

Fujitsu Software openUTM Enterprise Edition (Linux and Windows systems)

Version 7.0A

October 2025



Release Notice

All rights reserved, including intellectual property rights. Technical data subject to modifications and delivery subject to availability. Any liability that the data and illustrations are complete, actual or correct is excluded. Designations may be trademarks and/or copyrights of the respective manufacturer, the use of which by third parties for their own purposes may infringe the rights of such owner.

*2 Copyright © 2025 Fujitsu

The Fujitsu brand and the Fujitsu logo are registered trademarks of Fujitsu Limited, Japan in Europe and other countries.

BS2000 is a trademark of Fujitsu Germany GmbH in Europe.

*2 Oracle™ and Java™ are trademarks or registered trademarks of Oracle Corporation,
*2 Redwood Shores, USA.

*2 Windows™ is a trademark or registered trademark of Microsoft Corporation, Redmond,
*2 USA

*2 IBM™, CICS™ and z/OS™ are trademarks or registered trademarks of International
*2 Business Machines Corporation, USA.

*2 Linux® is a registered trademark of Linus Torvalds.

1	General	1
1.1	Ordering	2
1.1.1	Licenses	2
1.2	Delivery	3
1.2.1	Software on data media	3
1.2.2	Software for downloading	3
1.3	Documentation	4
1.3.1	Availability	4
1.3.2	Documentation of the Open Group interfaces	4
2	Software extensions	5
2.1	New functions in openUTM (Linux and Windows systems) V7.0A	5
2.1.1	UTM as HTTP Server	5
2.1.2	Encryption	6
2.1.3	Subnets	7
2.1.4	Access data for the XA database connection	7
2.1.5	Reconnect for the XA database connection	7
2.1.6	Other changes	7
2.2	New functions in openUTM (Linux and Windows systems) V7.0A23	8
2.3	New functions in openUTM (Linux and Windows systems) V7.0A35	8
2.4	Implemented change requests (CR)	8
3	Technical information	9
3.1	Resource requirements	9
3.2	Software configurations	9
3.3	Product installation	10
3.3.1	Installation on Linux platforms	10
3.3.2	Installation on Windows platforms	11
3.3.3	Installation together with C++ environment on Linux platforms	11
3.3.4	Installation logs	11
3.3.5	Notes: Kernel parameters for openUTM	11
3.4	Product use	12
3.4.1	Notes for upgrading to openUTM V7.0A	12
3.4.2	Upgrading from openUTM (Linux and Windows systems) V7.0A00 to V7.0A23	12
3.4.3	Upgrading from openUTM V7.0A23 to openUTM V7.0A35	12
3.4.4	Use of kdcdef utility under RHEL 10	12
3.4.5	Converting a stand-alone application to a UTM-Cluster application	13
3.4.6	Client programs with a local connection to openUTM	13
3.4.7	Linking UTM OSI-TP applications on Linux platforms	13
3.4.8	Linking and running UTM applications on Linux systems	13
3.4.9	Notes on a sample application for UTM-Cluster applications	13
3.5	Discontinued functions and those to be discontinued	13
3.5.1	Discontinued functions	13
3.5.2	Functions to be discontinued	14
3.6	Incompatibilities	14
3.7	Restrictions	14
3.8	Procedure in the event of errors	14
3.9	Additions to the User Guide	16
3.9.1	openUTM Messages, Debugging and Diagnostics	16
3.9.2	openUTM Administering Applications	16
3.9.3	Additional inserts in U315	16
4	Hardware support	17

1 General

The open and universal transaction monitor openUTM supports the creation and operation of transactional applications under the Fujitsu operating system BS2000, Linux and Windows systems. Some commonly available UNIX systems can be supported on special request.

Fujitsu Software openUTM lets you implement client/server architectures and create distributed transaction applications in homogenous and heterogeneous IT infrastructures (multi-tier architecture).

Some tools are included free of charge in openUTM product:

- The web based tool openUTM WebAdmin is ready for administration of openUTM applications via Internet browser.
- "XML for openUTM" provides openUTM with a convenient interface for creating and processing data in the form of XML documents.

Additionally, openUTM can be combined with further chargeable products:

- Fujitsu Software openUTM Client is a product which allows client/server communication with openUTM server applications and is available as a separate product for the Fujitsu operating system BS2000 as well as for Linux and Windows systems and on request also for UNIX systems.
- The add-on product Fujitsu Software openUTM LU62 (Unix, Linux and Windows systems) V5.1 supports connections to CICS/IMS applications on IBM systems via SNA LU6.2.
- *2 • The product Fujitsu Software BeanConnect enables UTM applications to connect to Jakarta EE Application Servers. BeanConnect also includes a component for creating Java clients for openUTM.

Separate Release Notices are available for all tools.

More information about the above mentioned products and the data sheets is available at:

*1 [openUTM - Transaction Monitor : Fujitsu EMEA](#)

openUTM is part of the comprehensive middleware suite openSEAS.

The middleware suite openSEAS enables the simple integration of individual applications and the interaction of existing and new applications.

openSEAS perfectly complements the Fusion middleware of our strategic partner Oracle. Oracle Fusion middleware is a market-leading family of pre-integrated products, which ranges from application infrastructure development applications through to business intelligence.

Together this provides the platform for:

- Individual business-critical IT solutions
- Company portals
- Integration solutions for service-oriented architectures
- Web applications and portals
- Business processes

This Release Notice is a summary of the major extensions, dependencies and operating information with respect to openUTM (Linux and Windows systems) V7.0A and all the information necessary for migrating from the predecessor version openUTM (Linux and Windows systems) V6.5A.

Changes which have become known subsequently are updated and provided in this file.

*2 The content refers to release level: October 2025.

*1 Changes compared to openUTM V7.0A from November 2019 are marked with *1.

*2 Changes compared to openUTM V7.0A23 from June 2022 are marked with *2.

*2 The content has been modified compared to release status V7.0A23.

*2 It refers to release level V7.0A35: October 2025.

All product Release Notices including this one are available on the Internet at the following URL:

<https://bs2manuals.ts.fujitsu.com>

If one or more predecessor versions are skipped when this current product version is in use, it may be necessary to also take account of the information from the Release Notices (and/or README files) of the predecessor versions.

The use of names, trademarks, etc. in this Release Notice does not entitle readers to assume that these names/designations may be used without restriction by anyone; often the names/designations are protected by law or contract, even if this is not indicated here.

openUTM (Linux and Windows systems) is the server component of the client/server package openUTM Enterprise Edition.

This package also includes:

- openUTM-Client (client/server communication),
- openUTM WebAdmin (graphical administration interfaces).

Not all products are available on all platforms.

The UPIC carrier system for UTM-Client applications with local connections (UPIC local) is part of openUTM.

In the case of openUTM clients on Unix, Linux and Windows platforms connected remotely over a network, two carrier systems are available (UPIC and OpenCPIC), each of which delivers a different functionality.

1.1 Ordering

openUTM Enterprise Edition V7.0A can be ordered from your local distributors.

openUTM Enterprise Edition V7.0A is subject to the general terms and conditions of the software product purchase, maintenance, use and service agreement.

1.1.1 Licenses

*2 The licenses are part of the contract that you entered with Fujitsu Germany GmbH when you purchased openUTM Enterprise Edition.

openUTM Enterprise Edition V7.0A is a product for which a license must be purchased. Licenses are required to use the product. These are provided as a basic license for each system or partition plus licenses for each user.

For distributed transaction processing, there are also additional licenses in 5 different variants depending on the number of parallel connections between the applications.

For development purposes, development packages are available with one basic license for each system or partition and 2 users for developing UTM applications. Licenses for additional developers can be purchased separately.

The user licenses for openUTM-Enterprise Edition include client and server users.

Licenses for UTM-Cluster applications:

There is no special cluster license for the use of UTM-Cluster functions.

You have to license openUTM on every node regardless of whether you run a stand-alone application or a node application on this computer.

1.2 Delivery

1.2.1 Software on data media

openUTM

The openUTM V7.0A files are supplied on the CD data media openUTM EED-CD V7.0. This data media contains additional software products from the openUTM product family and communication products.

openUTM WinAdmin

*2 As of openUTM (Linux and Windows systems) V7.0A35 the product Fujitsu Software openUTM
*2 WinAdmin is no longer delivered.

*1 Notice:

*1 If you are currently using openUTM WinAdmin, please switch to the product Fujitsu Software
*1 openUTM WebAdmin. openUTM WebAdmin also offers a graphical user interface for the
*1 administration of UTM applications.

1.2.2 Software for downloading

openUTM WebAdmin

The product Fujitsu Software openUTM WebAdmin V7.0A is released together with openUTM V7.0A.

openUTM WebAdmin provides an additional graphical user interface for administering UTM applications which runs as a web server in contrast to WinAdmin.

openUTM WebAdmin is available for downloading on the openUTM web site
[openUTM - Transaction Monitor : Fujitsu EMEIA](#)

XML for openUTM

*2 The software "XML for openUTM" is a free add-on to openUTM which is free of charge. Fujitsu
Germany GmbH does not accept obligation for bug-fixing.

The Software is available under
[openUTM - Transaction Monitor : Fujitsu EMEIA](#)

This component contains the Open Source GNOME XML parser, which is provided in accordance with the conditions of the MIT license. This is included in the application package ('rt') and is also available under
<http://www.opensource.org/licenses/mit-license.html>

1.3 Documentation

1.3.1 Availability

The documentation is available on the Internet:

<https://bs2manuals.ts.fujitsu.com>

The OSS documentation is also recommended for connection using the OSI-TP protocol.

1.3.2 Documentation of the Open Group interfaces

<u>Title:</u>	<u>Order number:</u>
Distributed Transaction Processing The XCPI-C Specification, Version 2 X/Open CAE Specification	ISBN 1 85912 135 7
Distributed Transaction Processing The XATMI Specification X/Open CAE Specification	ISBN 1 85912 130 6
Distributed Transaction Processing The TX (Transaction Demarcation) Specification X/Open CAE Specification	ISBN 1 85912 094 6

The X/Open manuals are available on the internet (see above). They can be obtained in printed form from bookstores quoting the ISBN number.

2 Software extensions

2.1 New functions in openUTM (Linux and Windows systems) V7.0A

2.1.1 UTM as HTTP Server

A UTM application can also act as an HTTP server.

GET, PUT, POST and DELETE are supported as methods. In addition to HTTP, access via HTTPS is also supported.

The following interfaces have been changed:

- Generation
 - KDCDEF statements BCAMAPPL:
 - Additional specification for the transport protocol for the operand T-PROT= with value SOCKET:
 - *USP: The UTM socket protocol is to be used on connections from this access point.
 - *HTTP: The HTTP protocol is to be used for connections from this access point.
 - *ANY: Both the UTM socket protocol and the HTTP protocol are supported on connections from this access point.
 - Additional specification for encryption for the operand T-PROT= with value SOCKET:
 - SECURE: On connections from this access point, communication takes place using Transport Layer Security (TLS).
 - New operand USER-AUTH = *NONE | *BASIC:
 - Herewith you can specify which authentication mechanism HTTP clients must use for the access point.
 - KDCDEF statement HTTP-DESCRIPTOR:
 - This statement defines a mapping of the path received in an HTTP request to a TAC and additional processing parameters can be specified.
- Programming
 - KDCS communication area (KB):
 - In the header of the KDCS communication area, there are new indicators for the client protocols HTTP, USP-SECURE, and HTTPS in the *kccp/KCCP* field.
 - KDCS call INIT PU:
 - The version of the interface has been increased to 7.
 - To obtain the complete available information, the value 372 must be specified in the KCLI field.
 - New fields for requesting (*KCHTTP/http_info*) and returning (*KCHTTPINF/httpInfo*) of HTTP-specific information.
- Administration interface KDCADMI
 - The data structure version of KDCADM has been changed to version 11 (field *version_data* in the parameter area).
 - New structure *kc_http_descriptor_str* in the identification area to support the HTTP descriptor.
 - New structure *kc_character_set_str* in the identification area for supporting the HTTP character set.
 - New fields *secure_soc* and *user_auth* in structure *kc_bcamappl_str* for the support of HTTP access points.
- UTM-HTTP program interface
 - In addition to the KDCS interface, UTM provides an interface for reading and writing HTTP protocol information and handling the HTTP message body.
 - The functions of the interface are briefly listed below:
 - Function *kcHttpGetHeaderByIndex()*
 - This function returns the name and value of the HTTP header field for the specified index.

- Function *kcHttpGetHeaderByName()*
This function returns the value of the HTTP header field specified by the name.
 - Function *kcHttpGetHeaderByCount()*
This function returns the number of header fields contained in the HTTP request that can be read by the program unit.
 - Function *kcHttpGetMethod()*
This function returns the HTTP method of the HTTP request.
 - Function *kcHttpGetMputMsg()*
This function returns the MPUT message generated by the program unit.
 - Function *kcHttpGetPath()*
This function returns the HTTP path of the HTTP request normalized with `KC_HTTP_NORM_UNRESERVED`.
 - Function *kcHttpGetQuery()*
This function returns the HTTP query of the HTTP request normalized with `KC_HTTP_NORM_UNRESERVED`.
 - Function *kcHttpGetRc2String()*
Help function to convert a function result of type enum into a printable zero terminated string.
 - Function *kcHttpGetReqMsgBody()*
This function returns the message body of the HTTP request.
 - Function *kcHttpGetScheme()*
This function returns the schema of the HTTP request.
 - Function *kcHttpGetVersion()*
This function returns the version of the HTTP request.
 - Function *kcHttpPercentDecode()*
Function to convert characters in percent representation in strings to their normal one-character representation.
 - Function *kcHttpPutHeader()*
This function passes an HTTP header for the HTTP response.
 - Function *kcHttpPutMgetMsg()*
This function passes a message for the program unit, which can be read with MGET.
 - Function *kcHttpPutRspMsgBody()*
This function passes a message for the message body of the HTTP response.
 - Function *kcHttpPutStatus()*
This function passes an HTTP status code for the HTTP response.
- Communication via Secure Socket Layer (SSL)
Another network process of the type *utmnetssl* is available for secure access with TLS. If BCAMAPPL is generated with `T-PROT=(SOCKET,...,SECURE)` for a UTM application, a number of *utmnetssl* processes are started when UTM is started. The number of these processes depends on the value LISTENER-ID of these BCAMAPPL objects. All TLS communication for the assigned BCAMAPPL port numbers is handled in a *utmnetssl* process.

2.1.2 Encryption

The encryption functionality in UTM between a UTM application and a UPIC client has been revised. Security gaps have been closed, modern methods have been adopted and delivery has been simplified as follows:

- UTM-Crypt variant
Previously, the encryption functionality in UTM was only available if the product UTM-CRYPT had been installed. With openUTM V7.0 this is no longer necessary. As of this version, the decision as to whether or not to use the encryption functionality is made via generation or at the time of application start.

- Security
A vulnerability has been fixed in the communication between a UTM application and a UPIC client.
Caution: This means that encrypted communication with a UTM application V7.0 is only possible together with UPIC client applications as of UPIC V7.0!
- Encryption Level 5
KDCDEF statements PTERM, TAC and TPOOL
The operand ENCRYPTION-LEVEL has an additional level 5, where the Diffie-Hellman method based on Elliptic Curves is used to agree the session key and input/output messages are encrypted with the AES-GCM algorithm.

2.1.3 Subnets

When generating subnets name resolution can be controlled via DNS.

The following interfaces have been changed for this purpose:

- Generation
 - KDCDEF statement SUBNET:
RESOLVE-NAMES=YES/NO can be used to specify whether or not a name resolution via DNS is to take place after a connection is established.
If name resolution takes place, the real processor name of the communication partner is displayed via the administration interface and in messages. Otherwise, the IP address of the communication partner and the name of the subnet defined in the generation are displayed instead of the processor name.
- Administration interface KDCADMI
The structures *kc_subnet_str* and *kc_tpool_str* contain a new field *resolve_names*.

2.1.4 Access data for the XA database connection

A modified but not yet activated user name for the XA database connection can be read by Administration (KDCADMI):

- Operation code KC_GET_OBJECT:
Data structure *kc_db_info_str*. New field *db_new_userid*.

2.1.5 Reconnect for the XA database connection

If an XA action to control the transaction detects that the connection to the database has been lost, the system tries to renew the connection and repeat the XA action.

Only if this is not successful, the affected UTM process and the UTM application are terminated abnormally. Previously, the UTM application was terminated abnormally, if a XA-Connection was lost without trying to reconnect.

2.1.6 Other changes

- XA messages
The messages regarding the XA interface were extended by the inserts UTM-Userid and TAC. The messages K204-K207, K212-K215 and K217-K218 are affected.
- UTM-Tool KDCEVAL
In the TRACE 2 record of KDCEVAL the type of the last order (bourse announcement) was recorded in the WAITEND record (first two bytes can be printed).

*1 2.2 New functions in openUTM (Linux and Windows systems) V7.0A23

- *1 • Replacement of PCMX by own implementation
*1 As of openUTM (Linux and Windows systems) V7.0A23 for network connections no longer the
*1 CMX library is used, but an own UTM implementation.
*1 This means that the software package PCMX will no longer be necessary for running UTM
*1 applications on Linux and Windows systems.
- *1 • Up to 20,000 connections per UTM net process (Linux systems)
*1 The maximum number of connections per UTM net process has been increased to 20.000 on
*1 Linux systems. On Windows systems the restriction of 1000 connections per UTM net process
remains.
- *1 • Support of Micro Focus COBOL compiler V5, V6, V7
*1 With openUTM (Linux and Windows systems) V7.0A23 the versions 5, 6 and 7 of the Micro
*1 Focus COBOL compiler are supported.
- *1 • Custom error handling
*1 After you have requested the exchange of the utmwork process with
*1 KCX_SET_RELOAD_FLAG you can end the transaction with all PGWT calls or the PEND calls
*1 KP, RE, FI, FC, RS, ER and FR as of openUTM (Linux and Windows systems) V7.0A23. The
*1 PEND PA/PR and SP calls are not allowed in this situation.
- *1 • Performance optimization
*1 Several changes have been made to increase the performance, especially on Linux systems.
- *1 • New message K523
*1 The utility program KDCDEF performs a check on the LISTENER-PORT / LISTENER-ID tuple.
*1 In case of an error the message K523 is printed, and the execution is terminated with an error.
- *1 • New message U321
*1 If UTM rejects a connection request because the transport selector (TSEL) does not match the
*1 port number / TSEL tuple configured in UTM, the message U321 is printed.
- *1 • New message K078 SSL 03
*1 The user will be informed with a message during KDCDEF run or application start if a
*1 deprecated OpenSSL version < V1.1.1 is used. The message "K078 SSL 03 OpenSSL version
*1 is deprecated" is printed to stderr.

2.3 New functions in openUTM (Linux and Windows systems) V7.0A35

- *2 • Support of Rocket Visual COBOL V8, V9 and V10
*2 With openUTM (Linux and Windows systems) V7.0A35 the versions 8, 9 and 10 of the Rocket
*2 Visual COBOL compiler are supported.
- *2 • Message destination OSXLOG
*2 With openUTM (Linux and Windows systems) V7.0A35 there is the possibility to define the linux
*2 system logfile SYSLOG as message destination. For detailed information please contact your
*2 local Fujitsu regional office.
- *2 • Bug fixes
*2 openUTM (Linux and Windows systems) V7.0A35 contains several bug fixes.

2.4 Implemented change requests (CR)

None

3 Technical information

3.1 Resource requirements

Hard disk storage requirements for installation:

<u>openUTM:</u>		
*1	Linux	< 100 MB
	Windows	< 100 MB

The main memory requirements for an executable UTM application depends on the one hand on the types of products used, such as language connections, databases, X/Open interface and distributed processing, and on the other hand on the number and extent of the application programs.

The dynamic memory requirements of the UTM resources for process communication (shared memory) depend on the extent and parameterization of the KDCDEF generation.

3.2 Software configurations

*2	Linux (SuSE)	64 bit	as of SLES 15.6
*2	Linux (RedHat)	64 bit	as of RHEL 8.10
	further Unix systems on demand		
*2	Windows	64 bit	as of Windows 11
*2	Windows Server	64 bit	as of Windows Server 2022

UTM-Cluster applications are supported on platform Linux. On Windows systems the general release of UTM-Cluster applications isn't freed.

*1	The node applications of a UTM-Cluster application must run on identical platforms.		
	NFS V4 is needed for access to the global cluster files of openUTM-Cluster applications.		

In order to use the openSM2 function the following are valid:

openSM2 V9

The following versions are supported for distributed transaction processing with UTM applications:

*1	openUTM (BS2000)	as of V7.0A
*1	openUTM (Unix, Linux, Windows systems)	as of V7.0A

*2	If problems occur with connections with UTM versions less than V7.0 that Fujitsu Germany GmbH is no longer obliged to maintain, the user is not entitled to warranty or correction claims. In this case we recommend that you update the partner application to version of openUTM equal or higher than V7.0.	
*1		

Alternatively, the following products can be used for communication with IBM-SNA systems via LU6.2:

openUTM-LU62 (Unix, Linux and Windows systems) as of V5.1
and, depending on the operating system, the following third-party products:
Linux: IBM Communications Server for Linux as of version 6.4
Windows: IBM Communications Server for Windows, as of version 6.4

*2	The following versions are supported for distributed transaction processing with Jakarta EE applications:	
*2	BeanConnect	as of V7.0A

For client/server communication optionally:

*1	openUTM-Client (Unix, Linux and Windows systems)	as of V7.0A
*1	(contains carrier systems UPIC V7.0; OpenCPIC V4.0)	
*1	openUTM-Client (BS2000)	as of V7.0A

*2 BeanConnect (Unix, Linux and Windows systems) as of V7.0A
(contains the component openUTM JConnect)

Caution: Encrypted communication with a UTM application V7.0 is only possible together with UPIC client applications as of UPIC V7.0!

For administering UTM applications via graphical interfaces:
openUTM WebAdmin V7.0A

*1 Notice:

*1 With Fujitsu Software openUTM (Linux and Windows systems) V7.0A23 openUTM WinAdmin will
*1 be supported **for the last time**.

A C/C++ compiler is required for all Linux and Windows systems, for Windows Visual Studio 2010 or higher.

For the use of the QuickStart Kit on Windows Visual Studio 2010 or higher is required.

COBOL-Compiler:

*2 Development of COBOL UTM applications:
Rocket Visual COBOL V8.0, V9.0, V10.0

For runtime:

COBOL runtime licenses from Rocket

Note: The introduction of UTM system processes can result in problems with COBOL runtime licenses if the calculation is too exact.

When calculating the required number of licenses, a maximum of three additional licenses must be taken into account.

*2 To use the encryption functionality on Linux and Windows systems, OpenSSL has to be provided
*2 by the customer:

*2 OpenSSL as of V3

On UNIX, Linux and Windows systems, the following database system can be used:

*1 ORACLE as of V19c

Note:

The software configuration includes some products that are still technically usable but for which later versions are available. You should always try and use the latest versions that are available.

3.3 Product installation

3.3.1 Installation on Linux platforms

*2 Before installing the software for openUTM please import the OpenSEAS key file with the
*2 following command:

*2 `rpm --import openseas-buildkey.key`

*2 You will find the key openseas-buildkey.key on the CD in the directory OpenSEAS.

On **Linux** use the standard procedure „rpm“:

`rpm -i <CD>/<package>.rpm --nodeps [--ignorearch] [--replacefiles]
[--prefix=<install-dir>]`

For installing OUTM-LU62, please read the corresponding readme file.

*1 **ATTENTION:** With this Version the use of the 32-bit variant is no longer supported by
*1 Fujitsu.

3.3.2 Installation on Windows platforms

*1 For the installation, please change to the product directory and run "utm-64.msi" under
*1 administration rights.

The recommended installation sequence is as follows:

- *1 1. Microsoft Visual C++
- *1 2. (optional) Cobol Compiler
- *1 3. openUTM

*1 openUTM must not be installed into a directory which contains spaces, e.g. "Program Files",
another directory has to be used.

The length of the installation path must not be longer than 39 characters.

*1 **ATTENTION:** With this Version the use of the 32-bit variant is no longer supported by
*1 Fujitsu.

3.3.3 Installation together with C++ environment on Linux platforms

When using C++ subprograms make sure that the C++ development environment of openUTM can be activated with the installation.

The following applies for the C++ Compiler:

Linux: type g++ must return as result the C++ Compiler

mainutmCC.o is then created during installation of openUTM.

mainutmCC.o can also be created after installation of openUTM (if required) by making the following entries:

*1 `<utmpath>/shsc/CCmainutm`

For further details, see the manual "Using openUTM Applications".

3.3.4 Installation logs

During installation on Linux systems, a file with the name <utmpath>/utm.log is created. This file stores the information output to the terminal during installation.

3.3.5 Notes: Kernel parameters for openUTM

It may be necessary to increase the value of some of the Linux kernel parameters in order to run a UTM application. This affects the following kernel parameters (if more than one kernel parameter is specified per line, all the specified parameters must be changed to the same value):

- MAXUP Maximum number of processes per user
It is always possible that a process bottleneck can arise for individual users (error message: fork failed) if they have to start a very large number of processes. In this event, MAXUP must be adjusted.
- SEMMNS, SEMMNU Maximum number of semaphores and semaphore structures
The maximum number of semaphores required can be calculated using the description of MAX SEMKEY in the "Generating Applications" manual.
*1 You must take into account the fact that other products may be installed on the computer that also require semaphores.
- SEMMAP, SEMMNI Maximum number of semaphore identifiers
The semaphores are grouped to form semaphore fields and a (unique) semaphore identifier is assigned to each semaphore field.
*1 openUTM groups 20 semaphores to form one semaphore field.
- SFSZLIM, HFSZLIM Maximum size of a file (in bytes)
Values as of 48 MB (0x3000000) are generally sufficient even for large UTM dumps.

The entry `ULIMIT=...` in the file `/etc/default/login` must also be adjusted to match the new value. `ULIMIT` specifies the maximum file size (for normal users) in blocks (1 block = 512 bytes).

- `SDATLIM`, `HDATLIM` Maximum swap space per process (in bytes)
 - `SVMMLIM`, `HVMMLIM` Maximum virtual memory per process (in bytes)
- In order to allow larger UTM applications to be linked with COBOL program units, these values must be increased to at least 48 MB(0x3000000). When doing this, you must make sure that the swap area of the system is \geq 48 MB.

It is not generally necessary to change the kernel parameters. Please refer to the documentation from your Linux distributor for details on the files in which the values can be changed and on how to generate a new kernel.

3.4 Product use

3.4.1 Notes for upgrading to openUTM V7.0A

Perform the following steps when upgrading from an earlier version

- Generate `KDCFILE` and `KDCROOT` with the `KDCDEF` of V7.0A
- Compile the `ROOT` source
- Compile the `MSGTAC` program unit if present
- If a separate message module is used, the source for the message module must first be newly generated with `KDCMMOD` and then compiled again
- *1 - Remove the library `libutmcrpt` from the link instruction
- Link the application program
- *1 - Transfer the user data with the `KDCUPD` to the new `KDCFILE` if the data is relevant

`KDCUPD` supports the following version upgrades:

openUTM (Linux and Windows systems) V6.4A to openUTM (Linux and Windows systems) V7.0A

openUTM (Linux and Windows systems) V6.5A to openUTM (Linux and Windows systems) V7.0A

openUTM (Linux and Windows systems) V7.0A to openUTM (Linux and Windows systems) V7.0A

Note:

A transfer of the user data with `KDCUPD` from 32-bit mode to 64-bit mode is only possible without an additional version transition, i.e. only within the current version.

*1 3.4.2 Upgrading from openUTM (Linux and Windows systems) V7.0A00 to V7.0A23

*1 openUTM V7.0A23 is fully compatible to openUTM V7.0A00. UTM applications and `KDCFILE`
 *1 from V7.0A00 can be used in V7.0A23 without any changes. Users who want to use all
 *1 corrections must generate all application components again.

*1 Some fixes take effect only after relinking the application.

*2 3.4.3 Upgrading from openUTM V7.0A23 to openUTM V7.0A35

*2 openUTM V7.0A35 is compatible with openUTM V7.0A23. UTM applications and `KDCFILE` that
 *2 were used in V7.0A23 can also be used without changes in V7.0A35. Users, who want to use all
 *2 corrections, must generate all application components again.

*2 3.4.4 Use of `kdcdef` utility under RHEL 10

*2 To use the tool `kdcdef` under RHEL 10, an additional symbolic link is required.

*2 Please create the following symbolic link in `/usr/lib64`:

*2 `ln -s /usr/lib64/libcrypt.so.2 libcrypt.so.1`

3.4.5 Converting a stand-alone application to a UTM-Cluster application

The following conversions are possible with a UTM-Cluster application:

- Conversion of a stand-alone application V7.0A to a UTM-Cluster application V7.0A
 - Conversion of a UTM-Cluster application V7.0A to a stand-alone UTM application V7.0A
- For detailed description see the manual "Using under Unix and Windows Systems".

3.4.6 Client programs with a local connection to openUTM

Client programs with a local connection to openUTM must be recompiled and linked again.

3.4.7 Linking UTM OSI-TP applications on Linux platforms

The following sequence must be adhered to when specifying the OSS libraries:

The library libossutm.so must be specified first followed by the library liboss.so.

If the sequence is not adhered to, the following startup error occurs:

```
P001 Error on OSS call (o_create() ·): - 1, 300, 199, 0
K060 Application run aborted; reason = XINI06
```

3.4.8 Linking and running UTM applications on Linux systems

The environment variable LD_LIBRARY_PATH or LD_LIBRARY_PATH_64 must be set to <utmpath>/sys when linking and running a UTM application on Linux systems.

3.4.9 Notes on a sample application for UTM-Cluster applications

The following must be taken into account when installing a sample application for a UTM node:

- For a sample application the conversion of an installed UTM node application into a stand-alone application is not supported.
- A filebase directory for a node application and a filebase directory for a stand-alone application must not be created in the file tree under the same superordinate directory.

3.5 Discontinued functions and those to be discontinued

3.5.1 Discontinued functions

Utility Program KDCDEF

Several functions have been deleted and can no longer be generated in KDCDEF. If they are still specified, this will be rejected with a syntax error in the KDCDEF run.

The following has been deleted:

- TNS operation:
When starting a UTM application, the TNS generation is no longer read. The addressing information must be stored completely during configuration with KDCDEF.
- KDCDEF statement PTERM
Operand values 1 and 2 for *ENCRYPTION-LEVEL*
- KDCDEF statement TPOOL
Operand values 1 and 2 for *ENCRYPTION-LEVEL*
- KDCDEF statement TAC
Operand value 1 for *ENCRYPTION-LEVEL*

*1 NetCOBOL

*1 With openUTM V7.0 the COBOL compiler NetCOBOL is for technical reasons no longer supported.

3.5.2 Functions to be discontinued

- *1 • Fujitsu Software openUTM WinAdmin
- *1 With this version openUTM WinAdmin will be supported for the last time.

3.6 Incompatibilities

- **Compilation of ROOT C SOURCE**
The needed C Header files for the compilation of the ROOT C Sources have been changed in openUTM V7.0.
Because the ROOT Source is created completely by the utility program KDCDEF this adjustment takes place automatically.
However it can be that in the compiling procedure (e.g. makefile) C Header files are specified as dependence, which will no more available as of UTM V7.0. These dependences have to be deleted for the upgrade to UTM V7.0.

- **Returned length KCRLM at MGET with KCRCCC 02Z at USP Clients**
If a program unit is reading a message part with MGET and the length in parameter KCLA is smaller than the actual length of the (remaining) message part, the KDCS return code 02Z is set and the actual length of the (remaining) message part is returned in the return field KCRLM now and not the value of KCLA.

Example at a message part with length 20:
MGET KCLA=5 -> 02Z, KCRLM=20 (instead of KCRLM=5)
MGET KCLA=7 -> 02Z, KCRLM=15 (instead of KCRLM=7)
MGET KCLA=8 -> 000, KCRLM=8

- *1 • **Utility program utmwork**
- *1 As of openUTM (Linux and Windows systems) V7.0A, the UTM library libutmcrpt is no longer
- *1 required due to the elimination if the UTM-Crypt variant. Please remove this library from your
- *1 link instruction.

3.7 Restrictions

Notes on using the sample application:

No warranty is offered for the sample application and no service commitment is entered into.

3.8 Procedure in the event of errors

The following documentation is required for diagnosis if an error occurs:

- Detailed description of the error situation
- Versions/correction statuses of the software involved
- Exact details on the computer type

All error documentation should be as complete as possible. The following error documentation is required:

- UTM dumps for all work processes and the associated gcores. These files should be available as binary files, i.e. not in print-edited form
- SYSLOG file(s)
- stdout/stderr logs of the UTM processes
- KDCDEF generation, start procedure and start parameter
- core files with the associated phases (utmwork) and shared objects. The shared objects can be determined using "ldd utmwork"

- Linker and compiler lists

Also, in the event of errors in a UTM-Cluster application:

- all Cluster global files, logs, and DUMPS of all node applications
- In case of problems that were caused by the interaction of the node applications the log files of all other node applications
- The start procedure and the procedures specified during generations as EMERGENCY-CMD and FAILURE-CMD

Procedure in the event of errors:

- Analyze the PENDER dump
- Reproduce the error using a suitable debugger such as dbx, gdb
- Determine the call hierarchy for the core using a debugger

An attempt should be made to reproduce the errors using static libraries.

If the application terminates abnormally, the UTM dumps and gcores of all processes must be saved. If possible, the termination should be reproduced with TESTMODE=ON (start parameter) while TRACEREC and IPTRACE should be set to minimum 10000.

*2

Termination with signals:

If a PENDER dump occurred with 70Z/XT10 or XT11 or if the application terminated with SIG010/SIG011 (SIGBUS/SIGSEGV signal), UTM signal handling should be deactivated with the start parameter START STXIT=OFF so that a core dump will immediately be generated when the faulty command is issued.

Before the application is restarted again, it is absolutely necessary to call kdcrem in order to delete or reset any semaphores and shared memories which may still be present as well as any status information on the application in the *<utmpath>/applifile*.

The following additional documentation can also be created in the event of errors relating to the openUTM network connection:

- Messages from the UTM network processes in stdout/stderr
- OSS traces
- UTM-BCAM trace
- In the case of heterogeneous connections, generation information and trace from openUTM LU6.2 and its SNA components and from the IBM SNA system

*1

For details on creating the UTM-BCAM traces and the OSS traces, refer to the manual "Messages, Debugging and Diagnostics".

The following additional documentation can also be created in the event of errors relating to databases:

- XA debug trace
- For further documentation, see the relevant Release Notice for the database system

Following startup errors (message K049), for instance with insert 40 / 32, the UTM tool kdcrem must be called before the program is restarted.

*1 **3.9 Additions to the User Guide**

*1 **3.9.1 openUTM Messages, Debugging and Diagnostics**

*1 With the release of Fujitsu Software openUTM (Linux and Windows systems) V7.0A23, the
 *1 manual “openUTM Messages, Debugging and Diagnostics on Unix, Linux and Windows Systems”
 *1 has also been updated.

*1 The new edition is available online at

*1 <https://bs2manuals.ts.fujitsu.com>

*1 **3.9.2 openUTM Administering Applications**

*1 **Chapter 11.3.1.25 kc_pterm_str – Clients and printers**

*1 t_prot

*1 ...
 *1 'H' native TCP/IP transport protocol for communication with Hypertext Transfer Protocol
 *1 (HTTP) via the socket interface (SOCKET)

*1 ...

*2 **3.9.3 Additional inserts in U315**

*2 There are several new inserts in message U315 as of openUTM V7.0A35.

8	openUTM net process decided to clear down the connection because of an error situation. For details see previous U message.
9	openUTM net process rejected a request to establish a connection because of an error situation. For details see previous U message.
17	Partner rejected a request to establish a connection

4 Hardware support

Support is provided for the hardware, on which the mentioned operating system versions can run. This includes all systems based on Intel x86 technology, such as laptops, PCs, PRIMERGY systems.

Other platforms on request.

CPU at least 250 MHz, RAM at least 512 MB

UTM-Cluster applications:

- A network file system/service (NFS) is necessary for the operation of UTM-Cluster applications.
- Nodes of a UTM-Cluster application can be systems with different Linux versions. It is not possible to use nodes with other operating systems (other Linux distributions).

*1