemens 3964 device. You can use the same variable for both read and write operations.Table 4-5 lists the attributes that you must specify for a correct definition of an Unnamed_Variable that can be associated with a VMD that models a real or an emulated Siemens 3964 device.

Table 4-5: Variable Attributes (3964[R]_RK512 Protocol)

Attribute Name	Description and Values
-DATATYPE *datatype_name*	Name of a BASEstar datatype. Its size (in bytes) must match the size (in bytes) of the associated Siemens 3964 device memory block. See Appendix B for the sizes of the basic BASEstar Open datatypes.
-SEND *notification*	Only significant for read operations. You can set any value provided it is consistent with the overall configuration.
-ACCESSMODE *access_mode*	You are suggested to set this attribute to WRITE, to have the same variable supporting both read and write operations.
-ADDRTYPE *address_type*	It must be set to UNCONSTRAINED.
-ADDRESS *address*	This attribute must specify a valid Siemens 3964 memory block address. See Appendix D.

Creating Data Services Objects

BASEstar Open users exchange data messages with devices by means of Device_Data_Points without taking into consideration communication and device characteristics.

Creating Device_Data_Points for Communication Via the 3964[R] Protocol

Table 4-6 lists the attributes you must set to define Device_Data_Points that will be used to receive data from a Siemens device using the 3964[R] protocol.

Table 4-6: Device_Data_Point Attributes for Receive Operations (3964[R] Protocol)

Attribute Name	Description and Values
-VMD_NAME *vmd_name*	Must be set with the name of a remote VMD.
-VAR_NAME *var_name*	Must be set with the name of the "reception variable" that is associated with the *vmd_name* VMD.
-VAR_CLASS *var_class*	Must be set to NAMED or UNNAMED according to the class of the *var_name* Device Services variable.
-DATATYPE *datatype_name*	Must be set with the name of the same datatype associated with the *var_name* variable specified by the -VAR_NAME attribute.
-DEVICE_ACCESS *device_access*	Must be set to READ to indicate that the linked Device Services variable produces the value of the Device_Data_Point.
-UPDATE_POLICY *update_policy*	Must be set to UNSOLICITED to indicate that Data Services updates the value of the Device_Data_Point according to what was specified by the -SEND attribute of the linked Device Services variable.

Table 4-7 lists the attributes you must set to define Device_Data_Points that will be used to send data to a Siemens device using the 3964[R] protocol.

Table 4-7: Device_Data_Point Attributes for Send Operations (3964[R] Protocol)

Attribute Name	Description and Values
-VMD_NAME *vmd_name*	Must be set with the name of a local VMD.
-VAR_NAME *var_name*	Must be set with the name of a "transmission variable" associated with the *vmd_name* VMD.
-VAR_CLASS *var_class*	Must be set to NAMED or UNNAMED according to the class of the *var_name* Device Services variable.
-DATATYPE *datatype_name*	Must be set with the name of the same datatype associated with the *var_name* variable specified by the -VAR_NAME attribute.
-DEVICE_ACCESS *device_access*	Must be set to WRITE to indicate that Data Services will use the value provided by the PUT_VALUE command to update the value of the linked Device Services variable (and therefore to cause the DAS for Siemens 3964 to send the value to the Siemens device).

Creating Device_Data_Points for Communication Via the 3964[R]_512 Protocol

A BASEstar Open user uses a Device_Data_Point associated with a remote (or local) VMD to write data directly to the memory of a real (or emulated) Siemens 3964 device.

Table 4-8 lists the attributes you must set to create a Device_Data_Point that will be used to read and write device memory blocks.

Table 4-8: Device_Data_Point for Send and Receive Operations (3964[R]_512 Protocol)

Attribute Name	Description and Values
-VMD_NAME *vmd_name*	Must be set with the name of a remote VMD (that is, a VMD that models a remote Siemens 3964 device), or a local VMD (that is, a VMD that emulates a Siemens 3964 device on the host system).
-VAR_NAME *var_name*	Must be set with the name of a Device Services Unnamed_Variable associated with the *vmd_name* VMD.
-VAR_CLASS *var_class*	Must be set to NAMED or UNNAMED according to the class of the *var_name* Device Services variable.
-DATATYPE *datatype_name*	Must be set with the name of the same datatype associated with the *var_name* variable specified by the -VAR_NAME attribute.
-DEVICE_ACCESS *device_access*	You are suggested to set it to RDWR to indicate that this Device_Data_Point can be used for both read and write operations.
-UPDATE_POLICY *update_policy*	Only significant for read operations. You can set any value provided it is consistent with the overall configuration.
-ACCESS_POLICY *access_policy*	Only significant for read operations. You can set any value provided it is consistent with the overall configuration.

5
Examples

This chapter contains three CLI examples:

- The first example shows how BASEstar Open users can exchange data with a Siemens 3964 device using the 3964[R] protocol.
- The second example shows how BASEstar Open users can read data from, and write data to memory blocks of a Siemens 3964 device using the 3964[R]_RK512 protocol.
- The third example shows how your system can emulate a Siemens 3964 device via the 3964[R]_RK512 protocol.

Example 1: Using the 3964[R] Protocol to Exchange Data

The example contained in this section describes how you can configure BASEstar Open to allow users to exchange data with a Siemens 3964 device using the 3964[R] protocol.

Example Overview

BASEstar Open being highly configurable, this example shows only one of many possible modes for implementing send and receive operations.

Send data operations

It is assumed that send data operations are executed in synchronous mode. According to configuration, each time a user issues a PUT_VALUE command BASEstar Open immediately sends the data. The command only terminates and returns control to the user when the operation has been completed.

Receive data operations

In this example, a user must keep reading the reception variable if he wants to know when its value changes due to reception of an unsolicited data message from the device. However, you could change the BASEstar Open configuration in order to receive an Event each time BASEstar Open receives an unsolicited data message from the device; this would allow BASEstar Open users to be notified in an immediate and asynchronous mode of any change occurring on the device.

Before Running the Example

It is assumed that the Siemens 3964 device has been configured, is functioning and can receive incoming data. Before you can use this example, you must:

- Replace *my_tcpnode* with the TCP/IP name of your the BASEstar Open Node.
- Replace *my_DAS_id* with the identifier of the Siemens 3964 protocol you wish to use.
- Replace O*DS_Remote3964* with the name of an ODS entry that complies with the communication characteristics of the Siemens 3964 device.

CLI Source

Example 5–1 shows the CLI commands used to create BASEstar Open objects and to activate the required servers.

Example 5–1: Creating BASEstar Open Objects (3964[R] Protocol)

```
! ----------------------------------------------------------------
! The example is set up to work with the VOLATILE database,
! but you can easily adapt it to work with the PERMANENT database.
! ----------------------------------------------------------------
SET SCOPE VOLATILE
! ----------------------------------------------------------------
! Create the Common Services objects: Nodes, Actors, Domains,
| Protocol_Profiles and VMDs
! ----------------------------------------------------------------
CREATE NODE NODE1 -PHYSICAL_NAME "my_tcpnode"
CREATE ARRAY_DATATYPE ARRAY_OF_INTEGERS_16 -NUM_OF_ELEMENTS 10\
       -DATATYPE INTEGER_32
```

```
CREATE DOMAIN /DOM_DATA

CREATE ACTOR /ACT_SERVERS

CREATE PROTOCOL_PROFILE PP_3964 -APPLPROFID My_DAS_id\
       -MAXPDUSIZE 1024 -LOG

CREATE VMD VMD_REM3964\
       -MODEL "3964R" -VENDOR "Digital"\
       -DVM_ACCESS_POINTS ODS_Remote3964:PP_3964)\
       -DESCRIPTION "VMD for data exchange"\
       -LOG

! ---------------------------------------------------------------------

! Create the objects needed to define the Data Services and

! Device Services servers.

! ---------------------------------------------------------------------

CREATE PROGRAM /ACT_SERVERS/PRG_DATA -PROGRAM_KIND DATADEV

CREATE ACTIVITY /ACT_SERVERS/ACY_DATA\
       -PROGRAM /ACT_SERVERS/PRG_DATA -NODES (NODE1)\
       -DOMAINS (/DOM_DATA)\
       -VMDS (/VMD_REM3964)\
       -STARTUP_TIMEOUT 120 -RECOVERY_POLICY NECESSARY

! ---------------------------------------------------------------------

! Execute the ACTOR to activate the required servers.

!(Connection with the device is set up by the Device Services
! server).

! ---------------------------------------------------------------------

EXECUTE ACTOR /ACT_SERVERS

! ---------------------------------------------------------------------

! Create the Device Services objects.

! You can change the Datatypes associated with the

| variables to meet your application dialogue requirements.

! ---------------------------------------------------------------------

CREATE UNNAMED_VAR VMD_REM3964.VAR_RCV -DATATYPE INTEGER_32\
       -SEND ALWAYS  -ACCESSMODE READ\
       -ADDRESS XXX -ADDRESSTYPE UNCONSTRAINED

CREATE UNNAMED_VAR VMD_REM3964.VAR_SEND001 -DATATYPE INTEGER_32\
       -ACCESSMODE WRITE

CREATE UNNAMED_VAR VMD_REM3964.VAR_SEND002\
       -DATATYPE ARRAY_OF_INTEGERS_32 -ACCESSMODE WRITE

! ---------------------------------------------------------------------

! Create the Data Services objects.

! ---------------------------------------------------------------------

CREATE DEVICE_DATA_POINT DP_RECEIVE\
       -VMD_NAME VMD_REM3964\
       -VAR_NAME VAR_RCV -VAR_CLASS UNNAMED\
       -DATATYPE INTEGER_32 -DEVICE_ACCESS READ\
       -UPDATE_POLICY UNSOLICITED

CREATE DEVICE_DATA_POINT DP_SEND001\
       -VMD_NAME VMD_REM3964\
```

```
       -VAR_NAME VAR_SEND001 -VAR_CLASS UNNAMED\
       -DATATYPE INTEGER_32 -DEVICE_ACCESS WRITE

CREATE DEVICE_DATA_POINT DP_SEND002\
       -VMD_NAME VMD_REM3964\
        -VAR_NAME VAR_SEND002 -VAR_CLASS UNNAMED\
       -DATATYPE ARRAY_OF_INTEGERS_32 -DEVICE_ACCESS WRITE
```

You can now receive data from, and send data to a Siemens 3964 device using the 3964[R] protocol. A possible operation sequence that uses the objects previously created is described in Example 5–2.

Example 5–2: Exchanging Data with a Siemens 3964 Device (3964[R] Protocol)

```
BSTR> PUT VALUE DEVICE_DATA_POINT DP_SEND001

(Integer 32) : 345

BSTR> GET VALUE DEVICE_DATA_POINT DP_RECEIVE

12

       .........

       "Keep receiving until you receive the answer!"/

       .........

BSTR> PUT VALUE DEVICE_DATA_POINT DP_SEND002

array[10] of INTEGER_16

   [0]: [Integer 32] : 3

   [1]: [Integer 32] : 12

   [2]: [Integer 32] : 4999

   [3]: [Integer 32] : 3

   [4]: [Integer 32] : 3

   [5]: [Integer 32] : 12

   [6]: [Integer 32] : 3333

   [7]: [Integer 32] : 33333

   [8]: [Integer 32] : 2

   [9]: [Integer 32] : 2

BSTR> GET VALUE DEVICE_DATA_POINT DP_RECEIVE

1234
```

Example 2: Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol)

The example contained in this section describes how you must configure BASEstar Open to allow users to read and write memory blocks of a remote Siemens 3964 device using the 3964[R]_RK512 protocol.

Example Overview

BASEstar Open being highly configurable, this example shows only one of many possible modes for implementing read and write operations.

It is assumed that read and write operations are executed in synchronous mode. According to configuration, BASEstar Open physically accesses the memory of the addressed device each time a user issues a PUT_VALUE or a GET_VALUE command. The command only terminates and returns control to the user when the operation has been completed.

Before Running the Example

It is assumed that the Siemens 3964 device has been configured, is functioning and can receive incoming data. Before you can use this example, you must:

- Replace *my_tcpnode* with the TCP/IP name of your the BASEstar Open Node.
- Replace *my_DAS_id* with the identifier of the Siemens 3964[R]_RK512 protocol you wish to use.
- Replace *ODS_remote3964_RK512* with the name of an ODS entry that complies with the 3964[R]_512 protocol communication characteristics.

CLI Source

Example 5–3 shows the CLI commands that are used to create BASEstar Open objects and to activate the required servers.

Example 5–3: Creating BASEstar Open Objects (3964[R]_RK512 Protocol)

```
! ----------------------------------------------------------------------
! The example is set up to work with the VOLATILE database,
! but you can easily adapt it to work with the PERMANENT database.
! ----------------------------------------------------------------------
SET SCOPE VOLATILE
! ----------------------------------------------------------------------
! Create the Common Services objects: Nodes, Actors, Domains,
| Protocol_Profiles and VMDs.
! ----------------------------------------------------------------------
CREATE NODE NODE1 -PHYSICAL_NAME "my_tcpnode"
CREATE DOMAIN /DOM_DATA
CREATE ACTOR /ACT_SERVERS
CREATE PROTOCOL_PROFILE PP_3964_RK -APPLPROFID My_DAS_id\
       -MAXPDUSIZE 138 -LOG
CREATE VMD VMD_REM3964_RK\
       -MODEL "S5-135U" -VENDOR "Siemens"\
       -DVM_ACCESS_POINTS ODS_remote3964_RK512:PP_3964_RK)\
```

```
       -DESCRIPTION "VMD for device access"\
       -LOG

! ------------------------------------------------------------------

! To simplify configuration, a single DATADEV server is started up.

! ------------------------------------------------------------------

CREATE PROGRAM /ACT_SERVERS/PRG_DATADEV -PROGRAM_KIND DATADEV

CREATE ACTIVITY /ACT_SERVERS\ACY_DEVICE\
       -PROGRAM /ACT_SERVERS/PRG_DEVICE -NODES (NODE1)\
       -DOMAINS (/DOM_DATA)\
       -VMDS (/VMD_REM3964_RK)\
       -STARTUP_TIMEOUT 120 -RECOVERY_POLICY NECESSARY

! ------------------------------------------------------------------

! Execute the ACTOR to activate the needed servers

! (The connection with the device is started up.)

! ------------------------------------------------------------------

EXECUTE ACTOR /ACT_SERVERS

! ------------------------------------------------------------------

! Create the Device Services variable to access the DB:100:10 device
! memory block.

! You can change it to access another device memory block.

! ------------------------------------------------------------------

CREATE UNNAMED_VAR VMD_REM3964_RK.REMVAR_DB10010\
       -DATATYPE INTEGER_32 -SEND NEVER -ACCESSMODE write\
       -ADDRTYPE UNCONSTRAINED -ADDRESS "DB:100:0"

! ------------------------------------------------------------------

! Create the Device_Data Point that allows BASEstar Open users to

! directly access the DB:100:10 device memory block.

! ------------------------------------------------------------------

   CREATE DEVICE_DATA_POINT DP_REMDB10010\
      -VMD_NAME VMD_REM3964_RK\
      -VAR_NAME REMVAR_DB10010 -VAR_CLASS UNNAMED\
      -DATATYPE INTEGER_32 -DEVICE_ACCESS RDWR\
      -UPDATE_POLICY SOLICITED -ACCESS_POLICY ALWAYS
```

Example 5–4 shows a possible sequence that allows a user to read and write a memory block of the remote Siemens 3964 device. On successful completion of the following command sequence, you have first read the current value of the DB:100:10 directly from the memory of the device, then you have written a new value, and finally you have checked the written value by issuing another read command.

Example 5–4: Reading and Writing the Device Memory (3964[R]_RK512 Protocol)

```
BSTR> GET VALUE DEVICE_DATA_POINT DP_REMDB10010
12
BSTR>
BSTR> PUT VALUE DEVICE_DATA_POINT DP_REMDB10010
(Integer 32) : 345
BSTR>
BSTR> GET VALUE DEVICE_DATA_POINT DP_REMDB10010
345
```

Example 3: Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol)

The example contained in this section describes how to configure BASEstar Open in order to allow your system to emulate a Siemens 3964 device. Other Siemens 3964 devices can access the emulated device using the 3964[R]_RK512 protocol.

Example Overview

Using the BASEstar Open configuration shown in this example, a remote Siemens 3964 device can access the memory of a locally emulated device. BASEstar Open automatically fulfills a request.

In addition, a BASEstar Open user can read and write the value of a variable on the emulated device.

Before Running the Example

It is assumed that the Siemens 3964 device has been configured, is functioning and can receive incoming data. Before you can use this example, you must:

- Replace *my_tcpnode* with the TCP/IP name of your the BASEstar Open Node.
- Replace *my_DAS_id* with the identifier of the Siemens 3964[R]_RK512 protocol you wish to use.
- Replace *ODS_local3964_RK512* with the name of an ODS entry that complies with the 3964[R]_512 protocol communication characteristics.
- Replace *parameter_file* with the pathname of the parameter file. As shown in Example 5–5, the BSTR_LOCAL_VMD global variable must specify the name of the local VMD, and BSTR_CONN_MODE must be set to PASSIVE (this directs BASEstar Open to make the locally emulated device available to the remote device).

Example 5–5: Parameter File (3964[R]_RK512 Protocol)

```
/ACT_SERVERS/PRG_DATADEV.env: BSTR_LOCAL_VMD = VMD_LOC3964_RK
/ACT_SERVERS/PRG_DATADEV.env: BSTR_CONN_MODE = PASSIVE
```

CLI Source

Example 5–6 shows the CLI commands that are used to create BASEstar Open objects and to activate the required servers.

Example 5–6: Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol)

```
! ----------------------------------------------------------------
! The example is set up to work with the VOLATILE database,
! but you can easily adapt it to work with the PERMANENT database.
! ----------------------------------------------------------------
SET SCOPE VOLATILE
! ----------------------------------------------------------------
! Create the Common Services objects: Nodes, Actors, Domains,
| Protocol_Profiles and VMDs.
! ----------------------------------------------------------------
CREATE NODE NODE1 -PHYSICAL_NAME "my_tcpnode"
```

Examples

```
CREATE DOMAIN /DOM_DATA

CREATE ACTOR /ACT_SERVERS

CREATE PROTOCOL_PROFILE PP_3964_RK -APPLPROFID My_DAS_id\
       -MAXPDUSIZE 138 -LOG

CREATE VMD VMD_REM3964_RK\
       -MODEL "S5-135U" -VENDOR "Siemesn"\
       -DVM_ACCESS_POINTS ODS_Local3964_RK512:PP_3964_RK)\
       -DESCRIPTION "Dummy remote VMD"\
       -LOG

CREATE VMD VMD_LOC3964_RK\
       -MODEL "3964_RK512" -VENDOR "Digital"\
       -CLIENT_ACCESS_POINTS ODS_Local3964_RK512:PP_3964_RK)\
       -DESCRIPTION "Local VMD for device emulation"\
       -LOG

! ---------------------------------------------------------------------

! To simplify configuration, a single DATADEV server is started up.

! ---------------------------------------------------------------------

CREATE PROGRAM /ACT_SERVERS/PRG_DATADEV -PROGRAM_KIND DATADEV\
       -PARAMETER_FILE parameter_file

CREATE ACTIVITY /ACT_SERVERS\ACY_DEVICE\
       -PROGRAM /ACT_SERVERS/PRG_DEVICE -NODES (NODE1)\
       -DOMAINS (/DOM_DATA)\
       -VMDS (/VMD_REM3964_RK\
       -STARTUP_TIMEOUT 120 -RECOVERY_POLICY NECESSARY

! ---------------------------------------------------------------------

! Execute the ACTOR to activate the needed servers

! (The connection with the device is started up.)

! ---------------------------------------------------------------------

EXECUTE ACTOR /ACT_SERVERS

! ---------------------------------------------------------------------

! Create the Device Services variable associated with the

! DB:100:10 device memory block.

! You can change it to make available other device memory blocks.

! ---------------------------------------------------------------------

CREATE UNNAMED_VAR VMD_REM3964_RK.LOCVAR_DB10010\
       -DATATYPE INTEGER_32 -SEND NEVER -ACCESSMODE WRITE\
       -ADDRTYPE UNCONSTRAINED -ADDRESS "DB:100:0"

! ---------------------------------------------------------------------

! Create the Device_Data Point that allows BASEstar Open users to

! directly access the DB:100:10 memory block on the Siemens device.

! ---------------------------------------------------------------------

CREATE DEVICE_DATA_POINT DP_LOCDB10010\
       -VMD_NAME VMD_LOC3964_RK\
       -VAR_NAME LOCVAR_DB10010 -VAR_CLASS UNNAMED\
       -DATATYPE INTEGER_32\
       -DEVICE_ACCESS RDWR\
       -UPDATE_POLICY UNSOLICITED -ACCESS_POLICY ALWAYS
```

Example 5–7 shows a possible sequence that allows a BASEstar Open user to read and write a memory block of the locally emulated Siemens 3964 device.

Example 5–7: Reading and Writing the Emulated Device Memory (3964[R]_RK512 Protocol)

```
BSTR> GET VALUE DEVICE_DATA_POINT DP_LOCDB10010
12
BSTR>
BSTR> PUT VALUE DEVICE_DATA_POINT DP_LOCDB10010
(Integer 32) : 345
BSTR>
BSTR> GET VALUE DEVICE_DATA_POINT DP_LOCDB10010
345
```

PART II

Using DEComni API to Communicate With Siemens 3964 Devices

This part explains how to prepare DEComni API applications to communicate with Siemens devices using the features of the DAS for Siemens 3964.

6
Overview

This chapter describes the DEComni API environment components involved when you access Siemens 3964 devices via the DEComni API. It also provides an overview of the operations a DEComni API application can perform using the features provided by the DAS for Siemens 3964:

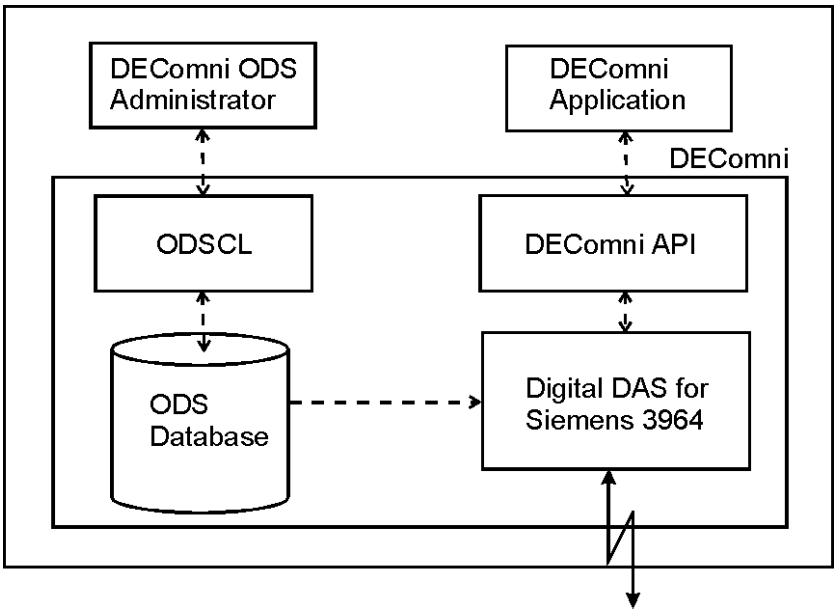
- Exchanging data messages with a Siemens 3964 device (3964[R] protocol).
- Reading and writing the memory of a Siemens 3964 device (3964[R]_RK512 protocol).

The chapter also discusses how a system equipped with DEComni API can emulate Siemens 3964 devices that accept and fulfill requests coming from the remote devices through the Siemens 3964[R]_RK512 protocol.

DEComni API Environment Components

Figure 6-1 shows the components involved when DEComni API applications exchange data with Siemens 3964 devices.

Figure 6-1: DEComni API Environment Components



DEComni API

The DEComni API provides applications with a unique set of operations for device-independent access to any type of device or network. This device-independent approach to device connectivity is possible because each device is modeled as a MMS Virtual Manufacturing Device (VMD). A VMD is associated with a Siemens 3964 device. A variable object corresponds to data stored in the device memory.

The DEComni API also provides a standardized means of adding new DAS modules.

The Omni Directory Services (ODS) database is the DEComni API component that allows you to define the information required for addressing Siemens 3964 devices connected via an RS232 port or a terminal server.

DIGITAL DAS for Siemens 3964

The DIGITAL DAS for Siemens 3964 is a BASEstar Open device connectivity module that implements the Siemens 3964[R] and 3964[R]_RK512 protocols.

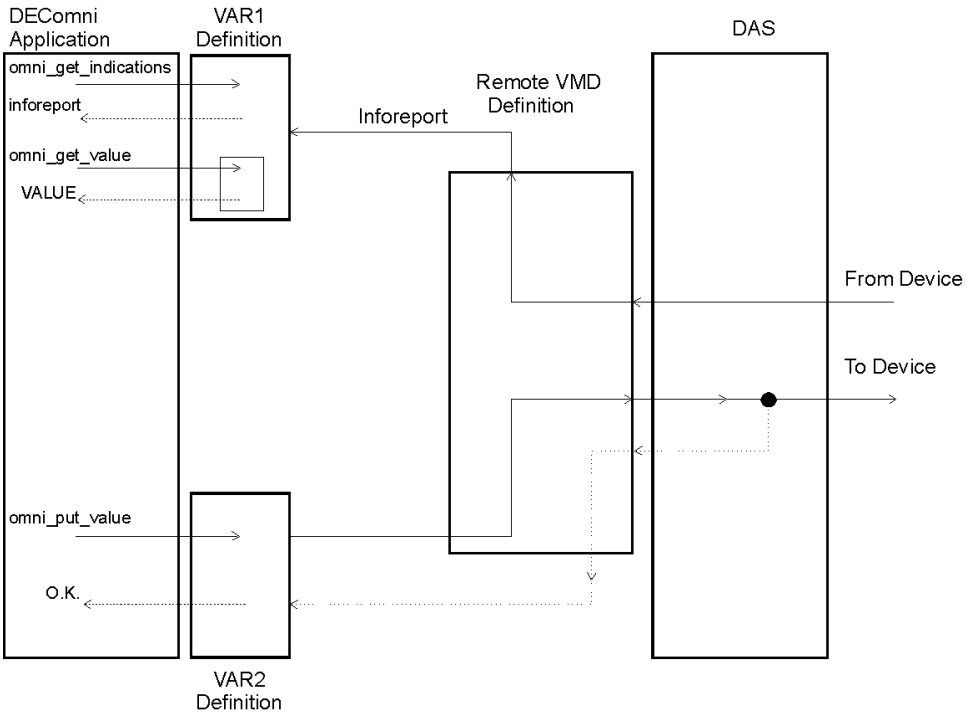
Exchanging Data Messages With a Siemens 3964 Device (3964[R] Protocol)

The DAS for Siemens 3964 allows a DEComni API application to communicate with a Siemens 3964 device according to the 3964 [R] protocol.

Figure 6-2 shows the DEComni API definitions and the DEComni API procedures an application must use to receive data messages from (and send data messages to) a remote Siemens 3964 device.

Before a DEComni API application can exchange data messages with a remote Siemens 3964 device, it must create the definitions of a remote VMD and its associated variables.

Figure 6-2: DEComni Definitions and API Procedures (3964[R] Protocol)



Receiving Data

In order to receive data messages from a remote Siemens 3964 device, a DEComni API application must issue an *omni_get_indications* procedure call. When DEComni API receives a data message from the remote device, it generates an Information report on the remote VMD definition. This causes the completion of the *omni_get_indications* procedure.

The returned context refers to the "reception variable" in which DEComni API returns the received data message. The “reception variable” is the first Named_Variable that has been created for the remote VMD definition. If no Named_Variable has been created, the first Unnamed_Variable is used. If no variable is associated with the remote VMD, any data received from the remote device are lost and the *omni_get_indications* procedure returns an error.

DEComni API interprets the data received from the remote device in one message according to the MMS data type associated with the "reception variable". Note that if an insufficient number of bytes is received, the value returned in the user buffer is unpredictable; if, instead, the number of the received bytes exceeds that required to meet the data type, the exceeding bytes are lost.

Sending Data Messages

To send a data message to a Siemens 3964 device, a DEComni API application must issue an *omni_put_value* procedure call on the remote VMD definition.

The DEComni API application can associate several variables with the local VMD, and can use each of them to send data messages of different lengths.

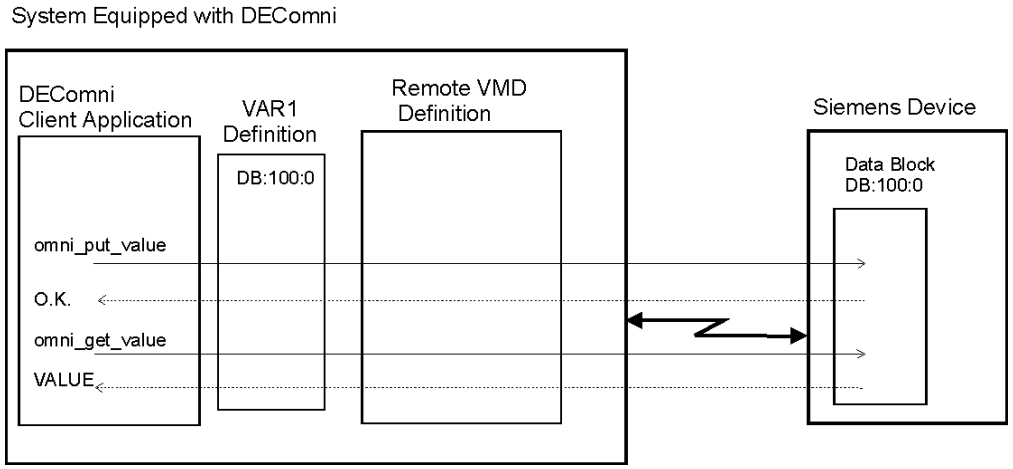
The DAS for Siemens 3964 sends the remote device a number of characters taken from the user buffer equal to the size of the MMS data type that has been associated with the variable definition used for the send operation.

The *omni_put_value* procedure returns control to the DEComni API application as soon as the DAS for Siemens 3964 has sent the data through the local port successfully.

Accessing the Memory of a Siemens 3964 Device (3964[R]_RK512 Protocol)

The DAS forSiemens 3964 allows a DEComni API application to access the memory of a Siemens 3964 device via the 3964 [R]_RK512 protocol. The Siemens 3964 device is modeled by a remote VMD definition. A DEComni API application can act as client, that is, it can read and write the values of memory blocks on a Siemens 3964 device.

Figure 6-3: Using the DEComni API to Access the Memory of a Siemens 3964 Device



To act as a client of a Siemens 3964 device (that is, to read data from, and write data to, a physical memory location of a Siemens 3964 device), a DEComni API application uses a remote VMD definition. An Unnamed Variable associated with this VMD must specify the address of a valid memory block on the addressed Siemens 3964 device.

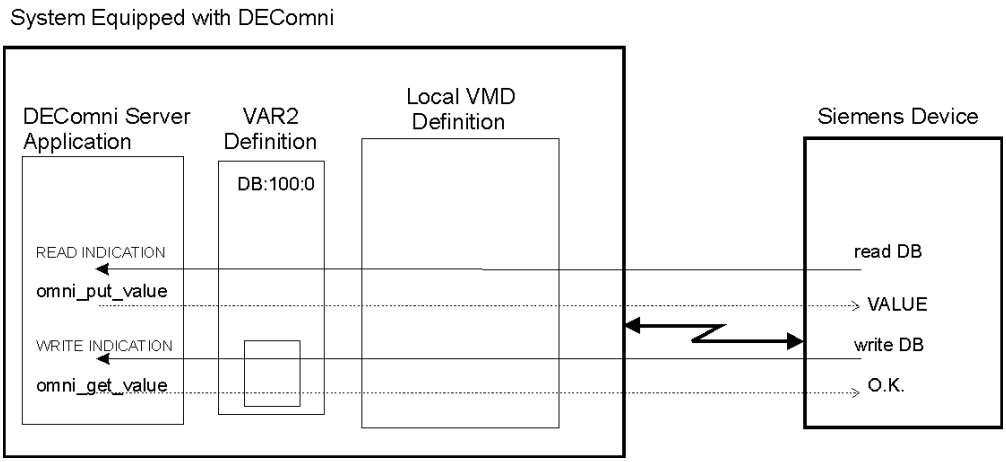
The same Unnamed Variable can be used in read and write operations.

A DEComni API client application calls the *omni_put_value* (or the *omni_get_value*) procedure, and specifies the handle of an Unnamed Variable associated with a remote VMD to write (or read) a memory block of the remote device.

Emulating a Siemens 3964 Device (3964[R]_RK512 Protocol)

To act as a server of a remote Siemens 3964 device (that is, to emulate a Siemens 3964 device and therefore to fulfill write and read requests coming from a remote Siemens 3964 device), a DEComni API server application uses a local VMD definition and its associated variables (see Figure 6-4).

Figure 6-4: Using the DEComni API to Emulate a Siemens 3964 Device



Each Unnamed Variable associated with the local VMD must specify the address of a valid memory block at the emulated Siemens 3964 device. In this way, the block can be correctly addressed by other Siemens devices.

When DEComni API receives a read (or write) request from a remote device, it generates a Read (or Write) Indication, which causes the completion of the *omni_get_indications* procedure call.

The context returned to the application by the *omni_get_indications* procedure refers to the variable, among those defined for the local VMD, whose Siemens 3964 memory address matches that specified in the remote client request.

MMS Types and Siemens 3964 Memory Block Sizes

The *omni_get_value* procedure returns in the user buffer a number of characters that is equal to the size of the MMS Type associated with the variable used in the read operation. The size (in bytes) of this data type must therefore be equal to the size of the data that are contained in a data block (represented by the variable). A similar consideration applies to the *omni_put_value* procedure.

In order to simplify the calculation of the size of an MMS Type, Appendix C lists the sizes of the basic DEComni MMS types.

7
Configuring DEComni API

You must configure DEComni API in order to allow a DEComni API application to access a Siemens 3964 device. Configuring DEComni API implies creating the required definitions using the following API procedures:

- *omni_define*
- *omni_get_definition*
- *omni_modify_definition*

Note

For details of the DEComni API definition attributes whose values are not specified in this chapter, refer to the *DEComni API and DEComni MMS Programmer's Guide* for your platform.

To configure DEComni API, you must:

- Register the appropriate entries in the ODS database.
- Set the required application profile.
- Create the required DEComni VMD and Variable definitions.

Registering ODS Entries

For each Siemens 3964 device you must create an ODS entry which specifies Siemens 3964 protocol-specific information. See Appendix A for details.

Setting the Application Profile

An application must invoke the *omni_set_application_profile* procedure to specify the appropriate application profile. This ensures that the definitions created after this procedure is called are correctly associated with the desired Siemens 3964 protocol. Table 7-1 lists the available application profiles.

Table 7-1: DEComni Application Profiles

Protocol Name	Application Profile Constant
3964	omni_c_app_profile_3964
3964R	omni_c_app_profile_3964r
3964_RK512	omni_c_app_profile_3964_rk512
3964R_RK512	omni_c_app_profile_3964r_rk512

The following example shows how an application can invoke the *omni_set_application_profile* procedure to set the application profile for the Siemens 3964 protocol.

```
status=omni_set_application_profile (omni_c_app_profile_3964)
```

Creating VMD Definitions

A Siemens 3964 device is modeled by means of a VMD definition. Table 7-2 lists the mandatory attributes that you must specify for a correct definition of a VMD.

Table 7-2: DEComni VMD Attributes

Attribute Name	Description and Values
omni_c_attr_name	The name of a VMD definition.
omni_c_attr_appl_simple_name	The name of an ODS entry. See Appendix A for a description of ODS entry attributes.
omni_c_vmd_max_segment	The maximum length of the data exchanged with the Siemens 3964 device. For the 3964[R] protocol, there is no limit in the size of the message exchanged. You can specify any value, provided it is supported by DEComni API. For the 3964[R]_RK512 protocol, there is no limit in the size of the message exchanged between DEComni API applications and Siemens 3964 devices, but the message is segmented for transmission over the communication line. This attribute must specify the segment size, which must be less than, or equal to 138 (including 10 bytes for the header).

Using the 3964[R] Protocol to Create a VMD for Communication

A VMD is a gate that DEComni API makes available to its applications to exchange data messages with a remote Siemens 3964 device using the Siemens 3964[R] protocol.

Using the 3964[R]_RK512 Protocol to Create a VMD for Communication

Depending on the operation you want to perform, you must define a remote VMD, a local VMD, or both:

- A remote VMD models a remote Siemens 3964 device and associated variables. For a remote VMD, a DEComni API client application acts as a client that writes and reads the value of memory blocks physically located on the associated Siemens 3964 device.
- A local VMD models a Siemens 3964 device emulated on the host system. For a local VMD, BASEstar Open acts as a server that fulfills write and read requests coming from a Siemens 3964 device.

Creating Variable Definitions

You must create the definitions of the variables that will be used for exchanging data with Siemens 3964 devices. Table 7-3 lists all the mandatory attributes that you must specify for correct definition of a variable.

Table 7-3: DEComni Variable Attributes

Attribute Name	Description and Values
omni_c_attr_app_type_descr	Name of the application type. The associated MMS type must be appropriate and must have a sufficient size in order to contain the data exchanged with the Siemens 3964 device. Appendix C specifies the size of the basic DEComni MMS types. Remember that the data type associated with these variables drives the DAS for Siemens 3964 in encoding/decoding the data exchanged between your application and the Siemens 3964 device.
omni_c_attr_address_type	Only significant for Unnamed Variables, and must be set to omni_c_address_unconstrained.
omni_c_attr_address_string	Only significant for Unnamed Variables. The value that you must provide depends on the protocol used. For the 3964[R] protocol, this attribute is not considered by the DAS for Siemens 3964, but you can use it to describe the type of the data message exchanged with the device. For the 3964[R]_RK512 protocol, this attribute must specify a valid memory block address of the Siemens 3964 device. See also the section below.

Using the 3964[R] Protocol to Create Variables for Communication

To exchange data with Siemens 3964 devices, you can use Named or Unnamed Variables.

The first Named Variable you define at a given VMD is assumed to be the "reception variable", and it will be used for read operations only. If no Named_Variables are defined, the first Unnamed Variable is used.

You can define as many Named or Unnamed Variables associated with the VMD as you need to send data to a Siemens 3964 device. Each of these variables can have a different data type, which allows you to send different types of data to the Siemens 3964 device.

Using the 3964[R]_RK512 Protocol to Create Variables for Communication

When using the 3964[R]_RK512 protocol, you must use Unnamed Variables whose omni_c_attr_address_string attribute specifies a valid memory address of a Siemens 3964 device. See also Appendix D.

8
Programming

This chapter describes how you can use the DEComni API procedures to write a DEComni API application that is able to exchange data with a Siemens 3964 device. Bear in mind the following considerations when using the API procedures to write your application:

- General syntax and programming rules, that are described in the *DEComni API and DEComni MMS User Guide* for your platform, are also valid when the API procedures are used in the DAS for Siemens 3964 context.
- When using the *DEComni API and DEComni MMS User Guide* for your platform, ignore the descriptions referring to services and functions that are not supported by the DAS for Siemens 3964.

Programming With the 3964[R] Protocol

Programming with the 3964[R] protocol means preparing a DEComni API application that is able to exchange data messages with a remote Siemens 3964 device using the 3964[R] protocol.

Table 8-1 lists the DEComni API procedures that a DEComni API application can invoke to exchange data with a Siemens 3964 device.

Table 8-1: DEComni API Procedures for Data Exchange (3964[R] Protocol)

Procedure Name	Description
omni_abort	Causes the forced closure of a connection. Only local operations are performed.
omni_conclude	Causes the forced closure of a connection. It behaves like the *omni_abort* procedure.
omni_connect	Opens a connection. The request always completes successfully and the control is immediately returned to the calling application. Only local operations (for example, opening of the local port) are performed, and no negotiation is performed with the remote device.
omni_get_indications	Returns connection-related and Inforeport indications (for example, data received from the Siemens 3964 device, or connection aborted).
omni_get_value	Returns the data associated with a previously detected Inforeport indication.
omni_put_value	Writes data to the Siemens 3964 device. It is an unconfirmed operation because the procedure call completes as soon as the data have been sent through the local port.

Opening a Connection

To open a connection with a remote Siemens 3964 device when using the 3964[R] protocol, a DEComni API application must call the *omni_connect* procedure and specify the handle of the remote VMD definition.

A DEComni API application must call the *omni_get_indications* procedure to monitor an incoming omni_c_ind_abort indication which notifies that the connection has been terminated for any reason.

To close a connection, an application must call the *omni_conclude* or the *omni_abort* procedure. These procedures perform the same operations, have only local effect, and cause a forced closure of the connection.

Sending Data Messages

To send a data message to a Siemens 3964 device, a DEComni API application must call the *omni_put_value* procedure, and specify the handle of a Named or Unnamed_Variable that is associated with the corresponding VMD definition.

The *omni_put_value* returns control to the application as soon as the DAS for Siemens 3964 has sent the data through the local port.

Receiving Data Messages

A DEComni API application that wants to wait for data coming from a remote Siemens 3964 device must call the *omni_get_indications* procedure, and specify the handle of the remote VMD associated with that device. Each time DEComni API receives data from the remote

device, it generates an Inforeport indication that causes the completion of the *omni_get_indications* procedure call.

The *omni_get_indications* procedure returns to the application a context which *always* refers to the "reception variable": the application must issue an *omni_get_value* procedure call, and specify this context to get the value of the "reception variable", that is, the data received from the device.

Programming With the 3964[R]_RK512 Protocol

Programming with the 3964[R]_RK512 protocol means preparing server applications that are able to perform one or both of the following tasks:

- To behave as a client of a Siemens 3964 device, and therefore to read and write memory blocks of that device.
- To behave as a server that fulfills read and write requests coming from remote Siemens 3964 devices and addressed to the locally emulated device.

Table 8-2 lists the API procedures that an application invokes to exchange data with a Siemens 3964 device.

Table 8-2: DEComni API Procedures for Data Exchange (3964[R]_512 Protocol)

Procedure Name (Local/Remote VMD)	Description
omni_abort (remote)	Causes the forced closure of a connection.
omni_conclude (remote)	Causes the forced closure of a connection. It behaves like the *omni_abort* procedure.
omni_connect (remote)	Opens a connection to the Siemens 3964 device modeled by the specified remote VMD definition. The request always completes successfully and the control is immediately returned to the calling application. Only local operations (for example, opening of the local port) are performed, and no negotiation is performed with the remote device.
omni_listen (local)	Makes the local VMD accessible to remote clients. The request always completes successfully and the control is immediately returned to the calling application. Only local operations (for example, opening of the local port) are performed, and no negotiation is performed with the remote device.
omni_get_indications (remote)	Obtains connection-related indications and must always specify the handle of the remote VMD. When returning a Read Request or Write Request indication (corresponding to read or write requests issued by a Siemens 3964 device towards the locally emulated Siemens 3964 device), the returned context refers to a variable that is associated with the local VMD definition.
omni_get_value (both local and remote)	When specifying a variable associated with a remote VMD, it allows the DEComni API client application to read the value of the associated Siemens 3964 device memory block. When specifying a variable associated with a local VMD, it allows the DEComni API server application to obtain the value associated with a Write Request indication.
omni_put_value (both local and remote)	When specifying a variable associated with a remote VMD, it allows the DEComni API client application to write the value of the associated Siemens 3964 device memory block. When specifying a variable associated with a local VMD, it allows the DEComni API server application to provide the DAS for Siemens 3964 with the value required to fulfill a previously received Read Request indication.

Opening and Closing a Connection

protocolAn application can use the *omni_connect* or *omni_listen* procedure to open a connection. The choice depends on specific application requirements and role (for example, if it is a server or client application), and on application programming interface rules.

Both the *omni_connect* and *omni_listen* procedures have only local effect: they open the port that was configured in the ODS entry and was associated with the remote (or local) VMD.

When using both a remote and a local VMD, you are suggested to define identical ODS entries (apart from the name) to be associated with the local and remote VMDs; in particular, you must specify the same operating system port.

A DEComni API application must call the *omni_get_indications* procedure to monitor an incoming omni_c_ind_abort indication which notifies that the connection has been terminated for any reason.

To close a connection, an application calls the *omni_conclude* or the *omni_abort* procedure. These procedures perform the same operations, have only a local effect, and cause a forced closure of the connection.

Reading and Writing Memory Blocks of a Siemens Device

To read (or write) the value of a memory block of a remote Siemens 3964 device, a DEComni API client application must call the *omni_get_value* (or *omni_put_value*) procedure, and specify the handle of an Unnamed Variable associated with corresponding remote VMD definition.

Fulfilling Requests for an Emulated Device

.A DEComni API server application that wants to receive and fulfill Read and Write Request indications related to a locally emulated Siemens 3964 device, must call the *omni_get_indications* procedure, and specify the handle of the appropriate remote VMD definition.

When the *omni_get_indications* procedure returns control to the application with an omni_c_ind_read indication, the application must issue an *omni_put_value* procedure call to provide the value to be returned to the client application.

The *omni_put_value* procedure must specify the context returned by the *omni_get_indications* procedure. The context refers to the variable definition associated with the local VMD whose device memory address, specified in the omni_c_address_unconstrained attribute, corresponds to that specified in the client request.

Similarly, when the *omni_get_indications* procedure returns control to the application with an omni_c_ind_write indication, the application must issue an *omni_get_value* procedure call to obtain the value provided by the client application; this value will be used to update the value of the variable specified in the context.

PART III

Appendices

A

ODS Entry Attributes for the Siemens 3964 Protocol

This appendix describes the ODS attributes (and their associated values) that allow you to create the ODS entries required to access Siemens 3964 devices. An ODS entry contains the addressing parameters needed by the Siemens 3964 protocol, and must be specified in VMD definitions used to access Siemens 3964 devices.

The information in this appendix is valid for all the supported protocols, and for both the BASEstar Open and DEComni API environments.

For details of how to use the ODS CLI, refer to the DEComni API documentation included in your documentation kit.

Creating ODS Entries for the Siemens 3964 Protocol

Bear in mind that a connection to a Siemens 3964 device must be based on a LAT port.

The table below provides a description of each ODS attribute, together with its class and accepted values.

ODS Entry Attributes for the Siemens 3964 Protocols

Attribute	Description and Allowed Values
\oc	Identifier of the ODS object class. Must be set to DTK232.
portname	Name of the LAT communication port. Examples of valid values are tty00, TT00 and COM1.
baudrate	The supported speed is 9600 bits/sec.
charlen	The number of data bits must be set to 8.
stopbit	The number of stop bits must be set to 1.
prtycheck	The parity check must be set to NONE.
flowctrl	The flow control is not provided and must be set to NONE.
modem	Modem presence. Allowed values are 0 (modem not present, direct connection), and 1 (modem present, port handles modem signal and handshaking).
stationid	Not significant.
lev7conn	Specifies if the connection is handled with messages at application level. Must be set to 0 because Siemens 3964 protocols do not exchange messages for connection at application level.
rsptimeout	Timeout (in seconds) for the reception of the response to a confirmed service, after which the service is terminated with failure. Only applicable to the 3964[R]_RK512 protocol.

Example A–1 shows a configuration file command that can be used to create an ODS registration valid for all the supported Siemens 3964 protocols.

Example A–1: ODS Entry for LAT Port

```
register directory name "/cn=SIEMENS_3964" attributes "/oc=DTK232
/levconn=0 /portname=tty00 /baudrate=9600 /charlen=8 /stopbit=1
/prtycheck=NONE /flowctrl=NONE /modem=0"
```

Configuring a LAT Port

OpenVMS Platform

You can physically connect a device to a LAT port.

To use a LAT port, the port must be configured using the LAT control program. For example, to configure port LTA001:, connected to the PORT_1 port of the TSRV1 terminal server, perform the following steps:

1. Log onto an account with the OPER privilege.
2. Invoke the LAT control program:

   ```
   $ mc latcp
   ```

   Create the LAT port:

   ```
   LATCP> create port LTA001: /application
   ```

3. Set the Terminal Server node and port names:

   ```
   LATCP> set port LTA001: /node=TSRV1/port=PORT_1
   ```

DIGITAL UNIX Platform

To set up the LAT service on a DIGITAL UNIX system use the following command:

```
/usr/sbin/latsetup
```

Please refer to the *DIGITAL UNIX Network Manager's Guide* or to the online man pages for details of the `latsetup` command.

If you are not sure about which terminal special files are assigned to the LAT service, you can issue the following command:

```
file /dev/tty*
```

and look for the entries like:

```
/dev/tty02: character special (5/0) LAT #0 DS200 terminal #0
```

Once you know which special files are assigned to the LAT services, you can set a reverse LAT connection to a terminal server port by issuing the command:

```
/usr/sbin/latcp -A -ptty02 -HSERVER -RPORT_1
```

where *tty02* is the name of the local LAT port you want to use for the connection, *SERVER* is the terminal server name, and *PORT_1* is the name of the port you want to connect to, that is, the port on which the device is hooked on.

In order to have this command automatically executed at system startup, you can add the command to the */sbin/init.d/lat* file at the end of the start case (look at the end of file).

In the example above, the LAT port *tty02* is used, which corresponds to the special file */dev/tty02*. To use that port from a DEComni API application, you should register a VMD entry by means of the odscl utility, using tty02 as the value for the /PORTNAME attribute (note that /dev is omitted).

B
BASEstar Open Datatype Sizes

The number of bytes involved in a write or read operation depends on the datatype associated with the Device Services Variable in use.

The table below lists the size (in bytes) of the basic BASEstar Open datatypes. Use this table to ensure that the size of a datatype is appropriate for the data to be exchanged with the Siemens 3964 device.

BASEstar Open Simple Datatype Sizes

Datatype	Exchanged as
BOOLEAN	1 byte
TIME	Not supported
VISIBLE_CHAR	1 byte
INTEGER_8	1 byte
INTEGER_16	2 bytes
INTEGER_32	4 bytes
UNSIGNED_8	1 byte
UNSIGNED_16	2 bytes
UNSIGNED_32	4 bytes
F_FLOAT	4 bytes (IEEE float format)
D_FLOAT	Not supported
BCD1, BCD2, BCD3, BCD4, BCD5, BCD6, BCD7, BCD8	Not supported
OCTET	1 byte
BIT	1 byte
OBJECT_STATUS	Not supported
OBJECT_NAME	Not supported
ARRAY_DATATYPE	As many times as the number of elements it contains. A variable length array is treated as an array having a fixed length; the maximum length is assumed
STRUCTURE_DATATYPE	Packing of the Simple_Datatypes without padding

C
DEComni MMS Type Sizes

The number of bytes involved in a write or read operation depends on the MMS type associated with the DEComni variable in use.

The table below specifies the size (in bytes) of the basic DEComni MMS types. This allows you to define a data type with a size appropriate for the data to be exchanged with the Siemens 3964 device.

DEComni MMS Type Sizes

Data Type	Exchanged as
omni_c_mmstype_boolean	1 byte
omni_c_mmstype_bit_str	(*n* + 7)/ 8) bytes, where *n* is the number of bits in the string
omni_c_mmstype_integer	1, 2 or 4 bytes, depending on omni_c_symbol_size, that specifies the size of the variable (8, 16, or 32 bits)
omni_c_mmstype_unsigned	1, 2 or 4 bytes, depending on omni_c_symbol_size, that specifies the size of the variable (8, 16, or 32 bits)
omni_c_mmstype_floating_point	4 bytes (SINEC float format)
omni_c_mmstype_real	Not supported
omni_c_mmstype_octet_str	*n* bytes (as returned by sizeof())
omni_c_mmstype_visible_str	*n* bytes (as returned by strlen() if varying or as returned by sizeof() otherwise)
omni_c_mmstype_generalized_time	Not supported
omni_c_mmstype_binary_time	Not supported

D
Siemens 3964 Memory Addresses

This appendix describes how to specify a valid Siemens 3964 memory address. For full details, refer to the Siemens product documentation for the specific Siemens 3964 device.

The syntax example below illustrates how to define a valid memory address of a Siemens 3964 device:

```
BlockType:BlockNumber:StartingAddress
```

BlockType can assume any of the values specified in the table below.

DEComni API *BlockType* Attribute Values

Block Type	Description	Block Number	Starting Address
DB	Data Block	From 1 to 255	From 0 to 2047
DX	Extended DB	From 1 to 255	From 0 to 2047
IB	Input Byte		From 0 to 255 For S5 150: 0 ÷ 199
QB	Output Byte		From 0 to 127
FY	Flag Byte		From 0 to 127
PY	Peripheral Byte		From 0 to 255
CB	Counter Location		From 0 to 255
TB	Timer Location		From 0 to 255
AS	Absolute Address		From 0 to FFFF
RS	System Address		From 0 to 511
OY	Extended I/O		From 0 to 511

E

DEComni API Network IOSB Messages

This appendix lists the DEComni API messages returned by the DAS for Siemens 3964 in the Network IOSB.

dastk_s_success 80125953

Explanation: das toolkit service completed successfully.

User action: none.

dastk_s_asoabort 80125962

Explanation: the association has been aborted by the peer.

User action: none.

dastk_s_asoalrdyexist 80125970

Explanation: an association with the same parameters already exists.

User action: change vmd addressing parameters.

dastk_s_nolicense 80125978

Explanation: no valid licenses have been loaded.

User action: install and load a valid pak for this product.

dastk_s_nosuchasn 80125986

Explanation: the specified application simple name has not been found in ODS.

User action: check ODS configuration.

dastk_s_nosuchaso 80125994

Explanation: the specified association does not exist.

User action: check if you refer to the right association.

dastk_s_reqabort 80126002

Explanation: the specified request has been aborted.

User action: none.

dastk_s_reqtimeout 80126010

Explanation: the timer for the specified request expired.

User action: none.

dastk_s_connerr 80126018

Explanation: a generic error occured on the communication layer.

User action: none.

dastk_s_insfmem 80126026

Explanation: insufficient virtual memory.

User action: none.

1333dastk_s_lentoolrg 80126034

Explanation: pdu length exceeds the maximum allowed value.

User action: redesign user application to use smaller object values.

dastk_s_generr 80126042

Explanation: generic error.

User action: none.