Entire Net-Work

Administration

Version 6.3.2

April 2014

ADABAS & NATURAL

This document applies to Entire Net-Work Version 6.3.2 and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: WCPMF-ADMIN-632-20191112

Table of Contents

Entire Net-Work Administration

This document addresses administrators responsible for configuring and running an Entire Net-Work environment once the product is installed.

The Entire Net-Work Administration document is organized as follows:

Heterogeneous Platform Considerations	Describes the data translation capabilities of Entire Net-Work to allow heterogeneous platform configurations.
IUCV Line Driver Administration	Provides information for administrators responsible for configuring and running the Entire Net-Work IUCV line driver once Entire Net-Work is installed.
VTAM Line Driver Administration	Provides information for administrators responsible for configuring and running the Entire Net-Work VTAM line driver once Entire Net-Work is installed.
Connecting to UES-Enabled Adabas Databases	Describes how to connect to UES-enabled databases.
Tuning the Compression Algorithm	Describes how to tune the compression algorithm used for message compression on LINK statements.
Target ID Handling in the Network	Describes how Entire Net-Work handles duplicate target IDs in the network
Estimating Entire Net-Work Storage Requirements	Provides tables to assist in estimating the storage requirements of Entire Net-Work.

1 Conventions

Notation *vrs* or *vr*: When used in this documentation, the notation *vrs* or *vr* stands for the relevant version, release, and system maintenance level numbers. For further information on product versions, see *version* in the *Glossary*.

This document covers the following topics:

Syntax Conventions

The following table describes the conventions used in syntax diagrams of Entire Net-Work statements.

Convention	Description	Example
uppercase, bold	Syntax elements appearing in uppercase and bold font are keywords. When specified, these keywords must be entered exactly as shown.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** The syntax elements DRIVER, TCPI, and DRVCHAR are Entire Net-Work keywords.
lowercase, italic, normal font	Syntax elements appearing in lowercase and normal, italic font identify items that you must supply.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** The syntax element *driver-char* identifies and describes the kind of value you must supply. In this instance, you must supply the special character used to designate that an operator command is directed to the TCP/IP line driver, rather than to a specific link.
underlining	Underlining is used for two purposes: 1. To identify default values, wherever appropriate. Otherwise, the defaults are explained in the accompanying parameter descriptions. 2. To identify the short form of a keyword.	**DRIVER TCPI [DRVCHAR =** *driver-char* **| #]** In the example above, # is the default that will be used for the DRVCHAR parameter if no other record buffer length is specified. Also in the example above, the short version of the DRVCHAR parameter is D.

Convention	Description	Example
vertical bars (|)	Vertical bars are used to separate mutually exclusive choices. **Note:** In more complex syntax involving the use of large brackets or braces, mutually exclusive choices are stacked instead.	DRIVER TCPI API = { BS2 | CNS | EZA | HPS | OES } In the example above, you must select BS2, CNS, EZA, HPS, or OES for the API parameter. There are no defaults.
brackets ([ ])	Brackets are used to identify optional elements. When multiple elements are stacked or separated by vertical bars within brackets, only one of the elements may be supplied.	DRIVER TCPI [DRVCHAR = *driver-char* | #] In this example, the DRVCHAR parameter is optional.
braces ({ })	Braces are used to identify required elements. When multiple elements are stacked or separated by vertical bars within braces, one and only one of the elements must be supplied.	DRIVER TCPI API = { BS2 | CNS | EZA | HPS | OES } In this example, one of the following values is required for the API parameter: BS2, CNS, EZA, HPS, or OES.
other punctuation and symbols	All other punctuation and symbols must be entered exactly as shown.	LINK *linkname* TCPI [NETADDR = *n1.n2.n3.n4*] [,][-] In this example, the periods must be specified in the IP address. In addition, options must be separated by commas and dashes should be used as needed to indicate that parameter settings continue on the next line.

Syntax Rules

The following rules apply when specifying Entire Net-Work parameter statements:

- Each Entire Net-Work parameter statement occupies positions 1 - 72 of at least one line.
- The statement type (NODE, LINK, TRANSDEF, or DRIVER) must be specified as the first non-blank item on the statement.
- The node name, driver name, translation definition function, or link name follows the statement type, separated by at least one blank (space).

- Keyword parameters may be specified following either the node name on NODE statements or the driver name on DRIVER and LINK statements. Keyword parameters are separated from their arguments by an equal (=) sign, and from other keyword parameters by at least one blank (space) or a comma (,).
- When the acceptable values for a parameter are Y and N (yes and no), any other value is treated as an N, unless there is a documented default, and processing continues without any warning.
- When the acceptable values for a parameter fall within a range (e.g., 1 - 2147483647) and a value outside the range is specified, the value is automatically reset to the maximum value within the range, unless documented otherwise for the parameter. Processing continues without any warning.
- A statement can be continued beginning in any column of the next line by specifying a dash (-) as the last nonblank character in any column of the current line, before column 73.
- Comment lines begin with an asterisk (*) in position 1 and can be inserted anywhere in the statement sequence.
- Some keywords may require a list of subparameters separated by commas; the list must be enclosed in parentheses ( ) unless only the first subparameter is to be entered. Omitted ("defaulted") subparameters must be represented by placeholder commas if subsequent parameters are to be entered. The following are examples of correct subparameter strings:

```
KEYWORD=(value1,value2,value3)
KEYWORD=(value1,,value3)
KEYWORD=(,,value3)
KEYWORD=(,value2)
KEYWORD=value1
```

- Hexadecimal keyword values can be entered by prefixing the value with an "X". For example:

```
LINK . . . ADJID=X0064, . . .
```

2 About this Documentation

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
`Monospace font`	Identifies service names and locations in the format *`folder.subfolder.service`*, APIs, Java classes, methods, properties.
`Italic`	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at **http://documentation.softwareag.com**. The site requires credentials for Software AG's Product Support site Empower. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at **https://empower.softwareag.com/**.

You can find product information on the Software AG Empower Product Support website at **https://empower.softwareag.com**.

To submit feature/enhancement requests, get information about product availability, and download products, go to **Products**.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the **Knowledge Center**.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at **https://empower.softwareag.com/public_directory.asp** and give us a call.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at **http://techcommunity.softwareag.com**. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

3 Heterogeneous Platform Considerations

The major obstacle in communication between client and server applications on diverse systems is that different computer systems use different formats to represent similar types of data. The mainframe (IBM. BS2000, or Facom) is architecturally an EBCDIC format, Big-Endian machine, while most machines running OpenVMS, UNIX, and Windows operating systems are ASCII format machines that use a combination of Big-Endian and Little-Endian architectures.

Note: For the lowest addressable byte, Big-Endian architecture uses the most significant byte; Little-Endian architecture uses the least significant byte.

These different architectures mean that data must be translated in communications between heterogeneous client/server applications to allow for interoperability. Data translation is required in order for a mainframe (IBM, BS2000, or Facom) client application program to communicate with a UNIX or OpenVMS Adabas server or for a UNIX or OpenVMS client application program to access a mainframe (IBM, BS2000, or Facom) Adabas server.

The data translation that is part of this client/server communication occurs partially within Entire Net-Work and the Adabas server. The client application need not be concerned with data conversion.

Entire Net-Work's translation process is centered around the format and length of each field specified in the search and format buffers that are passed with each Adabas call, along with special translation definition parameters. When a request goes through the network conversion routines, each individual field is translated according to the format and length defined for it in the associated search or format buffer.

Note: The format of all fields in the record and value buffers must be fully qualified with length and field type if Entire Net-Work will be performing the translation; otherwise, response code 229 (ADARSP229) is returned to the client application.

Enhanced Translation Capabilities

In addition to fully qualified field definitions, Entire Net-Work provides enhanced translation capabilities that allow:

- proper data translation of superfields and superdescriptors
- alternative EBCDIC /ASCII translation tables
- user exits that override the normal translation process
- special handling of field format "X".

Enhanced translation capabilities apply to communications between a client application running on any non-mainframe Entire Net-Work node (e.g., Windows, UNIX, OpenVMS) and an Adabas database on an Entire Net-Work mainframe node.

Superfields and Superdescriptors

A superfield is a field that logically contains multiple fields, each of which can have its own format. For example, superfield "SF" can consist of two fields, "SA" which has the format P9, and "SB" which has the format A50. Similarly, a superdescriptor comprises multiple descriptors.

Entire Net-Work supports the translation of superdescriptors and superfields on a field by field basis. This allows superdescriptors and superfields accessed through Entire Net-Work to be properly translated.

Alternative Translation Tables

Alternative EBCDIC to ASCII and ASCII to EBCDIC translation tables can be specified with Entire Net-Work, allowing you to accommodate unusual translation requirements. For example, if the ASCII character set in use includes characters different from the standard set, a translation table can be provided to translate EBCDIC characters into the required ASCII characters.

User Translation Exits

Entire Net-Work allows user-written translation exits to override the normal translation process. The Entire Net-Work administrator can develop a translation exit to perform extended translation, such as conversion to or from double-byte character format. User exit modifications can be applied to any field translated by Entire Net-Work.

Special Handling of Field Format X

Within the Entire Net-Work conversion routines, a field format of "X" in the format or search buffer instructs Entire Net-Work to perform no translation of the field. Entire Net-Work only changes the "X" to an "A" before passing the request to the server.

Enhanced Translation Definitions

Translation definition statements are required in order to use Entire Net-Work's enhanced translation capabilities. These definitions can be either:

- permanently added to the DDKARTE or operating system-dependent configuration member, like other Entire Net-Work statements or
- entered at the console as operator commands.

For each superfield or superdescriptor to be translated, a translation definition statement must be added to define the format and length of each field contained in the superfield.

If an alternative translation table or a translation exit is used, a translation definition statement must be added to identify the name of the translation module. A translation module can be applied

to all fields within a given file or database. If the name of a translation table or translation exit is specified in a TRANSLATE DEFINE statement for the database or file, it is applied to all fields in the database or file that are specified in lower-level translation definition statements. The translation definition statement for the individual fields need not include the translation module specification if the module has already been specified for the entire database or the entire file.

Translation Definition Statement Types

There are three types of translation definition statements:

Statement Type	Used to
TRANSlat DEFine | Add	add a translation definition.
TRANSlat DELETE | Remove	delete a translation definition.
TRANSlat DISplay | LISt	display the currently active translation definitions.

Adding Translation Definitions

The TRANSlat DEFine | Add statement is used to add a translation definition, in the following format:

```
TRANSlat DEFine | ADD  ID=(dbid,file), -
                       FIeld=field,-
                       TOAscii=etoatbl,-
                       TOEbcdic=atoetbl,-
                       EXIT=userexit,-
                       FOrmat=(F1,L1,...,F20,L20)
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Note: If a required parameter is missing, the message NET004I: INVALID VALUE FOR KEYWORD FIELD is displayed. If this occurs, reissue the Add statement with all the required parameters.

```
ID=(dbid,file)
```

Specifies the database ID and file number to which this statement applies. Specifying "0" for a database or file identifier indicates all databases or all files. A zero dbid cannot be specified with a non-zero file identifier.

```
Field=field
```

Specifies the two byte Adabas short name for the field. Not specifying this operand indicates that the translation definition applies to all fields within the specified database and file. This operand must be specified if the FORMAT operand is specified. This operand cannot be specified if either

the database ID or file number is zero (all databases or all files). If this operand is omitted, the TOAscii, TOEbcdic, or EXIT operand must be specified.

```
TOAscii=etoatbl
```

Specifies the name of an optional user supplied EBCDIC to ASCII translation table. The name must be 1 to 8 characters long, and the load module or core image module must be accessible at the time the translation definition is entered.

```
TOEbcdic=atoetbl
```

Specifies the name of an optional user supplied ASCII to EBCDIC translation table. The name must be 1 to 8 characters long, and the load module or core image module must be accessible at the time the translation definition is entered.

```
EXIT=userexit
```

Specifies the name of a user supplied translation exit. The name must be 1 to 8 characters long, and the load module or core image module must be accessible at the time the translation definition is entered.

```
FOrmat=(F1,L1,....,F20,L20)
```

Defines 1 to 20 pairs of formats and lengths. These format and length pairs override the format supplied in the search or format buffer and are used to describe the format conversion performed by Entire Net-Work. The formats must be valid Adabas field formats (A,B,F,G,P,U,X) and the length must be valid for the related format.

The specified formats apply to the subfields contained in the field specified with the Field operand. Format can be specified only if the Field operand is also specified.

If the Format operand is omitted, translation is based on the format supplied in the search or value buffer of the Adabas call being processed.

Validating a Translation Definition

When a translation definition is entered, it is validated; then an attempt is made to add it to the translation definition table. Before adding it, however, the identifying triplet (dbid, file, field) is used to search the table for an existing entry. Duplicate entries are not permitted.

Identifiers must be specified in one of four combinations. The following combinations are supported:

DBID	File	Field	Comments
non-zero	non-zero	non-zero	Specific field translation
non-zero	non-zero	not specified	Specific exit or tables
non-zero	zero	not specified	Specific exit or tables
zero	zero	not specified	Specific exit or tables

Deleting a Translation Definition

The TRANSlat DELETE | REMove statement is used to delete a translation definition, in the following format:

```
TRANSlat DELETE | REMove  ID=(dbid,file), -
                          FIeld=field
```

```
ID=(dbid,file)
```

Specifies the database and file identifiers that this statement applies to. Both the database and file identifiers are numeric values with a range of 0 to 255. The database and file identifiers, along with the field identifier, are used to locate the active translation definition to be removed.

```
Field=field
```

Specifies the two byte Adabas short field name. If the translation definition to be deleted was added without the FIeld operand, the FIeld operand must also be omitted in the TRANSlat DELETE statement.

Displaying Translation Definitions

The TRANSlat DISplay | LISt command is used to display the currently active translation definitions. There are no operands for this command format, as follows:

```
TRANSlat DISplay | LISt
```

Creating Translation Tables

ASCII to EBCDIC Translation Modules

To create an ASCII to EBCDIC (TOEbcdic=) translation module, perform the following steps:

1. Create a source assembler member (see NETUA2E for an example) that defines a 256-byte translation table.

   For each ASCII character being translated, its hexadecimal value is used to index into the translation table, and the EBCDIC value at that location is used as the substitution value.
2. Assemble the source member.
3. Link the object code, specifying linkage editor parameters RENT,REUS to prevent loading multiple copies of the load module during normal execution.

For BS2000/OSD, the library WCP*vrs*.LIB contains two modules, with source code, for ASCII and EBCDIC code page translation. The module E2ASIE translates code page ISO 646, German reference version in code page ISO 8859-1, Latin 1 (for platforms such as Windows). The module A2ESIE translates a code page ISO 88591-1, Latin 1 in code page ISO 646, German reference version.

EBCDIC to ASCII Translation Modules

To create an EBCDIC to ASCII (TOAscii=) translation table, perform the following steps:

1. Create a source assembler member (see NETUE2A for an example) that defines a 256-byte translation table.

   For each EBCDIC character being translated, its hexadecimal value is used to index into the translation table, and the ASCII value at that location is used as the substitution value.
2. Assemble the source member.
3. Link the object code, specifying linkage editor parameters RENT,REUS to prevent loading multiple copies of the load module during normal execution.

In addition to creating explicit translation tables at the DBID, file, and field level, you can also alter the two default translation tables used for all ASCII/EBCDIC translation. For modification instructions, see the source member NETCAS for EBCDIC to ASCII translation; see the source member NETCEB for ASCII to EBCDIC translation.

Creating Translation Exits

A translation exit provides more detailed control over Entire Net-Work translation. Entire Net-Work calls the exit before and after translating a field for which the database, file, and field IDs match that of a translation definition that specifies the user exit.

The translation exit should be coded to be as efficient as possible. Software AG strongly discourages using supervisor services that may block the network task. The sample application program NETUTRNX may be used as an example.

Registers

When the exit is called, the following registers are passed to the application. The exit must save and restore Registers 0-14.

Register	Description
R1	Address of parameter list.
R13	Address of 18 word save area.
R14	Return address.
R15	Entry point address.

Parameter List

The address and length of the field can be obtained from the parameter list passed to the exit. The address of the parameter list is contained in Register 1. The DSECT of this parameter list can be found as the source member TDSPARMS. Refer to this source member for details.

User Work Fields

The four user work fields in the parameter list are the only fields that can be altered. If the exit requires more work areas, storage can be allocated to obtain a work area. The work area address can then be stored in one of the user work fields. The user work fields are cleared when Entire Net-Work is initialized and are not subsequently altered by Entire Net-Work.

The contents of the user work fields are shared between all active translation exits; care should be taken if more than one exit is concurrently defined. Altering any other fields within the parameter list may cause unpredictable results.

Return Codes

The translation exit may or may not translate a given field; for a superfield, it may translate only certain subfields. If the exit translates the field, it must not alter any storage areas before or after the field being processed.

Upon leaving the exit, a return code can be set in Register 15 to inform Entire Net-Work of the field's translation status. Setting the return code has no effect on Entire Net-Work's storage areas.

If the field is translated, the return code must be set to 4 before leaving the exit. A value of 4 instructs Entire Net-Work to not perform translation. The exit is not called again for the field.

If the field is not translated by the exit, the return code must be set to 0. A value of 0 instructs Entire Net-Work to translate the field.

If only certain subfields within the field are translated, the return code must be set to 8. A value of 8 instructs Entire Net-Work to not process the rest of the field.

Restrictions and Considerations

Format and Search Buffer Restrictions

The following restrictions apply to the Format and Search Buffers when being processed by the Entire Net-Work conversion routines:

- All fields must be specified with length and format. Otherwise, response code 229 (ADARSP229) is returned.
- No compressed records. Entire Net-Work translates uncompressed data only.
- Text literals are not translated. A text literal (an alphanumeric character string surrounded by quotes) is ignored by the translation process, and its value is not changed.
- No field ranges. Fields must be specifically identified.
- No undefined MU/PE counts; for example, AA1-N.

Platform Implementation Restrictions

The following restrictions are due to different platform implementations:

- The maximum length of unpacked numbers is 27 (ASCII), 31 (EBCDIC).
- The maximum length of packed numbers is 14 (ASCII), 15 (EBCDIC).
- User data provided in a mainframe link routine user exit is not sent to the ASCII-based machine.
- ET data (user data that is stored with an ET or CL command) is stored unconverted. When retrieved, it is filled up with blanks in the architecture code of the database if the record buffer length (RBL) is greater than the length of the data actually stored.
- ET data stored with command option 1='p' is not supported.

Adabas Restrictions

Certain Adabas commands cannot be used in communications between heterogeneous platforms:

- Soft coupling is not supported.
- Sort sequence for letters, numbers and special characters is different in ASCII (numbers first) and EBCDIC (letters first). This results in different sort sequences, and is relevant for the following Adabas commands:

Command	Function
L3/L6	Reads a file in logical sequential order based on the sequence of the values for a given descriptor.
S2	Selects a set of records that satisfy given search criteria.
S9	Sorts an ISN list provided by a user or created by a previous Sx Command.

- An L9 command to a mainframe (IBM. BS2000, or Facom) database without a Search Buffer / Value Buffer combination will result in a response code 57 (ADARSP057). Natural for OpenVMS will not deliver the required buffers unless the "starting from" option is specified on the histogram statement.
- The C3 command, which is used to store user restart data, is not supported on EBCDIC-based databases.
- The following Adabas commands are not supported on ASCII-based databases:

Command	Explanation
A4/E4	The A4 command updates the contents of one or more fields within a record. The E4 command deletes a record.
BT with F option	The BT command backs out all the updates that have been performed during the transaction currently being processed. The F option, which is used to exclude one or more files from the backout process, is not supported on ASCII-based databases.

Command	Explanation
L9 with I or F option	The L9 command returns each value contained in the inverted list for a given descriptor, and the number of records in which the value is contained. The I option returns the ISN for each value in the record buffer; the F option is used to specify the number of the file that contains the descriptor for which values are to be returned. The I option and the F option are not supported on ASCII-based databases.
LF with I option	The LF command returns the field definitions for a file. The I option, which causes all field information to be returned in Adabas internal format, is not supported on ASCII-based databases.
S5	The S5 command is used to select the records in one file that are coupled to a given record in another file.
Prefetch option	The prefetch option is not supported on ASCII-based databases. However, a new option 'M' has been defined, giving a functionality comparable to the prefetch option.

Adabas Considerations

When accessing a mainframe (IBM. BS2000, or Facom) database from OpenVMS or UNIX, an application may receive an Adabas response code 17 (ADARSP017 - file not found), even though the file does exist.

This happens because Natural on these platforms issues a restricted OPEN command only, allowing access to files specified in the record buffer on the OPEN command. To resolve the problem, modify the NATPARM module to provide access / update authority to the required files.

Translating Different Data Types

The translation process for the different data type fields is described as follows:

Field Type	Translation Process
Alpha	The Entire Net-Work task on the mainframe (IBM. BS2000, or Facom) node has two standard translation tables used by the main Entire Net-Work converter (NETCONV). These two tables handle the pure EBCDIC to ASCII and ASCII to EBCDIC translation. Note that Adabas long alphanumeric fields are not translated.
Binary	The translation consists of byte-flipping; that is, the first byte of the binary field is swapped with the last byte; the second byte is swapped with the next to the last byte, and so on, if the partner machine is a byte-swapped architecture.
Unpacked	All but the last byte of the field are translated using the described ASCII and EBCDIC translation tables. The last byte, which is the sign, is translated to its appropriate value.
Packed	The sign is translated to the appropriate value.

Field Type	Translation Process
Floating	The appropriate type of translation is performed; that is, IEEE, VAX to IBM, IBM to IEEE or VAX.

Hexadecimal ASCII and EBCDIC Translation

Tables translating hexadecimal values to ASCII and EBCDIC values can be found on the internet. Here is one such link: ***http://www.simotime.com/asc2ebc1.htm***.

4 IUCV Line Driver Overview

This chapter provides information for administrators responsible for configuring and running the Entire Net-Work IUCV line driver once Entire Net-Work is installed.

Notes:

1. IUCV is not supported on the z/OS platform.
2. Entire Net-Work 6.3 SP1 requires that zap WD631001 be installed in z/VSE environments. This zap is needed to successfully complete the NETSIP, which in turn is needed for IUCV to run properly.

The IUCV line driver documentation is organized as follows:

Defining Virtual Machines	Describes defining virtual machines for Entire Net-Work.
Authorizing Virtual Machines for IUCV	Describes how to authorize virtual machines for IUCV.
Guest Operating System Support	Describes how z/VSE guest operating systems can gain access to IUCV.
Installing the IUCV Line Driver in a z/VSE Environment	Describes how to install the IUCV line driver in a z/VSE environment.
Factors Affecting Compatibility and Performance	Describes factors that affect the compatibility and performance of the IUCV line driver.
Defining the IUCV Line Driver	Explains how to define the IUCV line driver
IUCV DRIVER Statement	Describes the syntax and parameters of the IUCV DRIVER statement.
IUCV LINK Statement	Describes the syntax and parameters of the IUCV LINK statement.
IUCV Operator Commands	Describes the operator commands available for the IUCV line driver.
IUCV Statistics	Describes the IUCV line driver statistics.

5 Defining Virtual Machines

In general, a separate virtual machine is required for each Entire Net-Work node.

Note: IUCV is not supported on the z/OS platform.

To insure that enough IUCV connections can be made for Entire Net-Work, the MAXCONN parameter of the OPTION statement may need to be coded in the CP directory. The MAXCONN value for each virtual machine in the network must be set high enough to:

- to allow one path for every LINK statement for the IUCV line driver; and
- to accommodate all other products using IUCV communications or services that may be installed.

The default value is 4. If no more than 4 virtual machines exist in the network and no other products using IUCV are installed, the MAXCONN parameter need not be coded.

The MAXCONN parameter can be added to the other parameter in the OPTION statement as follows:

```
OPTION parm1 parm2 ... MAXCONN 10 ... parmn
```

6 Authorizing Virtual Machines for IUCV

Any virtual machine requiring Entire Net-Work IUCV services must be authorized to use IUCV. This is accomplished by coding the IUCV statement in the CP directory to authorize either:

- the virtual machines where Entire Net-Work Communicators are installed (global); or
- specific connections (individual).

Coding the IUCV Statement

For both types of authorization, the PRIORITY parameter is optional. If specified, PRIORITY allows the Entire Net-Work IUCV line driver to issue priority messages when establishing sessions. Priority messages are not used for normal message traffic.

The MSGLIMIT parameter of the IUCV statement should not be coded. If required, it can be specified as a parameter on the Entire Net-Work LINK statement for the IUCV line driver.

For Global Authorization

Global authorization has the advantage that no CP directory modifications are required for existing entries when new virtual machines are added to the network.

Global authorization is defined in the CP directory for a virtual machine as follows:

```
IUCV ALLOW (PRIORITY)
IUCV ANY (PRIORITY)
```

The ALLOW option indicates that communication paths can be established from any other virtual machine to this virtual machine. No specific authorization is required for the virtual machine initiating the connection.

The ANY option indicates that this virtual machine can establish communication paths to any other virtual machine.

To Authorize Specific Connections

You may individually authorize each virtual machine using IUCV communications by coding an IUCV statement for each connect. For example:

```
IUCV vmid1 (PRIORITY)
IUCV vmid2 (PRIORITY)
```

7 Guest Operating System Support

The z/VSE guest operating system is supported.

Guest systems are generally designed to run in physical environments where IUCV is not supported. Therefore, they usually have no interface to IUCV.

To gain access to IUCV, a supervisor call routine (NETSVCD) and an IUCV support program (NETFLIH) are supplied with the IUCV line driver for the z/VSE guest operating system.

Note: The routine NETSVCD is not the Adabas SVC and must be installed under a different SVC number.

Because paging is required in z/VSE guest systems with 370 (VAE) and ESA mode supervisors, page-fixed buffers are required for the IUCV line driver.

Page-fixed buffers require the definition of real storage for the Entire Net-Work partition under z/VSE. The BUFFERS parameter is described in the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*. For information about the definition of real storage, see the section *z/VSE Environments* in the *Entire Net-Work Installation Guide*.

8 Installing the IUCV Line Driver in a z/VSE Environment

This section describes the installation procedure for the Entire Net-Work SVC and the IUCV support module. For compatibility reasons, the Entire Net-Work SVC is installed in all z/VSE systems whether they support IUCV communications or not; the IUCV support module is installed only where necessary.

IUCV Support Module

The IUCV support module is necessary to intercept IUCV external interruptions in environments where they are not recognized by the operating system, i.e., 370 or ESA paging environments.

In order to maintain a queue of pending IUCV requests, the IUCV support module allocates an area of page-fixed SVA GETVIS storage. The size of this SVA GETVIS area can be calculated using the following formula:

```
INTQ=QSIZE*LNIQ
```

where:

QSIZE	is the value of the QSIZE parameter specified on the z/VSE EXEC statement for NETSIP. If no QSIZE parameter is specified, this value defaults to the value of the MAXCONN parameter of the OPTION statement in the CP directory times the MSGLIMIT parameter value of the IUCV statement in the CP directory. If MAXCONN and MSGLIMIT have not been specified, their values default to 4 and 10, respectively.
LNIQ	has the constant value of 48.

Round the value for INTQ up to the next page boundary. The total page-fixed SVA requirement is then:

```
sva=ROUND(INTQ)+12288
```

Note: If NETSVCD is linked with AMODE(31), then this storage is acquired above the 16-MB line.

Once installed, the IUCV support module remains anchored in the system. It remains dormant when Entire Net-Work is not active.

Using NETSIP to Install Under z/VSE

The NETSIP program is used to install the Entire Net-Work SVC and the IUCV support module. It can be used to install

- during the IPL procedure; or
- dynamically, without the need for an IPL.

NETSIP requires the use of a B transient, $$BNETV5, which must be in a library in the library search chain.

The following (or equivalent) JCS should be inserted in the ASI BG JCS procedure immediately before the START of the POWER partition so that the Entire Net-Work SVC and the IUCV support module are installed during the IPL procedure.

Note: To install the Entire Net-Work SVC and IUCV support module dynamically without having to do an IPL, execute the following JCS in BG instead.

```
// DLBL SAGLIB,'NETWORK.Vvrs.LIBRARY'
// EXTENT SYS010,vvvvvv
// ASSGN SYS010,DISK,VOL=vvvvvv,SHR
// LIBDEF PHASE,SEARCH=SAGLIB.WCPvrs
SET SDL
NETSVCD,SVA
/*
// OPTION SYSPARM='svc'
// UPSI 00000010
// EXEC NETSIP,PARM='QSIZE=nnnnn,RESID=xxxxxxxx'
```

where:

Substitution	Description
vvvvvv	Specifies volume for the Entire Net-Work 5 library
vrs	Entire Net-Work version level
svc	SVC number to be used as the Entire Net-Work SVC

At execution time, the NETSIP program determines whether a printer is assigned to system logical unit SYSLST. If no printer is assigned, messages are written to SYSLOG instead of SYSLST.

More information about the NETSIP program is provided in the following paragraphs.

Choosing the Entire Net-Work SVC Number

Entire Net-Work requires an entry in the z/VSE SVC Table. The number you choose for the Entire Net-Work SVC must be currently unused.

To ensure that the number you choose is unused, run the program NETSIP with the UPSI byte option "S".

This produces a list of the used and unused SVCs in the z/VSE SVCTAB. See the section *z/VSE UPSI Byte Options*.

Alternatively, you may use the following procedure to find an unused SVC:

1. Obtain a listing of the supervisor being used.
2. Use the assembler cross-reference to locate the label SVCTAB.

   This is the beginning of the z/VSE SVC Table. The table contains a four-byte entry for each SVC, starting with SVC 0.
3. Locate an entry higher than 31 that has a value of "ERR21".

   This value indicates an unused SVC Table entry.
4. Use the entry number as input to NETSIP.

z/VSE UPSI Byte Options

The z/VSE UPSI byte options are used to select NETSIP options. The UPSI byte format is:

```
// UPSI dsxxxxxc
```

where:

d	If 1, NETSIP uses PDUMP to dump the Entire Net-Work SVC and data areas. This option should be used only after the SVC is installed. The dump option is for debugging purposes only; do not use it unless instructed to do so by your Software AG technical support representative.
s	If 1, NETSIP lists the z/VSE SVC Table and indicates whether each SVC is used or unused. No SVC number is required when using this function. This feature is used to determine which SVCs can be used for Adabas and Entire Net-Work.
c	If 1, NETSIP will not check whether the Entire Net-Work SVC number specified by SYSPARM is valid.

NETSIP Execution Parameters

The NETSIP execution statement has the following syntax:

```
// EXEC NETSIP,PARM='QSIZE=qsize,RESID=resid'
```

where:

Substitution	Description
qsize	The number of the interrupt queue elements to be allocated in SVA GETVIS storage. The default value is the setting of the MAXCONN parameter of the OPTION in the CP directory times the MSGLIMIT parameter value of the IUCV statement in the CP directory. This parameter is optional.
resid	The IUCV resource ID for the Entire Net-Work IUCV line driver. The resource ID is a logical name for an IUCV service. The standard resource ID for Entire Net-Work is SAGNETWK. This parameter should be issued only when it is necessary to run two separate versions of Entire Net-Work with IUCV line drivers concurrently. An example for this could be a test and a production network. This parameter is optional; if specified, RESID should be the same on all nodes

Note: If neither QSIZE nor RESID is specified, the PARM option on the z/VSE EXEC statement must not be coded.

z/VSE 31-Bit Support

The information in this section applies only to users of z/VSE 1.3 or above running in ESA mode.

When the NETSIP program recognizes a system running z/VSE Version 1.3 or above, it attempts to place the interrupt queue and path table in the SVA above the 16-megabyte line if such storage is available for 370 or ESA paging environments.

9 Factors Affecting Compatibility and Performance

The IUCV line driver supports the IUCV SEND/REPLY and the SEND 1WAY protocols, allowing the Entire Net-Work administrator to select the environments most appropriate for the node.

- The SEND/REPLY protocol is most appropriate for environments with heavy two-way traffic. In most cases, you will find the SEND/REPLY protocols to be advantageous.
- The SEND 1WAY protocol may be more appropriate for light message traffic, or where returning messages are routed through another link.

  The Entire Net-Work IUCV line driver uses established and documented interfaces to ensure mutual use of IUCV services by different applications.
- Where supported in guest operating systems, Entire Net-Work uses the VCNA macro VSIUCV.
- In z/VSE environments with 370 mode supervisors, Entire Net-Work IUCV adheres to the principles of the Cross Product Interface (CPI). For more information about CPI, contact your Software AG technical support representative.

Products that do not support the interfaces and macros mentioned above cannot be used simultaneously with Entire Net-Work IUCV. If used, unpredictable results can occur. Such results may include operating system-disabled waits, abnormal termination, or indefinite wait situations in user programs, Entire Net-Work Communicators, or user targets.

10 Defining the IUCV Line Driver

The operating characteristics for a single Entire Net-Work line driver are defined using the Entire Net-Work IUCV DRIVER statement and one or more LINK statements.

The following is an example of a guest operating system definition for an external VTAM link and a local IUCV group link:

```
NODE          VSEPROD...
DRIVER            IUCV     ...
DRIVER            VTAM     ...
LINK TOITM        IUCV     USERID=VSETEST,...
LINK TOEXTERN VTAMAPPLID=VSEPROD,....
```

11 IUCV DRIVER Statement

The DRIVER statement must follow the related NODE statement, which in turn is followed by the related **LINK statements**. For information about the NODE statement, see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.

DRIVER Statement Format

The following is the format of the Entire Net-Work IUCV DRIVER statement. The access method name IUCV instructs Entire Net-Work to load the line driver module NETIUCV.

```
DRIVER  IUCV    [ACCEPTUI = { Y | N } ]
                [DRVCHAR = { character | $ } ]
                [FORCE = { Y | N } ]
                [RESID = resource-id ]
                [RESTART = (interval, retries} ]
                [STATINT = { stat-interval | 3600 } ]
                [SVCNO = svc-number ]
                [TRACE = { Y | N } ]
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the DRIVER Statement Parameters

The DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. See the ALTER operator command in the section *IUCV Operator Commands* .

Some parameters can be modified when the line driver is open or closed. Others can be modified only when the line driver is closed. The open/closed requirement for each parameter is included in the following descriptions.

DRIVER Statement Parameters

This section describes all of the parameters that can be used for the IUCV DRIVER statement.

For more information about syntax conventions and rules used in this section, read ***Conventions***.

ACCEPTUI Parameter

```
ACCEPTUI = { Y | N }
```

This optional parameter determines whether the line driver will accept connections from systems that have not been previously defined with LINK statements. The ACCEPTUI parameter can be modified when the line driver is open or closed.

Valid values are "Y" (Yes) or "N" (No).

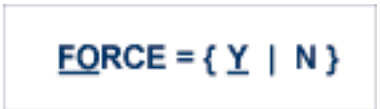
- If "Y" is specified, Entire Net-Work will accept connection requests from an undefined system and the required control blocks are built dynamically. Normal "handshaking" procedures with the new connections are performed.
- If "N" is specified, Entire Net-Work will reject incoming requests from unknown source nodes.

DRVCHAR Parameter

```
DRVCHAR = { char | $ }
```

This optional parameter specifies the special character used to designate that an operator command should be directed to this line driver rather than to a specific link. The DRVCHAR parameter can be modified only when the line driver is closed.

FORCE Parameter



This optional parameter causes the driver to issue an IUCV RETRIEVE BUFFER command if the initialization to IUCV services detects an IUCV user program active in the related virtual machine that is not using an IUCV macro interface. The FORCE parameter can be modified only when the line driver is closed.

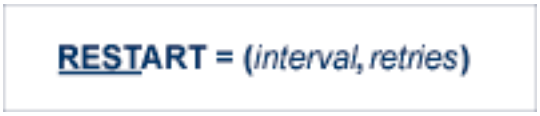
Specifying FORCE=Y will terminate IUCV communications for all other products in the virtual machine using IUCV communications. For more information about mutual use of IUCV, read *Factors Affecting Compatibility and Performance*.

RESID Parameter

This parameter is the IUCV resource ID for the Entire Net-Work IUCV line driver. The resource ID is a logical name for an IUCV service. The standard resource ID for Entire Net-Work is SAG-NETWK. This parameter should be issued only when it is necessary to run two separate versions of Entire Net-Work with IUCV line drivers concurrently. An example for this could be a test and a production network. If specified, the resource ID should be the same on all nodes. The RESID parameter can be modified only when the line driver is closed. Its value cannot be modified using the ALTER operator command.

If RESID is specified on the DRIVER statement, the same value for the RESID parameter must be specified on the LINK statements of the other nodes that are to be connected to this node. In addition to specifying the RESID parameter in the DRIVER statement, the RESID parameter of NETSIP or NETSIR must be specified on z/VSE nodes with non-z/VM mode supervisors.

RESTART Parameter



This optional parameter specifies the retry interval in seconds (*interval*) and the number of retries (*retries*) that Entire Net-Work will attempt to reopen the access method with the API after a shutdown due to a failure. The RESTART parameter can be modified when the line driver is open or closed.

If RESTART is not specified, or *interval* is specified as zero, no retry is attempted. By specifying (*retries*) as zero, an infinite number of retries can be requested.

If RESTART is specified on the IUCV DRIVER statement, a corresponding RESTART parameter value should be specified on each IUCV LINK statement to control restart attempts on the individual link.

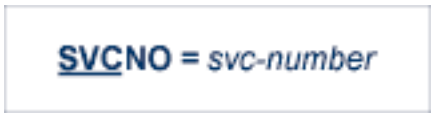
The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

STATINT Parameter

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. The STATINT parameter can be modified when the line driver is open or closed.

There is no maximum value. A value of 0 is reset to 3600. The default is 3600.

SVCNO Parameter



This parameter, which applies to z/VSE guest operating systems only, specifies the Entire Net-Work SVC number; that is, the number under which NETSVC has been installed, not the number of the Adabas SVC. The SVCNO parameter can be modified only when the line driver is closed.

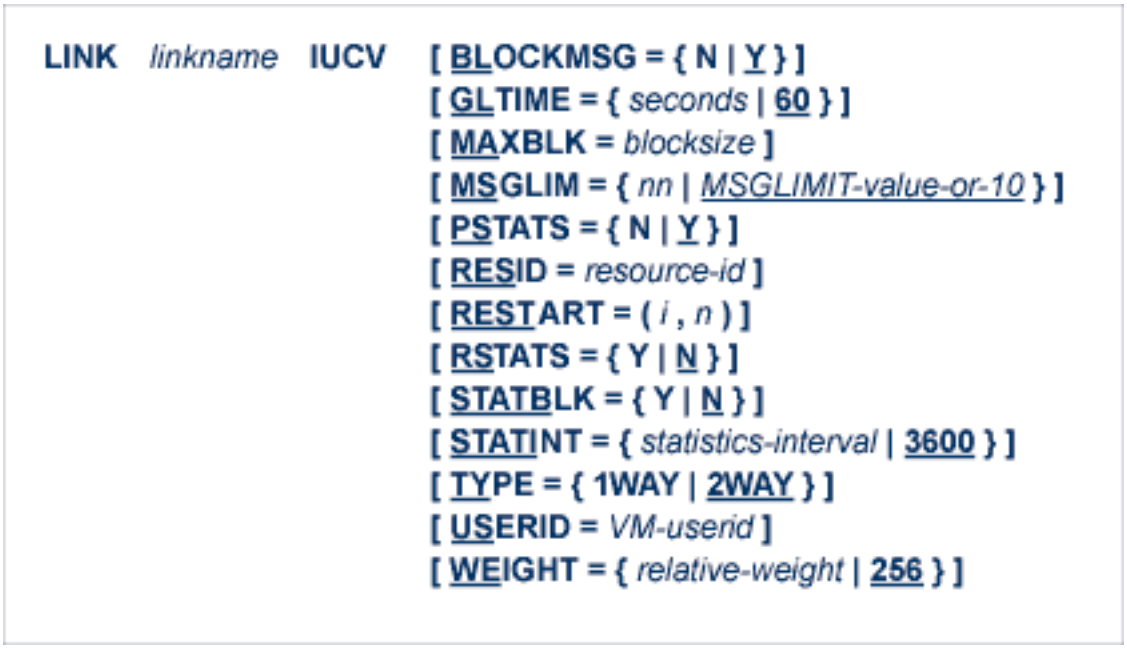
TRACE Parameter

This parameter indicates whether tracing for this line driver should be active (Y) or not (N). When tracing is activated, trace information is placed in the trace table. The default is N (no). The TRACE parameter can be modified when the line driver is open or closed.

This is equivalent to specifying `TRACE=linedriver-code` or `TRON=linedriver-code` in the NODE statement (for example, `TRACE=CTCA`).

12 IUCV LINK Statement

Each IUCV link must be defined with a LINK statement. LINK statements must follow the related DRIVER statement. The following is an example of the IUCV LINK statement:



For more information about syntax conventions and rules used in this chapter, read *Conventions*.

Modifying the LINK Statement Parameters

The LINK statement parameters are read from a sequential file during system startup. Some of the parameters can be modified after startup using the ALTER operator command. See the ALTER operator command in the section *IUCV Operator Commands* . The parameters that can be modified using ALTER are identified in the following descriptions.

LINK Statement Parameters

The USERID parameter is required. All other parameters are optional and have default values as described. The underlined portion of the parameter is the minimum abbreviation.

```
BLOCKMSG={Y | N}
```

The default BLOCKMSG=Y (Yes) causes messages for this link to be blocked for transmission. If N (No) is specified, no blocking occurs.

```
GLTIME={secs | 60}
```

This optional parameter specifies the maximum elapsed time in timer units (one unit = 1.05 seconds) allowed for a request over a group link to be completed. Accepted values range from 1 to 2147483647 (approximately 68 years; effectively, no timeout occurs). The default is 60 (approximately one minute).

```
MAXBLK=block-size
```

This optional parameter specifies the maximum transmission block size for the link. The value block-size can be up to 9 decimal digits in length. There is no default size; if no value is specified, no limit is set for the size of a transmission block. For group links between guest operating systems and other systems or users/targets, this value is replaced by the block size appropriate for the guest operating system, and must be a multiple of 2048 bytes. For direct links between virtual machines, both ends must specify the same block size, or errors may occur. If no value is specified, and the environment limits the transmission block size to a page, the parameter will be set automatically to the proper value. Specifying a value smaller than needed will influence the performance negatively.

```
MSGLIM=n
```

This optional parameter specifies the maximum number of outstanding IUCV messages allowed on this link. If the MSGLIMIT option was specified in the CP directory, this parameter can be used to decrease the value in the directory, but the directory value will override any larger value specified in the MSGLIM parameter. To allow for maximum flexibility, it is recommended that the MSGLIMIT option not be specified in the CP directory.

The value for the MSGLIM parameter can be 1-255. If not specified, MSGLIM is set to the value of the MSGLIMIT option. If no MSGLIMIT option is specified, MSGLIM is set to ten (10).

```
PSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are printed to DDPRINT when the statistics interval expires. The default value is PSTATS=Y. This parameter does not affect the STATS operator command. The PSTATS parameter can be modified with the ALTER operator command; the link may be open or closed.

```
RESID=resource-id
```

This parameter is the IUCV resource ID for the Entire Net-Work IUCV line driver. The resource ID is a logical name for an IUCV service. The standard resource ID for Entire Net-Work is SAGNETWK. This parameter should be issued only when it is necessary to run two separate versions of Entire Net-Work with IUCV line drivers concurrently. An example for this could be a test and a production network. If specified, the resource ID should be the same on all nodes.

The RESID specified on the LINK statement must have the same value as the RESID parameter specified on the DRIVER statements of other nodes that are to be connected to this node.

```
RESTART=retry interval and number retries (i,n)
```

RESTART specifies the retry interval in seconds (i) and the number of retries (n) to be used if you want Entire Net-Work to attempt to reconnect the link following a shutdown due to a failure. If RESTART is not specified or (i) is specified as zero, no retry is attempted. By specifying (n) as zero, an infinite number of retries can be requested.

The TIMER parameter on the NODE statement affects the RESTART parameter. (See the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead. The RESTART parameter can be modified with the ALTER operator command; the link may be open or closed.

If a RESTART interval and count are also specified for the IUCV DRIVER statement, it is recommended that the LINK restart interval and retry count be coordinated with those values to ensure that a link restart follows an IUCV driver restart, if it occurs.

```
RSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are automatically reset when the statistics interval expires. The default value is N. The RSTATS parameter can be modified with the ALTER operator command; the link may be open or closed.

```
STATBLK={N | Y}
```

This optional parameter specifies whether or not blocking statistics for the link are created. For this parameter to take effect, the BLOCKMSG parameter must also specify Y (Yes). The STATBLK parameter can be modified with the ALTER operator command; the link may be open or closed.

Blocking statistics are kept for

- the number of messages blocked into transmission blocks;
- the number of transmission blocks created;
- the number of messages rejected for lack of block space ("thrown back"); and
- the maximum number of messages "thrown back" for any given block.

```
STATINT={statistics interval | 3600}
```

This optional parameter specifies the amount of time, in seconds, before statistics are automatically printed or reset. The default is 3600. A value of 0 is reset to 3600. The STATINT parameter can be modified with the ALTER operator command; the link may be open or closed.

```
TYPE={1WAY | 2WAY}
```

This optional parameter specifies whether the IUCV protocol SEND/REPLY (SEND 2WAY) or the SEND 1WAY is to be used. The default is SEND/REPLY (2WAY). For more information about the appropriate protocol, see the section ***Factors Affecting Compatibility and Performance***.

```
USERID=vm-user-id
```

This parameter specifies the z/VM user ID (VMID) of the target virtual machine. This parameter must be specified; there is no default user ID.

```
WEIGHT={n | 256}
```

This parameter allows a performance- or cost-based value to be assigned to each link in order to provide Entire Net-Work with information on which to base path selection if more than one path to a destination is available. A higher value should be used for faster, more expensive links, and a lower value for slower links. The default value is WEIGHT=256.

13 IUCV Operator Commands

Entire Net-Work's IUCV line driver has the ability to process operator commands that are directed to a specific link or directly to the driver.

The IUCV line driver operator commands have the following format:

The following table describes this syntax.

Syntax Representation	Description
IUCV	Informs Entire Net-Work that the command is destined for the IUCV line driver.
target	A value that informs IUCV what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the DRVCHAR value ("$" is the default) if the target is the driver itself (see the DRVCHAR parameter on the ***IUCV DRIVER Statement***). ■ Specify the link name if the target is a specific link.
cmd	The operator command to be carried out.

Examples:

The following are examples of IUCV operator commands:

```
NETWRK IUCV * SHOW
```

```
NETWRK IUCV $ STATS
```

```
NETWRK IUCV link3 TRACE
```

Driver Commands

The Entire Net-Work IUCV line driver supports the commands listed in the following table when the target is the driver. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER driver configuration parameters	Dynamically changes the driver configuration. The ALTER command is followed by the driver configuration parameters to be altered. The driver configuration parameters are the same as those specified in the DRIVER statement. For example: `IUCV $ ALTER DRVCHAR=%` Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.
RESET	Resets all statistics for the driver. Statistics are printed only if the STATS command precedes the RESET command.
SHOW	Displays the current configuration of the driver. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all control blocks specific to the link to be snapped (printed in hexadecimal). Driver specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Prints statistics immediately.Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example: `STATS RESET`
STATUS	Displays the current status of the driver and the number of messages received and sent.
TRACE	Causes the IUCV driver to format and print the driver-specific trace table. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.

Note: When the driver is closed, it does not recognize the commands STATS or RESET.

Link Commands

The Entire Net-Work IUCV line driver supports the commands listed in the following table when the target is a link or all links. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER link configuration parameters	Dynamically changes the link configuration. The ALTER command is followed by the link configuration parameters to be altered. The link configuration parameters are the same as those specified on the LINK statement. For example: `IUCV GCS02 ALTER STATINT=1800`

Command	Action
	Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.
RESET	Resets all statistics for the link. Statistics are printed only if the STATS command precedes the RESET command.
SHOW	Displays the current configuration of the link. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all link specific control blocks and the link specific trace table to be snapped (printed in hexadecimal). Driver specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. This command has no effect on the next automatic printing of statistics. To print and reset statistics, specify RESET immediately after the STATS command. For example:
STATUS	Displays the current status of the link as well as a count of messages received and sent.
TRACE	Causes the link specific trace table to be formatted and printed. The trace table is formatted and printed in hexadecimal automatically when the SNAP command is processed.

14 IUCV Statistics

The Entire Net-Work IUCV driver issues API calls to communicate with CP. To help tune each link and the driver itself, the IUCV driver provides the statistics shown below. The statistics for a link and the driver are identical, with the exception of the title line (a).

```
    +-------------------------------------------------------------------+
(a) + Link linkname   State state   Period   hh:mm:ss (sssssss.sss Secs) +
    + -----------  ---Bytes----  --Messages--  ---Buffers--  ----------- +
(b) + Writes       ww,www,wwwu   mmm,mmm,mmm   bbb,bbb,bbb   Total       +
(c) +              ww,www,wwwu   mmm,mmm,mmm   bbb,bbb,bbb   Per Second  +
(d) + Reads        ww,www,wwwu   mmm,mmm,mmm   bbb,bbb,bbb   Total       +
(e) +              ww,www,wwwu   mmm,mmm,mmm   bbb,bbb,bbb   Per Second  +
    + -----------  ------------  ------------  ------------  ----------- +
```

This multiple line display is produced when the command IUCV xxxx STATS is issued, or when the automatic statistics interval expires and PSTATS=Y is specified in the LINK statement. Values are displayed and updated asynchronously; therefore, the totals displayed may not always be accurate. The contents are as follows:

Line	Shows the . . .
a	name of the link or driver, its state (e.g., active or inactive), and the length of time since statistics were last reset or the link was last connected. Length of time is displayed in *hours:minutes:seconds* format and in *seconds:milliseconds*.
b	cumulative number of bytes and messages written, and the cumulative number of API calls.
c	average number of bytes and messages written, and the average number of read API calls per second.
d	cumulative number of bytes and messages read, and the cumulative number of read API calls.
e	average number of bytes and messages read, and the average number of read API calls per second.

15 VTAM Line Driver Overview

The Entire Net-Work VTAM line driver provides transport connectivity between client and server processes on homogeneous or heterogeneous platforms using the SNA LU0, LU6.0, or LU6.2 protocol.

In order to achieve communication among VTAM, the Entire Net-Work VTAM line driver, and the partner nodes in the network, these three elements must be defined to each other. This documentation provides detail explanations of each of these requirements. Specifically:

- *Preparing the VTAM Environment* describes the steps involved in defining the Entire Net-Work VTAM line driver to VTAM.
- *VTAM DRIVER Statement* provides detailed information about the VTAM DRIVER statement parameters, which allow you to define the ACF/VTAM operating characteristics and requirements to the VTAM line driver.
- *VTAM LINK Statement* provides detailed information about the VTAM LINK statement parameters, which allow you to define the partners with which the node will communicate.

This chapter provides information for administrators responsible for configuring and running the Entire Net-Work VTAM line driver once Entire Net-Work is installed.

The VTAM line driver documentation is organized as follows:

VTAM Line Driver Installation Prerequisites	Describes the prerequisites for installing and using the VTAM line driver.
Preparing the VTAM Environment	Describes the steps involved to prepare your VTAM environment for the Entire Net-Work VTAM line driver.
VTAM DRIVER Statement	Describes the syntax and parameters of the VTAM DRIVER statement.
VTAM LINK Statement	Describes the syntax and parameters of the VTAM LINK statement.

VTAM Operator Commands	Describes the operator commands available for the VTAM line driver.
VTAM Line Driver Model Links	Describes the model link support offered by the VTAM line driver.
VTAM Line Driver User Exits	Describes the user exit support provided with the VTAM line driver.

16 VTAM Line Driver Installation Prerequisites

This chapter describes the VTAM Line Driver installation prerequisites.

Supported Operating Systems

The Entire Net-Work VTAM line driver supports the z/OS and z/VSE operating systems. For the currently supported versions, read *Supported Operating System Platforms* in the *Entire Net-Work Installation Guide*.

VTAM Requirements

The following level of VTAM must be installed:

- ACF/VTAM Version 2 or above;
- FACOM requires VTAM-G Version 10 or above.

Compatibility Considerations

The Entire Net-Work VTAM line driver provides transport connectivity to Entire Net-Work on the following systems:

- Another IBM mainframe running the VTAM line driver.
- An OpenVMS system running the SNA protocol handler.
- A BS2000/OSD machine running the DCAM line driver.
- An IBM RS/6000 or HP/UNIX system running the LU6.2 protocol handler.
- A Microsoft Windows 95, Windows 98, Windows NT 4.0, or Windows 2000 system running Novell Netware for SAA, Microsoft SNA Server 4.0, or IBM Secure Way Communications Server.

17 Preparing the VTAM Environment

Note: Some of the steps in this chapter may not be necessary, depending on your existing VTAM network.

To prepare your VTAM environment for the Entire Net-Work VTAM line driver:

1 Follow the Entire Net-Work installation instructions for your operating system.

2 Define the Entire Net-Work VTAM line driver to VTAM by coding a VTAM APPL macro. You may add the VTAM APPL macro to an existing member in the VTAMLST library, or place it in a new member. Refer to the *IBM VTAM Installation and Resource Definition Manual* and the *VTAM Customization Manual* for details on coding APPL definitions.

The Entire Net-Work VTAM line driver requires that the following parameters be coded on its APPL macro:

```
AUTH=(ACQ,NVPACE)
ENCR=NONE
PARSESS=YES
SONSCIP=YES
```

For improved performance, it is recommended that the VTAM APPL macro also contain the following parameters:

```
SRBEXIT=YES  (z/OS only)
```

3 If your Entire Net-Work will communicate with partners in other VTAM domains, you may need to code the appropriate multisystem networking definitions to allow cross-domain communications. The term domain refers to a VTAM-started task (or virtual machine or partition) and all resources defined to it.

Cross-domain communication has the following requirements:

- Each domain requires a CDRM statement identifying each partner domain with which it will communicate.
- Each domain requires a PATH statement that identifies each partner domain by its SUB-AREA number and the transmission group representing the first segment of the path. Also, if the path includes front-end processors, then the front-end processor network control program requires a PATH statement for each partner domain.
- Each domain requires a CDRSC statement that identifies the other domain network partner and its CDRM. If communication is always to be initiated by one partner, this CDRSC statement may be optional in the domain of the other partner.

Refer to the IBM *VTAM Installation and Resource Definition Manual* and the *VTAM Customization Manual* for assistance in coding these definitions.

4 Activate the VTAM definitions with the `V NET,ACTIVE,ID=` operator command.

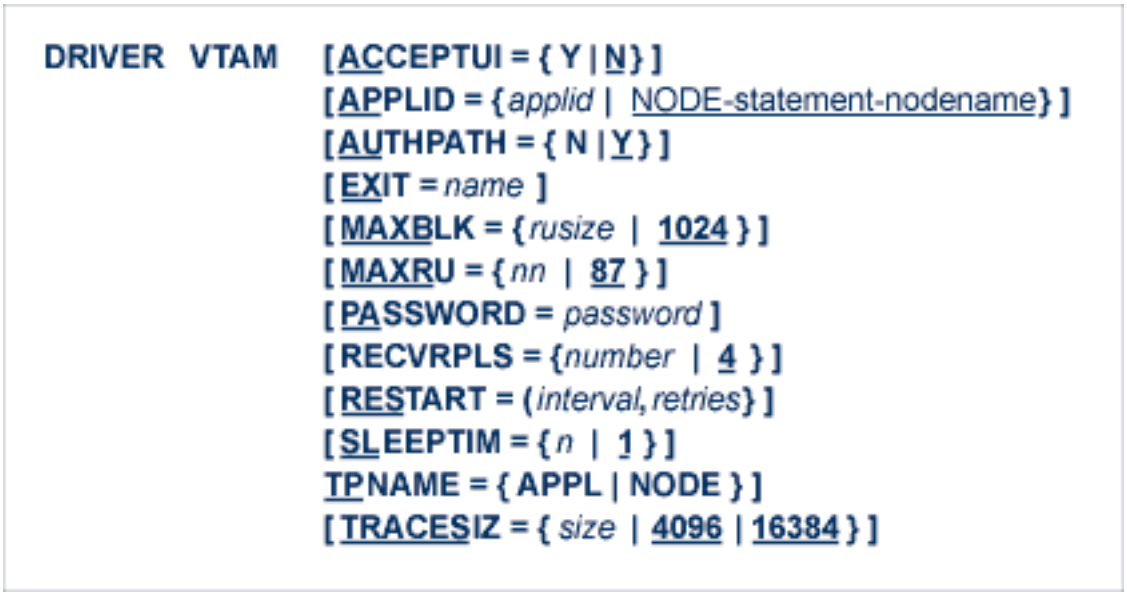
5 Verify the VTAM definitions with the `D NET,ID=` operator command.

18 VTAM DRIVER Statement

The VTAM DRIVER statement defines the ACF/VTAM operating characteristics and requirements to the Entire Net-Work VTAM line driver. The DRIVER statement must follow an Entire Net-Work NODE statement, but may be preceded by one or more non-VTAM DRIVER statements and their related LINK statements. Only one VTAM DRIVER statement can be specified for a node.

DRIVER Statement Format

The VTAM DRIVER statement has the following format:



For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the DRIVER Statement Parameters

The DRIVER statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the line driver is open or closed. Others can be modified only when the line driver is closed. See ALTER and CLOSE in the section *VTAM Operator Commands*. The open/closed requirement for each parameter is included in its description.

DRIVER Statement Parameters

This section describes all of the parameters that can be used for the VTAM DRIVER statement.

For more information about syntax conventions and rules used in this section, read ***Conventions***.

ACCEPTUI Parameter

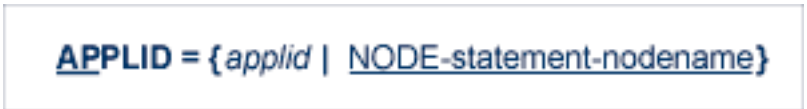
This optional parameter determines whether the line driver will accept connections from systems that have not been previously defined with LINK statements. The ACCEPTUI parameter can be modified when the line driver is open or closed.

Valid values are "Y" (Yes) or "N" (No).

- If "Y" is specified, Entire Net-Work will accept connection requests from an undefined system and the required control blocks are built dynamically. Normal "handshaking" procedures with the new connections are performed.

  If you are using the TCP/IP or VTAM line drivers, user exits can be defined to provide security for incoming connections. For more information, see the section *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.
- If "N" is specified, Entire Net-Work will reject incoming requests from unknown source nodes.

APPLID Parameter

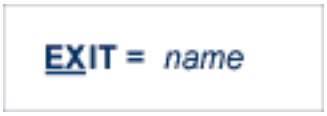


This parameter provides the 1 - 8 character application name by which the Entire Net-Work VTAM line driver identifies itself to VTAM. The name specified must match the value specified for the ACBNAME parameter of the APPL macro in the VTAMLST library. If this parameter is omitted, the node name specified in the NODE statement is used. The APPLID parameter can be modified only when the line driver is closed.

AUTHPATH Parameter

If "Y" (Yes) is specified for this parameter, all VTAM requests are issued using VTAM Authorized Path. This feature greatly reduces VTAM overhead for most requests. In GCS, the virtual machine must be authorized. In z/OS, all STEPLIB data sets must be authorized. Authorized Path is not available for z/VSE systems. The AUTHPATH parameter can be modified only when the line driver is closed.

EXIT Parameter



This optional parameter specifies the name of a user exit. The default is to run no user exit. For more information, read *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

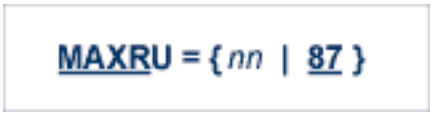
If EXIT is coded but the load module cannot be loaded, execution continues as if no exit were specified. The EXIT parameter can be modified only when the line driver is closed.

MAXBLK Parameter

This parameter specifies the size, in bytes, of the internal buffer used to receive data from VTAM. This is the size of the largest request unit that VTAM will deliver to this node. This size can also be specified with the MAXRU parameter. If values are specified for both parameters, the larger value is used. The MAXBLK parameter can be modified only when the line driver is closed.

Specify a decimal value from the MAXRU table shown below. Optionally, a value followed by a multiplier of either "M" (x 1048576, or 1 MB) or "K" (x 1024, or 1 KB) can be specified. The value can be 128 - 491520. A value less than 128 is reset to 128. A value greater than 128 but not matching one of those in the MAXRU table is reduced to the nearest table value. The default value is 1024.

MAXRU Parameter



This parameter specifies the maximum size, in bytes, of the internal buffer used to receive data from VTAM. This is the size of the largest request unit that VTAM will deliver to this node. This size can also be specified with the MAXBLK parameter. If values are specified for both parameters, the larger value is used. The MAXRU parameter can be modified only when the line driver is closed.

The MAXRU specification is determined by the following equation:

```
bytes = 1st-hex-digit * (2-raised-to-the-2nd-hex-digit-power)
```

For example:

```
87 = 8 * (2 to the 7th power) = 8 * 128 = 1024
```

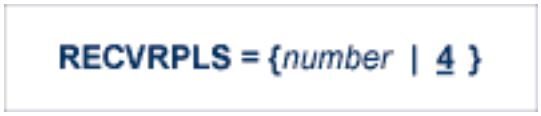
Specify a hexadecimal value from the MAXRU table shown below. The value can be 84 - FF. A value less than 84 is reset to 84. A value greater than 84 but not matching one of those in the MAXRU table is reduced to the nearest table value. The default is 87.

Hex Value of Bits 4-7	8	9	A	B	C	D	E	F
4	128	144	160	176	192	208	224	240
5	256	288	320	352	384	416	448	480
6	512	576	640	704	768	832	896	960
7	1024	1152	1280	1408	1536	1664	1792	1920
8	2048	2304	2560	2816	3072	3328	3584	3840
9	4096	4608	5120	5632	6144	6656	7168	7680
A	8192	9216	10240	11264	12288	13312	14336	15360
B	16384	18432	20480	22528	24576	26624	28672	30720
C	32768	36864	40960	45056	49152	53248	57344	61440
D	65536	73728	81920	90112	98304	106496	114688	122880
E	131072	147456	163840	180224	196608	212992	229376	245760
F	262144	294912	327680	360448	393216	425984	458752	491520

PASSWORD Parameter

This parameter specifies the 1 - 8 character password associated with this APPLID. The value specified for PASSWORD must be the same as that specified for the PRTCT parameter, if present, on the APPL macro in the VTAMLST library. The default is no password. The PASSWORD parameter can be modified only when the line driver is closed.

RECVRPLS Parameter



This parameter specifies the number of receive RPLs (Request Parameter Lists) to be kept active at all times. Receive RPLs are control blocks that Entire Net-Work uses to receive data from VTAM. The number of RPLs specified can be 4 - 32767 for the entire system.

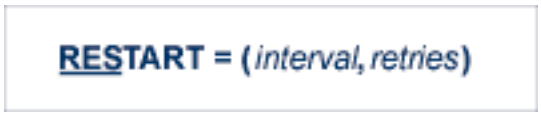
The maximum allowed may be limited by storage availability. At startup time, if the storage required to support the RECVRPLS requested is not available, Entire Net-Work resets the value to

RECVRPLS=4. A value less than 4 is reset to 4, which is also the default. The RECVRPLS parameter can be modified when the line driver is open or closed.

If a high number of receive overruns occurs (read about the STATS command in ***VTAM Operator Commands***), increase the RECVRPLS specification to provide more receive RPLs, or increase the receive buffer size specified with the MAXRU/MAXBLK parameters, or do both. Increasing the number of RPLs also increases the storage requirement, which is calculated as follows:

```
total bytes = (240 bytes + MAXBLK/MAXRU bytes) * RECVRPLS
```

RESTART Parameter



This optional parameter specifies the retry interval in seconds (*interval*) and the number of retries (*retries*) that Entire Net-Work will attempt to reopen the access method with the API after a shutdown due to a failure. The RESTART parameter can be modified when the line driver is open or closed.

If RESTART is not specified, or *interval* is specified as zero, no retry is attempted. By specifying (*retries*) as zero, an infinite number of retries can be requested.

If RESTART is specified on the VTAM DRIVER statement, a corresponding RESTART parameter should be specified on each VTAM LINK statement to control restart attempts on the individual link.

The RESTART parameter is particularly useful with the TCPI, VTAM, and TCPX line drivers when Entire Net-Work is started at IPL and communication with the API or VTAM ACB is unsuccessful because TCP/IP or VTAM is not yet fully initialized. Using this parameter, you can instruct Entire Net-Work to reopen the TCP/IP session or VTAM ACB, thereby giving TCP/IP or VTAM sufficient time to become active.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*.) The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead.

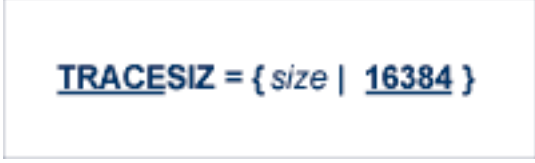
SLEEPTIM Parameter

This parameter specifies the number of seconds that the VTAM line driver will wait before attempting to reissue any VTAM macro after VTAM indicates that it requires more storage. The default is 1 (one second). The SLEEPTIM parameter can be modified when the line driver is open or closed.

TPNAME Parameter

This parameter specifies whether the applid-name or the node-name will be used as the transaction program name that is passed to the partner node.

TRACESIZ Parameter



This optional parameter specifies the size, in bytes, of the driver-specific trace table.

This parameter specifies the size of the internal wrap-around trace table for non-link-specific trace information. This trace can be formatted or printed with the operator command `VTAM # TRACE`.

Valid values can range from 16384 (the default) to 4194304. A value less than 16384 is reset to 16384; a value greater than 4194304 is reset to 4194304.

The TRACESIZ parameter can be modified when the line driver is open or closed.

19 VTAM LINK Statement

VTAM LINK statements describe the partners with which this node will communicate. If the ACCEPTUI parameter does not specify "Y" (Yes) on the VTAM DRIVER statement, a LINK statement is required for each partner.

LINK Statement Format

The VTAM LINK statement has the following format:

```
LINK  linkname  VTAM  [ ACQUIRE = { Y | YES | N | NO | ON | OFF } ]
                    [ APPLID = partner's-vtam-name ]
                    [ ASSOCLU = partner's-receive-lu ]
                    [ BLOCKMSG = { Y | YES | N | NO | ON | OFF } ]
                    [ COMPRMSG = { Y | YES | N | NO | ON | OFF } ]
                    [ CRYPT = { Y | YES | N | NO | ON | OFF } ]
                    [ DEFRESP = { Y | YES | N | NO | ON | OFF } ]
                    [ EXIT = name ]
                    [ LOGMODE = partner's-logmode-table-name ]
                    [ LUNAME = partner's-vtam-name ]
                    [ MAXBLK = { rusize | 1024 } ]
                    [ MAXRU = { maximum-buffer-size | 87 } ]
                    [ MINCMP = minimum-length-for-compression ]
                    [ MODEENT = { logmode-table-entry-name | SAGVTAM } ]
                    [ NETID = partner's-VTAM-network-ID ]
                    [ PSTATS = { N | Y } ]
                    [ RESTART = ( i , n ) ]
                    [ RSTATS = { Y | N } ]
                    [ SAF = { Y | L | N } ]
                    [ SNDTMOUT = { send-timeout | 10 } ]
                    [ STATBLK = { Y | N | YES | NO | ON | OFF } ]
                    [ STATCMP = { Y | N | YES | NO | ON | OFF } ]
                    [ STATINT = { statistics-interval | 3600 } ]
                    [ TRACESIZ = size ]
                    [ WEIGHT = { relative-weight | 256 } ]
                    [ ZEDC = { Y | N } ]
                    [ ZEDCLOG = { F | L | N } ]
```

For more information about syntax conventions and rules used in this section, read *Conventions*.

Modifying the LINK Statement Parameters

The LINK statement parameters are read from a sequential file during system startup, and can be modified after startup using the ALTER operator command. Some parameters can be modified when the link is open or closed. Others can be modified only when the link is closed. See ALTER and CLOSE in the section ***VTAM Operator Commands***. The open/closed requirement for each parameter is included in its description.

LINK Statement Parameters

In the following syntax-example diagrams, the underlined portion of the parameter name is the minimum abbreviation of the parameter.

linkname Parameter

```
linkname
```

This required parameter specifies the name by which this link is to be known; it is positional, and must be specified immediately after the LINK keyword and immediately before the driver name (VTAM); linkname must be unique on the node. All operator commands affecting the link must specify this name.

ACQUIRE Parameter

```
ACQUIRE={Y | N | YES | NO | ON | OFF}
```

This parameter is optional. ACQUIRE=Y (or "YES" or "ON") instructs Entire Net-Work to attempt to establish communications with this partner node as part of initialization. ACQUIRE=N (or "NO" or "OFF") instructs Entire Net-Work to wait for the partner to establish communications, or to wait for operator intervention. The default is N. The ACQUIRE parameter can be modified only when the link is closed.

This parameter is ignored for VAX and UNIX partners, which do not currently support session initiation from the IBM mainframe.

APPLID Parameter

```
APPLID=partner's-vtam-name
```

This optional parameter specifies this partner's VTAM application name. Specify the 1 - 8 character value from the ACBNAME parameter of the APPL macro that was used to define this partner to VTAM. If the ACBNAME parameter is omitted from the APPL macro, specify the 1 - 8 character label on the APPL macro. This parameter is synonymous with the LUNAME parameter. If both parameters are specified, the second one is used. If neither parameter is specified, or if the specified value contains wildcard characters, this link is considered a "model" link. (See the section ***Model Links***.)

ASSOCLU Parameter

```
ASSOCLU=partner's receive lu
```

This optional parameter specifies the VTAM name of the LU6.2 partner's Receive LU. The partner's Entire Net-Work simulates full duplex communication with the LU6.2 protocol by establishing two sessions with the VTAM line driver. The first session uses the LU specified by the LUNAME parameter in this LINK statement. The second session uses the LU specified by the ASSOCLU parameter. If you are using parallel sessions, omit the ASSOCLU parameter, or specify the same LU name for both parameters. Value may be from 1 - 8 characters in length. The ASSOCLU parameter can be modified only when the link is closed.

BLOCKMSG Parameter

```
BLOCKMSG={Y | N | YES | NO | ON | OFF}
```

Specify BLOCKMSG=Y to instruct Entire Net-Work to block as many messages as possible into one buffer for transmission to this partner. This parameter is reset to N for all partner types except IBM. The default is N. The BLOCKMSG parameter can be modified when the link is open or closed.

COMPRMSG Parameter

```
COMPRMSG={Y | N | YES | NO | ON | OFF}
```

Specify COMPRMSG=Y to instruct Entire Net-Work to attempt to compress duplicate characters out of the message before transmission to this partner. This parameter is reset to N for all partner types except IBM. The default is N. The COMPRMSG parameter can be modified when the link is open or closed.

CRYPT Parameter

```
CRYPT={Y | N | YES | NO | ON | OFF}
```

Specify CRYPT=Y to instruct Entire Net-Work to request that VTAM encrypt data before sending it to this partner and decrypt data received from this partner. This parameter is reset to N for all partner types except IBM. The default is N. The CRYPT parameter can be modified only when the link is closed.

DEFRESP Parameter

```
DEFRESP={Y | N | YES | NO | ON | OFF}
```

Specify DEFRESP=Y to instruct Entire Net-Work to request that this partner acknowledge receipt of each buffer by returning a definite response. The default is N. The DEFRESP parameter can be modified when the link is open or closed.

Note: Setting DEFRESP=Y may result in performance degradation.

EXIT Parameter

```
EXIT=name
```

This parameter specifies the name of a user exit. For more information, see the section *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

If a user exit is specified and the SAF parameter is set to Y or L, incoming requests are made available to the user exit before NETSAF is called.

If EXIT is coded but the load module cannot be loaded, execution continues as if no exit were specified. The EXIT parameter can be modified only when the link is closed.

LOGMODE Parameter

```
LOGMODE=partner's-logmode-table-name
```

This optional parameter specifies the name of the logmode table to be used for outbound connections to this partner and can be from 1 - 8 characters in length. This logmode table module is loaded at link initialization; therefore, the module must reside in one of the STEPLIB data sets. The LOGMODE parameter can be modified only when the link is closed.

LUNAME Parameter

```
LUNAME=partner's-vtam-name
```

This optional parameter specifies this partner's VTAM LU name. Specify the 1 - 8 character label from the LU statement used to define this partner to VTAM. This parameter is synonymous with the APPLID parameter. If both parameters are specified, the second is used. If neither parameter is specified, or if the specified value contains wildcard characters, this link is considered a "model" link. (See the section ***Model Links***.)

MAXBLK Parameter

```
MAXBLK={ru-size | 1024}
```

This parameter specifies the maximum size, in bytes, of the buffer used to send data to this partner. The maximum size can also be specified with the MAXRU parameter. If values are specified for both parameters, the larger value is used. The MAXBLK parameter can be modified only when the link is closed.

Specify a decimal value from the MAXRU table shown below. Optionally, a value followed by a multiplier of either "M" (x 1048576, or 1 MB) or "K" (x 1024, or 1 KB) can be specified. The value can be 128 - 491520. A value less than 128 is reset to 128. A value greater than 128 but not matching one of those in the MAXRU table is reduced to the nearest table value. The default value is 1024.

MAXRU Parameter

```
MAXRU= {maximum buffer size | 87}
```

This parameter specifies the maximum size, in bytes, of the buffer used to send data to this partner. The maximum buffer size can also be specified with the MAXBLK parameter. If values are specified for both parameters, the larger value is used.

The MAXRU specification is determined by the following equation:

```
bytes = 1st hex digit * (2 raised to the 2nd hex digit power)
```

For example:

```
87 = 8 * (2 to the 7th power) = 8 * 128 = 1024
```

Specify a hexadecimal value from the MAXRU table shown below. The value can be 84 - FF. A value less than 84 is reset to 84. A value greater than 84 but not matching one of those in the MAXRU table is reduced to the nearest table value. The default value is 87.

Hex Value of Bits 4-7	8	9	A	B	C	D	E	F
4	128	144	160	176	192	208	224	240
5	256	288	320	352	384	416	448	480
6	512	576	640	704	768	832	896	960
7	1024	1152	1280	1408	1536	1664	1792	1920
8	2048	2304	2560	2816	3072	3328	3584	3840
9	4096	4608	5120	5632	6144	6656	7168	7680
A	8192	9216	10240	11264	12288	13312	14336	15360
B	16384	18432	20480	22528	24576	26624	28672	30720
C	32768	36864	40960	45056	49152	53248	57344	61440
D	65536	73728	81920	90112	98304	106496	114688	122880
E	131072	147456	163840	180224	196608	212992	229376	245760
F	262144	294912	327680	360448	393216	425984	458752	491520

MINCMP Parameter

```
MINCMP=minimum-length-for-compression
```

This optional parameter specifies the minimum number of consecutive, identical bytes required for compression. Value can be 5 - 65535. A value less than 5 is reset to 5. This parameter is ignored if COMPRMSG=NO. The MINCMP parameter can be modified when the link is open or closed.

MODEENT Parameter

```
MODEENT= {SAGVTAM | logmode-table-entry-name}
```

This optional parameter specifies the name of the entry in the logmode table to be used for outbound sessions with this partner. The entry name can be 1-8 characters and must exist in the logmode table specified by the LOGMODE parameter for this LINK statement. The default value is SAGVTAM. The MODEENT parameter can be modified only when the link is closed.

NETID Parameter

```
NETID=partner's-VTAM-network-ID
```

This optional parameter specifies the partner's VTAM network ID. The values assigned to the NETID parameter and the APPLID parameter are combined to form the partner's fully qualified network name. The NETID parameter can be modified only when the link is closed.

PSTATS Parameter

```
PSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are printed to DDPRINT when the statistics interval expires. The default value is PSTATS=Y. This parameter does not affect the STATS operator command. The PSTATS= parameter can be modified when the link is open or closed.

RESTART Parameter

```
RESTART= (i,n)
```

This optional parameter specifies the retry interval in seconds (i) and the number of retries (n) that will be made to start the connection to the remote node. If RESTART is not specified, or (i) is specified as zero, no retry is attempted. By specifying (n) as zero, an infinite number of retries can be requested.

The TIMER parameter on the NODE statement affects the RESTART parameter (see the section *Entire Net-Work NODE Statement* in the *Entire Net-Work Reference Guide*section. The retry interval should not be less than the TIMER parameter, and should be a multiple of this value. If a retry interval other than zero is specified that is less than the value of the TIMER parameter, the TIMER value is used instead. The RESTART parameter can be modified when the link is open or closed.

RSTATS Parameter

```
RSTATS={N | Y}
```

This optional parameter determines whether or not (Y or N) statistics are automatically reset to zero when the statistics interval expires. The default value is RSTATS=N. The RSTATS parameter can be modified when the link is open or closed.

SAF Parameter

```
SAF={Y | L | N}
```

If SAF=Y or SAF=L is specified, Entire Net-Work will call the SAF Interface for all incoming requests on this link; failure to load the Interface is considered a security violation and Entire Net-Work will shut down. If SAF=L, the calls are traced and the output directed to DDPRINT. An error code is transmitted to the user if access to SAF is denied. If a user exit is specified by the EXIT parameter and the SAF parameter is set to Y or L, incoming requests are made available to the user exit before NETSAF is called. The SAF parameter can be modified when the link is open or closed. The default value is N (No).

SNDTMOUT Parameter

```
SNDTMOUT={send-timeout | 10}
```

This optional parameter defines the amount of time, in seconds, that the VTAM line driver will wait for a message to be received by the partner's VTAM link. When the SNDTMOUT value is exceeded, the link is disconnected and message NETV039I is issued. The default value is 10. The SNDTMOUT parameter can be modified when the link is open or closed.

STATBLK Parameter

```
STATBLK={Y | N | YES | NO | ON | OFF}
```

This optional parameter specifies whether or not statistics about the blocking of messages are to be collected. This parameter is ignored if BLOCKMSG=N. The STATBLK parameter can be modified when the link is open or closed.

STATCMP Parameter

```
STATCMP={Y | N | YES | NO | ON | OFF}
```

This optional parameter specifies whether or not statistics about data compression are to be collected. This parameter is ignored if COMPRMSG=N. The STATCMP parameter can be modified when the link is open or closed.

STATINT Parameter

```
STATINT={statistics-interval | 3600}
```

This optional parameter specifies the amount of time, in seconds, before statistics are automatically reported or reset. Value may be 0 - 2 gigabytes. The default value is 3600. The STATINT parameter can be modified when the link is open or closed.

TRACESIZ Parameter

```
TRACESIZ= trace-table-size
```

This optional parameter specifies the size of the VTAM link specific trace table. The TRACESIZ parameter can be modified when the link is open or closed.

- The minimum value is 16384 (x'4000') bytes. A value less than 16384 is reset to 16384. The default value is the value of the TRACESIZ DRIVER parameter or, if none is specified, 16384.
- The maximum value is 4194304 (x'400000') bytes. A value greater than 4194304 is reset to 4194304.

WEIGHT Parameter

```
WEIGHT={relative-weight | 256}
```

This optional parameter specifies the weight of this link with respect to other links connected to the same node. If a given target can be reached by more than one path (chain of connected links), the path with the lowest weight is used. Slow or expensive links should be given a higher value than fast or inexpensive links. Values range from 1 to 999999. The default is 256. The WEIGHT parameter can be modified when the link is open or closed.

ZEDC Parameter

```
ZEDC={ Y | N }
```

This parameter indicates whether zEnterprise Data Compression (zEDC) compression can occur for the link. Valid values are "Y" or "N"; "N" is the default. Determination of whether or not zEDC data compression occurs is based on a combination of the settings of this parameter and the ZEDCINIT parameter on the NODE statement, as described in the following table:

LINK ZEDC Parameter Setting	NODE ZEDCINIT Parameter Setting	Result
Y	Y	Outbound buffers for the link are compressed.
Y	N	Outbound buffers are not compressed.
N	Y	Outbound buffers for TCPI links are not compressed, but other outbound buffers might be (depending on the setting of their LINK statement ZEDC parameters).
N	N	Outbound buffers are not compressed.

Note: If the node-to-node handshake indicates that the destination node does not support zEDC data compression, the outbound payload will not be compressed, regardless of any zEDC parameter settings on the NODE statement or any LINK statement.

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, ZEDC=Y can be specified only on z/OS systems that support zEDC. For complete information on

z/OS requirements for zEDC support, refer to IBM documentation regarding *zEnterprise Data Compression (zEDC)*.

When compression occurs it occurs on buffers with sizes greater than the value defined by the NODE statement's ZEDCSZ parameter.

ZEDCLOG Parameter

```
ZEDCLOG={ F | L | N }
```

This optional parameter indicates what level of trace data will be logged for zEDC compression processing. This trace data logging occurs independently of Entire Net-Work's global tracing parameter setting (LOG=YES or LOG=FULL parameter settings on the NODE statement). Valid values are described in the following table:

ZEDCLOG Setting	Result
F	Trace data is logged prior to and after compression and decompression processing. The amount of data logged is equivalent to the length of the data.
L	Trace data is logged prior to and after compression and decompression processing. The amount of data logged is 100 (x'64') bytes.
N	This is the default. No trace data is logged.

Note: The F and L settings of ZEDCLOG should be used sparingly; these settings greatly increase the DDPRINT output size.

The ZEDCLOG parameter, can be modified when a link is open or closed.

Note: If the node-to-node handshake indicates that the destination node does not support zEDC data compression, the outbound payload will not be compressed, regardless of any zEDC parameter settings on the NODE statement or any LINK statement.

zEnterprise Data Compression (zEDC) can occur only on z/OS operating systems. Consequently, the ZEDCSLOG parameter specification should be made only on z/OS systems that support zEDC. For complete information on z/OS requirements for zEDC support, refer to IBM documentation regarding *zEnterprise Data Compression (zEDC)*.

20 VTAM Operator Commands

The Entire Net-Work VTAM line driver has the ability to process operator commands that are directed to a specific link, to all links, or directly to the driver.

Operator Command Syntax

The VTAM line driver operator commands have the following format:

The following table describes this syntax.

Syntax Representation	Description
VTAM	Informs Entire Net-Work that the command is destined for the VTAM line driver.
target	A value that informs VTAM what the target of the command is, as follows: ■ Specify an asterisk (*) if the target is all links. ■ Specify the pound sign (#) if the target is the driver itself. ■ Specify the link name if the target is a specific link.
cmd	The operator command to be carried out.

Examples:

The following are examples of operator commands:

```
VTAM * CLOSE
```

```
VTAM # STATS
```

```
VTAM link3 CONNECT
```

Driver Commands

The Entire Net-Work VTAM line driver supports the commands listed in the following table when the target is the driver. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *driver-parms*	Dynamically changes the driver configuration. The ALTER command is followed by the driver configuration parameters to be altered. The driver configuration parameters are the same as those specified in the DRIVER statement. For example: `VTAM # ALTER ACCEPTUI=Y,MAXRU=8A` Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.
CLOSE	Disconnects all links that are connected to other nodes. Releases all resources held by the driver as well as all open links, and closes the driver.
LOGDON VTAM	Turns on selective logging for the VTAM line driver.
LOGDOFF VTAM	Turns off selective logging for the VTAM line driver.
OPEN	Reopens the driver after it has been closed with the CLOSE operator command or because of an access method failure. Allocates all the resources needed by the driver to communicate with VTAM.
RESET	Resets all statistics for the driver. Statistics are printed only if the STATS command precedes the RESET command.
SHOW	Displays the current configuration of the driver. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all driver specific control blocks and the driver specific trace table to be snapped (printed in hexadecimal). Link specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. To print and reset statistics, specify RESET immediately after the STATS command. Statistics are printed to both the DDPRINT and SYSLOG. For example: `VTAM # STATS RESET`
STATUS	Displays the current status of the driver as well as the number of messages received and sent.
TRACE	Causes the driver specific trace table to be formatted and printed. The trace table is also formatted and printed automatically when the SNAP command is processed.

Link Commands

The Entire Net-Work VTAM line driver supports the commands listed in the following table when the target is a link or all links. The underlined portion of the command is the minimum abbreviation.

Command	Action
ALTER *link-parms*	Dynamically changes the link configuration. The ALTER command is followed by the link configuration parameters to be altered. The link configuration parameters are the same as those specified on the LINK statement. For example: `VTAM linkname ALTER,PSTAT=Y` Refer to the specific parameter description for information on possible restrictions about modifying the parameter using the ALTER command.
CLOSE	Disconnects the link if it is connected to another node and releases all resources held by the link.
CONNECT	Starts the connect sequence for the link. That is, attempts to establish one or more VTAM sessions with the LU(s) associated with the target link(s). If the link is already connected or is in the process of connecting, the command is ignored.
DISCONNECT	Starts the disconnect sequence for the target link(s). If the link is already disconnected or is in the process of disconnecting, the command is ignored.
LOGLON *linkname*	Turns on selective logging for the specified link.
LOGLOFF *linkname*	Turns off selective logging for the specified link.
OPEN	Allocates all of the resources needed by the link to communicate with VTAM. Does not initiate a connect to the remote node. The status of the link displayed with the SHOW operator command is not affected by the OPEN request.
RESET	Resets all statistics for the link. Statistics are printed only if the STATS command precedes the RESET command.
RESUME	Restarts processing on a link that was temporarily stopped due to a SUSPEND command.
SHOW	Displays the current configuration of the link. The current configuration is always shown automatically following an ALTER command.
SNAP	Causes all link specific control blocks and the link specific trace table to be snapped (printed in hexadecimal). Driver specific control blocks and Entire Net-Work specific control blocks are not snapped.
STATS	Causes the immediate printing of statistics and restarts the statistics interval. To print and reset statistics, specify RESET immediately after the STATS command. For example: `VTAM linkname STATS RESET`

Command	Action
STATUS	Displays the current status of the link as well as the number of messages received and sent.
TRACE	Causes the link specific trace table to be formatted and printed. The trace table is also formatted and printed automatically when the SNAP command is processed.

21 VTAM Line Driver Model Links

To reduce setup and maintenance efforts for the VTAM line driver, the "model" link facility has been implemented. This allows users to code one or more model links with parameter values that serve as default values for many partners, instead of coding one LINK statement for each partner. As each partner connects, new control blocks are allocated and initialized from the model link. A model link is defined as any link that meets one of the following criteria: the LUNAME or the APPLID parameter contains one or more wildcard characters (the wildcard character is a question mark), or neither the LUNAME nor the APPLID parameter is specified.

Once a model link is defined, it remains a model link even if its defining characteristics are modified with the ALTER operator command.

Selection Criteria

The following criteria are used for model link selection:

- LU type (LU6.0, LU6.2, LU0)
- Parallel session support (Yes/No)
- LUNAME or APPLID (wildcard characters match any character)
- MODEENT name

The model link selection process may include a call to a user exit to allow the user to make the final selection. The EXIT parameter in the LINK statement is used to specify the name of the user exit. See the section ***VTAM LINK Statement*** Parameters . If a user exit is specified, it is called after a model link is selected. The user exit can override the selection by modifying the link name.

22 VTAM Line Driver User Exits

Entire Net-Work supports a user exit interface that allows the VTAM line driver to call an optional user exit. Read *User Exit Interface* in the *Entire Net-Work Application Programming Guide*.

23 Connecting to UES-Enabled Adabas Databases

Prior to Adabas version 7, Entire Net-Work converted all data for mainframe Adabas when necessary from ASCII to EBCDIC. Starting with version 7, Adabas is delivered with its own data conversion capability (module LNKUES); that is, Universal Encoding Support (UES). Entire Net-Work detects when it is connected to a target database with UES support and passes the data through to Adabas without converting it.

In order for UES support to work, various ADALNK modules must be linked to the Adabas UES module, LNKUES. LNKUES converts data in the Adabas buffers and byte swaps, if necessary, depending on the data architecture of the caller. All currently supported versions of Adabas provide a UES-enabled ADALNK module; so there is nothing you need to do for UES support. However, if you have altered the translation table, you will need to assemble and link the ADALNK module again.

Important: If an ADALNK batch link routine has been linked or modified by Software AG product modules or user exits, it cannot be used in any application startups of Adabas utility jobs or Adabas, Entire System Server, Adabas Review Hub, or Entire Net-Work nuclei.

Overview of UES Support

This section provides a general overview of the UES support provided in Adabas. For detailed information about UES support in Adabas, refer to the Adabas installation documentation for the operating system you are using.

In order for UES support to work, various ADALNK modules and tables must be linked to the Adabas UES module, LNKUES. LNKUES converts data in the Adabas buffers and byte-swaps, if necessary, depending on the data architecture of the caller.

This section covers the following topics:

- Connection Possibilities

UES-Linked Load Modules

The ADALNK load modules that have been linked with module LNKUES vary, depending on the operating system and environment you are running. For a complete list of the modules that have been linked, read the UES sections of the appropriate Adabas installation documentation.

UES Translation Tables

Two standard translation tables are provided with Adabas UES support:

- ASC2EBC: ASCII to EBCDIC translation
- EBC2ASC: EBCDIC to ASCII translation

The Adabas and Entire Net-Work translation table pairs are provided in the appropriate Adabas installation documentation.

You can use the supplied translation tables, or you may prepare your own customized translation tables, reassemble them , and link them with the LNKUES module. Using your own customized translation tables should only be necessary if you require the use of some country-specific character other than the standard A-Z, a-z, or 0-9 characters in the additions 1 (user ID) or additions 3 field of the control block. For detailed information on using the default or customized translation tables, refer to the UES sections of the appropriate Adabas installation documentation.

If you prefer to use the same translation tables that are used in Entire Net-work, change the COPY statements in ASC2EBC and EBC2ASC from UES2ASC and UES2EBC to NW2ASC and NW2EBC, respectively. After modifying the translation tables, be sure to (re)assemble them and link them with the delivered LNKUES module. The sample jobs referenced in ***Connecting to UES-Enabled Databases through Entire Net-Work*** and ***Connecting to a UES-Enabled Database through ADATCP*** include steps that reassemble and link the translation tables with LNKUES.

Job Steps

Job library members are provided with Adabas for each operating system it supports to assemble and link the appropriate modules with the UES components. For more information, read the UES sections of the appropriate Adabas installation documentation.

Calling LNKUES

On all platforms, LNKUES receives control before UEXITB for UES requests and after UEXITA for UES replies.

Connection Possibilities

UES-enabled databases are connected to machines with different architectures through Entire Net-Work and optionally through ADATCP -- a direct TCP/IP link to the Adabas nucleus from web-based applications such as Software AG's Jadabas. These connection methods are described elsewhere in this section.

Environment Requirements

To support UES-enabled databases, be sure that your environment meets the requirements described in this section.

- Database Requirements
- Data Set Requirements
- SYSPARM Requirements
- Verify Required ADALNK Module Available

Database Requirements

The Adabas database must be UES-enabled (it must have been defined with UES=YES set during the ADADEF utility run). For complete information read about database maintenance tasks in the Adabas DBA tasks documentation and about the ADACMP and the ADADEF utilities in the Adabas utilities documentation.

In general, to UES-enable an Adabas database:

- Specify `MODIFY UES=YES` in the Adabas ADADEF utility settings for each target database.

Data Set Requirements

Make sure that the internal product libraries described in this section are loaded and concatenated correctly.

- Required Internal Product Libraries
- Adabas JCL Updates

- Disk Space Requirements for Internal Product Data Sets

Required Internal Product Libraries

Software AG internal product libraries that are required if you intend to enable a database for universal encoding service (UES) support are now delivered separately from the product libraries. For UES support, the following libraries must be loaded and included in the STEPLIB or LIBDEF concatenation:

```
APS272.MVSLDnn
```

where *nn* is the load library level. If the library with a higher level number is not a full replacement for the lower level load library(s), the library with the higher level must precede those with lower numbers in the steplib concatenation.

Also for UES support, the following library must be loaded and included in the session execution JCL:

```
ADABAS.Vvrs.ADAvrs.MVSECOn ↩
```

This library includes all supported code pages. For more information about the supported code pages, read about *Supplied UES Encodings* in your Adabas documentation.

Adabas JCL Updates

If you intend to enable your database for universal encoding service (UES), the startup job for the Adabas nucleus must be updated as described in this section.

- The MVSLD*nn* internal product libraries must be concatenated in the STEPLIB or LIBDEF. The following is an example of such a STEPLIB concatenation:

  ```
  //STEPLIB DD DISP=SHR,DSN=ADABAS.Vvrs.ADAvrs.MVSLOAD
  //        DD DISP=SHR,DSN=ADABAS.Vvrs.APSvrs.MVSLDnn
  ```

  where *nn* is the load library level. If the library with a higher level number is not a full replacement for the lower level load library(s), the library with the higher level must precede those with lower numbers in the steplib concatenation.
- If you intend to connect your UES-enabled database directly through ADATCP, the ADATCP library must also be concatenated in the STEPLIB. The following example shows a sample DD statement in a STEPLIB concatenation:

```
//DD DISP=SHR,DSN=SAG.WCPvrs.MVSLOAD
//DD DISP=SHR,DSN=SAG.WTCvrs.MVSLOAD
```

- Also for UES support, the following ECS objects data set must be loaded and included in the session execution JCL:

```
//DDECSOJ DD DISP=SHR,DSN=ADABAS.Vvrs.ADAvrs.MVSECOn ↩
```

Note: The data set DDECSMF (messages) previously required for UES support no longer exists and reference to it needs to be deleted from your JCL. Likewise, the CONFIG DD "dummy" data set is no longer needed.

Disk Space Requirements for Internal Product Data Sets

The minimum disk space requirements on a 3390 disk for the internal product libraries delivered with Adabas is as follows:

Libraries	3390 Cylinders	3390 Tracks	Directory Blocks
APS272.MVSLD00	6	86	100

SYSPARM Requirements

To support UES, you need to add SYSPARM statements and parameters to your session execution JCL, as follows:

```
//SYSPARM DD*
SYSTEM_ID=ADAAPS
ABEND_RECOVERY=NO
THREAD_ABEND_RECOVERY=NO
```

These SYSPARM statements and parameters are required for the APS internal product.

If you are using the ADATCP component with UES support, you will need an additional parameter, CDI_DRIVER, to identify the TCP/IP stack you are using, as follows:

```
CDI_DRIVER=('tcpip,PAAOSOCK,ADDRSPCE=STACKNAME') <--IBM TCP/IP stack for OES
```

Verify Required ADALNK Module Available

Ensure that the ADALNK module in Entire Net-Work's STEPLIB or LIBDEF is UES-enabled. ADALNK is UES-enabled by default. WAL Version 8 (or later) is included with Entire Net-Work Version 3.

Connecting to UES-Enabled Databases through Entire Net-Work

The ADALNK module has been UES-enabled for you, so there are no specific steps you must perform to connect to UES-enabled databases. However, if you have altered the translation tables, you will need to perform the step described in this section (depending on the platform) that assembles and links the updated translation tables into ADALNK.

In all cases, whenever you alter ADALNK, you must be sure to make the updated module available to Entire Net-Work.

- Assembling and Linking ADALNK on z/OS Systems
- Assembling and Linking ADALNK on z/VSE Systems
- Assembling and Linking ADALNK on BS2000 Systems

Assembling and Linking ADALNK on z/OS Systems

To assemble and link the ADALNK module on z/OS systems:

1 Assemble and link the modified batch ADALNK with the translation tables and LNKUES. You can use the sample JCL found in MVSJOBS member ALNKLNK7 in the WAL libraries. Make sure you:

- Provide all necessary job card information.
- Check the symbolic parameter value for version, revision level, and SP level (*vrs*). It must reflect the level of your Adabas source and load libraries.
- Check the data set names for SYSLIB, SYSIN, SYSLMOD, and SYSLIN in the SAGASM and LINKALL inline procedures.

2 Once it is successfully linked, make ADALNK available to Entire Net-Work's job STEPLIB concatenation list.

For complete instructions on UES-enablement in Adabas, refer to the UES sections of your Adabas documentation.

Assembling and Linking ADALNK on z/VSE Systems

To assemble and link the ADALNK module on z/VSE systems:

1 Assemble and link ADALNK with LNKUES and the translation tables into a separate sublibrary for use with UES only (this is sample job ALNKLNK8.X in the Adabas or WAL library). Otherwise, you can use the following sample JCS:

```
// OPTION CATAL,LIST
// EXEC PROC=ADAVVLIB
// LIBDEF OBJ,SEARCH=(ADALIB.ADAVRS)
// LIBDEF PHASE,CATALOG=ADALIB.ADAVRS
 PHASE ADALNK,*,NOAUTO
 MODE AMODE(31),RMODE(24)
 INCLUDE LNKVSE8
 INCLUDE LNKIND
 INCLUDE LNKUES
 INCLUDE LNKDSL
 INCLUDE RTRVSE
 INCLUDE ASC2EBC
 INCLUDE EBC2ASC
 INCLUDE LNKGBLS
 ENTRY ADABAS
// EXEC LNKEDT,PARM='MSHP'
```

2 Once it is successfully (re)linked, make ADALNK available to Entire Net-Work's job LIBDEF concatenation list. If you are calling Adabas version 7 and you do not have the correct LNKUES/ADALNK module, Adabas produces unexpected results: response code 022 (ADARSP022), 253 (ADARSP253), etc.

For complete instructions on UES-enablement in Adabas, refer to the UES sections of your Adabas documentation.

Assembling and Linking ADALNK on BS2000 Systems

To assemble and link the ADALNK module on BS2000 systems:

- Link or bind LNKUES, ASC2EBC and EBC2ASC to the ADALNK module.

  The following sample JCL takes modules from $SAG.ADABAS.MOD and makes a new ADALNK module with LNKUES and the translation tables (ASC2EBC and EBC2ASC) in $SAG.ADABAS.ADALNKUES.MOD:

```
/BEGIN-PROC A,PROC-PAR=(-
/ &ADAL=$SAG.ADABAS.ADALNKUES.MOD,-
/ &ADAS=$SAG.ADABAS.MOD,-
/ &LIB=$SAG.ADABAS.PROTOCOL,-
/ &MEMALNK,-
/ &MODALNK-
/ ),ESC-CHAR='&'
/REMA
/REMA
/REMA
/MOD-TEST DUMP=NO
/ASS-SYSLST *L(&LIB,LNK.&MEM)
/ASS-SYSDTA *SYSCMD
/REMA
/STA-PROG $TSOSLNK
MOD &MOD,LET=Y,LIB=&ADAL,XREF=Y,PR=Y,SORT=Y
NCAL
TRAITS AMODE=ANY,RMODE=ANY
LINK-SYMBOLS HIDE=LNKUES
INCLUDE &MEM,&ADAS
INCLUDE LNKUES,&ADAS
INCLUDE ASC2EBC,&ADAS
INCLUDE EBC2ASC,&ADAS
BIND
/ASS-SYSLST *PRIM
/ASS-SYSDTA *PRIM
/END-PROC            ↩
```

For complete instructions on UES-enablement in Adabas, refer to the UES sections of your Adabas documentation.

Connecting to a UES-Enabled Database through ADATCP

To connect to a UES-enabled database through ADATCP (the direct TCP/IP link), you must:

- Update the ADALNK LNKGBLS table so it identifies the SVC number that ADATCP should use. This is described in **Step 1** below.

- Ensure that the database has been UES-enabled. This is accomplished by updates to the ADALNK module. By default, the Adabas ADALNK module has been UES-enabled for you. However, if your site has altered this or if you have altered the translation tables, you will need to make the updates described in **Step 2** below and in ***Connecting to UES-Enabled Databases through Entire Net-Work***, earlier in this section.

  Whenever you alter ADALNK, you must be sure to make the updated module available to Entire Net-Work. This is described in **Step 3** below.

- Verify that you are using the correct libraries in your STEPLIB concatenation and that appropriate zaps have been applied. This is described in **Step 4** below.

Once you have completed these steps, you should activate the TCP/IP link, as described in ***Activating the TCP/IP Link***, elsewhere in this section.

1. Prepare a modified batch ADALNK.

Update the source LNKGBLS table as follows:

```
SVCNO=nnn,          Default Adabas SVC number ↩
```

where *nnn* is the SVC number on which the system is to run.

2. Assemble and link ADALNK with the translation tables and LNKUES

This can be performed by tailoring the supplied jobs ASMGBLS and ALNKUES. Make sure you:

- Provide all necessary job card information.
- Check the symbolic parameter value for version, revision level, and SP level (*vr* SP *sp*). It must reflect the level of your Adabas source and load libraries.
- Check the data set names for SYSLIB, SYSIN, SYSLMOD, and SYSLIN in the SAGASM and LINKALL inline procedures.

3. Make ADALNK Available to the ADATCP-Enabled Adabas Nucleus

The (re)linked ADALNK must be made available to the ADATCP-enabled Adabas nucleus.

On BS2000 systems do one of the following:

- In a permanent file attached to the link name DDLNKPAR, set parameter ADALNK IDTNAME=*iiiiiiii* (where *iiiiiiii* is the IDTNAME where the system in being run); or
- In the SSFB2C(S) job supplied in the Adabas source library, provide the following setting: `B2CONFIG MF=C,ENVNAME=`*iiiiiiii*, where *iiiiiiii* is the IDTNAME where the system is being run. Then use the sample job ASMBS2(J) to assemble it and use the sample job ALNKUES(J) to prepare the LNKUES.

4. Verify Correct Libraries are Concatenated and Zaps are Applied

Make sure the following libraries are used:

- Verify that the Adabas 8.2.6 L001 library (or the load library from a later Adabas release) containing the latest ADATCP module is included in the nucleus STEPLIB concatenation.
- Verify that the latest Entire Net-Work (WCP) and Entire Net-Work TCP/IP Option (WTC) 6.3 SP2 load libraries (or those of later versions) are in the STEPLIB concatenation.
- In Adabas 8.2 SP5 environments, verify that special Adabas ADASVC zap AY825032 or AI825031 have been applied to the SVC on which the ADATCP-enabled database will run. In Adabas 8.2 SP6 environments, verify that special ADASVC zap AI826007 has been applied to the SVC on which the ADATCP-enabled database will run.

Trouble Shooting

The following table lists some common errors and their causes:

If you get this error:	It means that:
ABEND U658	You forgot to link "LNKUES"
AP1001I TRANSPORT ENDPOINT ERROR	Someone else is using the same port number
response 254 or response 5632 (which is really response 22 but byte-swapped)	ADALNK is not configured correctly
ADACM00I:RECV error number 00015 socket 00006 or ADACM001:RECV error number 01121 socket 00006	A 'connection reset' took place. The client ended abnormally or failed to call the Adabas signoff() method. ADATCP is informing you that the client left; ADATCP will clean up the connection.

Activating the ADATCP Direct TCP/IP Link

To activate a direct TCP/IP link to the Adabas nucleus:

1 Set the ADARUN parameter TCPIP=YES.

2 Specify a universal resource locator (URL), as described in this section.

- Specifying a URL

- Managing URLs

Specifying a URL

The URL is a 20-byte address that conforms to the RFC specification for URLs. Use the ADARUN parameter TCPURL to specify the URL required to activate the direct TCP/IP link. For more information, read *TCPURL Parameter: TCP/IP Universal Resource Locator*, in the *Entire Net-Work Reference Guide*.

Managing URLs

The URLs can be managed using the TCPIP operator command, as illustrated below:

```
TCPIP={ OPEN=url | CLOSE=url | CLOSE }
```

where *url* is the URL for the TCP/IP link you want to open or close and has the same format as the ADARUN TCPURL parameter:

```
api-name:[//]stackid:port-number[:logging-setting[:allowipv6-setting]]
```

The command allows you to open or close a TCP/IP link to the Adabas nucleus or to close all links. It can only be used when ADARUN TCPIP=YES and all conditions for that setting have been met. This command can be used to close the URL set in the ADARUN TCPURL parameter, or to open or close additional TCP/IP links.

Examples

```
TCPIP=OPEN=OES://:1234
TCPIP=CLOSE=OES://:1234
```

To close all open URLs:

```
TCPIP=CLOSE
```

24 Tuning the Compression Algorithm

Entire Net-Work provides a message compression capability that can be selected or deselected for each link using the COMPRMSG parameter on the LINK statement. If COMPRMSG=YES is specified, outgoing messages over that link are compressed. Incoming compressed messages are always decompressed, regardless of the parameter setting. If a compressed message passes through a relay node en route to its destination, it is not decompressed until it reaches the target node.

Compression is accomplished "in place", but decompression requires a second buffer as an output area. This fact should be considered when estimating storage requirements. An output area is needed even without compression if blocking is in effect.

Compression consists of replacing sequences of identical characters with a record describing how many characters are deleted and where to reinsert them upon decompression.

The compression algorithm can be tuned beyond the simple yes/no decision afforded by the COMPRMSG parameter by using the MINCMP parameter or a translation table. The objective is to trigger compression where it is expected to be effective and to skip cases where the cost outweighs the benefits.

Note: The COMPRMSG and MINCMP parameters apply only to the *DCAM LINK statement* and the *VTAM LINK statement*. These parameters are not used by the other Entire Net-Work line drivers.

Using the MINCMP Parameter

The MINCMP parameter can be used to specify the shortest sequence of identical characters to be compressed. The nature of the algorithm is such that at least six characters must be found before anything is gained. Therefore, MINCMP defaults to 6. If a value less than 6 is specified, it is automatically reset to 6.

Values greater than 6 can be specified with the effect that compression takes place only where substantial savings can be expected. This results in less CPU overhead than full compression at the cost of a slight reduction in the overall compression factor.

A value for MINCMP should be selected based on typical message contents with the objective of reducing the number of decompressions needed, thereby reducing both storage and CPU requirements.

The following example describes three applications with various compression requirements, and provides a method for using the MINCMP parameter to effectively handle compression for all three applications.

Application	Characteristics
A	has record buffers defined that contain many relatively long alpha fields, some of which tend to be all blanks.
B	uses special characters such as dashes in its screen maps to form boxes or to prefill input fields.
C	processes bit maps. Bit maps contain many fixed-length binary fields that usually have rather diverse values. Occasionally, a few consecutive bytes may have the same value.

In applications A and B, compression is likely to produce a reasonable reduction in the amount of data transmitted.

In application C, the occasional sequences can be compressed away, but the message length will not be reduced by much. The expected savings is probably not worth the CPU cycles spent for compression and decompression, plus the extra storage needed for the output buffer required for decompression.

If these applications all run through the same link on the network, the MINCMP specification for the link can be used to tune compression. The value specified for MINCMP will be

- not larger than most of the sequences of blanks in application A; and
- not larger than the sequences of special characters in application B; but
- larger than most of the "accidental" sequences in application C.

The sequences in applications A and B are then compressed with good results, and no time is wasted finding the occasional sequences in application C.

Using a Translation Table

A CSECT named NETTRT is linked into the NETWRK control module. This CSECT is exactly 256 bytes long and contains a translation table used by a Translate-And-Test (TRT) instruction in the compression algorithm. The TRT and the table in NETTRT are used to find characters in the message that are likely candidates for compression. Only the characters translating into a non-zero function value are considered for compression. The assumption behind this algorithm is that certain characters, such as a blank or a binary zero, are more likely to occur in sequences than others, such as the lower case letter "k". The compression process is faster if "unlikely" characters are not considered for compression.

The translation table provided by Software AG is a reasonable starting point for most situations, but users can and should modify or replace NETTRT with their own version if their data streams differ from the usual pattern. If, for instance, your company logo appears on a map built from many occurrences of the character "O", you should probably change NETTRT to contain a non-zero value in the position related to "O". The following is AMASPZAP input for z/OS systems to change NETTRT so that "O" will be considered for compression:

```
NAME NETWRK NETTRT
VER  D6 00             X'D6' = C'0'
REP  D6 FF
```

25 Target ID Handling in the Network

(Global) Target IDs Must Be Unique

Entire Net-Work normally enforces unique global target IDs throughout the network, that is, across all connected nodes. Unless both targets are local, a target is not permitted to start if another target with the same ID is active anywhere in the network. The ADAM98 message (Target Initialization Error) will result.

When a link is established between two previously unconnected partial networks, it is possible that the same target ID is used in both parts simultaneously. Because target IDs must be unique within the network, this is considered an error situation. Entire Net-Work cannot determine which of the targets should be deactivated. Therefore the link between the two parts of the network is not established and explanatory messages (NET0108 and NET0106) are written to the operator consoles of both the nodes trying to connect. It is the responsibility of the installation to terminate one of the conflicting targets. The link can then be connected from either side by using the CONNECT LINK operator command.

Duplicate Target IDs Restricted to LOCAL Node

Adabas provides a feature that permits the use of duplicate target IDs under special circumstances. A target ID can be used on all nodes in the network simultaneously if all targets with that ID are declared LOCAL (see the *Adabas Operations Manual* for the appropriate ADARUN parameter). In this case, any one of the targets with that ID can be accessed only by users running on the same node.

Entire Net-Work does not transport requests to LOCAL targets. The RSP148 message is generated if a user program calls a target declared LOCAL from a remote node.

A target ID cannot be used simultaneously for both a local target and a global target. The reason for this restriction is illustrated in the following example.

Imagine that a (global) database, DBID 35, is active on node PROD, and user programs from all nodes in the network are issuing calls to it. If a local database, also with DBID 35, starts on node TEST, it immediately draws all calls that originate on node TEST to the local database DBID 35. Users on other nodes are not affected, but all users on node TEST are switched in mid-transaction from the global database DBID 35 to the local database DBID 35, a different and probably totally unrelated database. This error situation cannot be permitted.

Entire Net-Work enforces this rule and thus prevents the confusion that would result if a local target ID coincided with a global target ID by broadcasting information about local targets to all nodes the same way it does for global targets.

26 Estimating Entire Net-Work Storage Requirements

Given the complexity of today's data processing environments, it is almost impossible to provide methods to predict the exact storage requirements of a software product.

The following tables provides rough estimates about the fixed storage requirements of Entire Net-Work and its various components, ignoring operating system-related storage requirements, which typically vary from installation to installation.

Table 1 contains the amounts of storage obtained from the operating system based on parameter specification or appropriate defaults. It does not include storage areas that are directly related to the operating system, such as operating system control blocks, I/O-related buffers, and control blocks (except where they are part of Entire Net-Work program modules or data areas).

Table 2 contains the amounts of storage obtained from the Entire Net-Work buffer pools by the control module and the various line drivers.

Table 3 contains special storage requirements of the line drivers (such as special common system storage areas) in the various operating system environments.

Table 1: Storage Areas Obtained from System

Storage Area		Platform			
		z/OS	VSE	VM	BS2000/OSD
Request queue: (NC parameter+1)*192		AS(X)	SYS/Part	Virt.M	Comm.Pool
Attached buffers: (NAB parameter*4112)		AS(X)	SYS/Part	Virt.M	Comm.Pool
Entire Net-Work buffer pools*	Asynchronous buffers	AS(X)	Part	Virt.M	AS(X)
	Long-term buffers	AS	Part	Virt.M	AS
	Short-term buffers	AS(X)	Part	Virt.M	AS(X)
	Page-fixed buffers	AS	Part	Virt.M	AS
Entire Net-Work trace table		AS(X)	Part	Virt.M	AS(X)
Entire Net-Work control blocks	general	AS	Part	Virt.M	AS
	Node	48	48	48	48
	Target	32	32	32	32
	Path	32	32	32	32
	CTCA DRIVER	544	---	---	---
	DCAM DRIVER	---	---	---	848
	IUCV DRIVER	---	4KB	---	---
	TCPI DRIVER	4KB	4KB	4KB	4KB
	TCPX DRIVER	4KB	4KB	4KB	4KB

Storage Area		Platform			
		z/OS	VSE	VM	BS2000/OSD
	VTAM DRIVER	4KB	4KB	4KB	4KB
	XCF DRIVER	2048	---	---	---
	CTCA LINK	992	---	---	---
	DCAM LINK	---	---	---	56
	IUCV LINK	---	168	168	168
	TCPI LINK	1KB	1KB	1KB	1KB
	TCPX LINK	1KB	1KB	1KB	1KB
	VTAM LINK	256	256	256	256
	XCF LINK	2048	---	---	---
ADAIOR data areas	general	AS	Part	Virt.M	AS
	(for trace table, ECB list, etc.)	about 2KB	about 2KB	about 2KB	about 2KB

Abbreviation	Meaning
AS	from address space (private, below 16MB if XA or XS)
AS(X)	from address space (private, above 16MB if XA or XS)
SYS	from system GETVIS area (VSE in VAE mode)
Part	from partition (VSE not in VAE mode)
Virt.M	from virtual machine
Comm.Pool	from common memory pool

Table 2: Storage Obtained from Entire Net-Work Buffer Pools

Statistic	Buffer Pool Types			
	Asynch	Long-term	Short-term	Page-fixed
Segment size	64	64	512	2KB or 4KB
Control module buffer pool usage		UB	MSG RPLY	
Queue manager buffer pool usage			BLK	BLK
IUCV line driver buffer pool usage	2WAY INTQEL	PATHS SMBLK		2WAY RECV
DCAM line driver buffer pool usage	WTO		SMBLK	

Abbreviation	Meaning
BLK	Storage for outgoing transmission blocks (after compression and blocking), from short-term pool or page-fixed pool, depending on line driver requirements. Storage requirements for one transmission block include, in addition to the messages contained, 48 bytes for a transmission block header.
INTQEL	1 segment for each IUCV message received (held until reply).
MSG	All messages sent or received; output messages kept until acknowledged by the access method, input messages kept until processed.
	The size of a message can be computed in the following way: 56 bytes for a message header + maxpath * 2 bytes for a node stack + 128 bytes for UB, ACB, etc. + size of FB, RB, SB, VB, IB to send or receive
PATHS	4 * the number of possible IUCV paths specified in the MAXCONN option of the CP directory (rounded up to nearest segment) + 1 segment per active IUCV path.
RECV	Storage for incoming message data until transferred to short-term buffers (in paging systems only).
RPLY	A reply buffer for each user request for a target on this node if the information returned by the target will not fit into the original message buffer (that is, if a large record buffer or ISN buffer is to be returned to the user).
SMBLK	1 segment for each message that cannot be sent in one transmission, kept until the message is acknowledged by the adjacent node.
UB	(only if 31-bit mode:) 64 bytes per user request for a target on this node, for the duration of the Adabas call.
WTO	Operator messages; kept until the message is written. DCAM: 2 asynchronous segments for messages from asynchronous routines.
2WAY	Storage for reply messages in the length of messages sent when the SEND 2WAY protocol is used (page-fixed in paging systems only).

Table 3: Special Storage Requirements of Line Drivers

Driver	Special Storage Requirements
IUCV	12288+(maxcon*msglim)*48 (for FLIH, SVC, subsystem vector table, IUCV buffers, interrupt queue elements, working storage, and the cross-product interface tables):
	z/OS: from CSA (SP228) or SQA (SP245) VSE: from system GETVIS (page-fixed)
DCAM	MAXBLK (for receive buffer):
	BS2000/OSD: from AS
TCPX	NUMUSERS*256 is initially allocated from buffer pool storage for the Active Client Table (ACT). This value may dynamically expand if required.

Note: In addition to the storage estimates shown in the table, approximately 250KB storage is required for executable code.

Index

I

L

M

N

O

P

Q

R

S

T

U

V

W

Z