

Fujitsu Software openUTM Enterprise Edition (Linux- und Windows-Systeme)

Version 7.0A

Oktober 2025



Freigabemitteilung

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

*2 Copyright © 2025 Fujitsu

Die Marke Fujitsu und das Fujitsu Logo sind registrierte Marken der Fujitsu Limited, Japan in Europa und in anderen Ländern.

BS2000 ist eine Marke der Fujitsu Germany GmbH in Europa.

*2 Oracle™ und Java™ sind Warenzeichen oder eingetragene Warenzeichen der Oracle Corporation, Redwood Shores, USA.

*2 Windows™ ist ein Warenzeichen oder eingetragenes Warenzeichen der Microsoft Corporation, Redmond, USA

*2 IBM™, CICST™ und z/OS™ sind Warenzeichen oder eingetragene Warenzeichen der International Business Machines Corporation, USA.

*2 Linux® ist ein eingetragenes Warenzeichen von Linus Torvalds.

1	Allgemeines	1
1.1	Bestellung	2
1.1.1	Nutzungsrechte	2
1.2	Auslieferung	3
1.2.1	Software auf Datenträger	3
1.2.2	Software zum Download	3
1.3	Dokumentation	4
1.3.1	Verfügbarkeit	4
1.3.2	Dokumentation der Open Group-Schnittstellen	4
2	Software-Erweiterungen	5
2.1	Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A	5
2.1.1	UTM als HTTP-Server	5
2.1.2	Verschlüsselung	7
2.1.3	Subnetze	7
2.1.4	Zugangsdaten für den XA-Datenbank-Anschluss	7
2.1.5	Reconnect für den XA-Datenbank-Anschluss	7
2.1.6	Sonstige Änderungen	8
2.2	Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A23	8
2.3	Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A35	9
2.4	Realisierte Change-Requests (CR)	9
3	Technische Hinweise	10
3.1	Ressourcenbedarf	10
3.2	SW-Konfigurationen	10
3.3	Produkt-Installation	11
3.3.1	Installation auf Linux-Plattformen	11
3.3.2	Installation auf Windows-Plattformen	12
3.3.3	Installation zusammen mit C++ Umgebung auf Linux-Plattformen	12
3.3.4	Installationsprotokoll	12
3.3.5	Hinweise: Kernelparameter für openUTM	12
3.4	Produkt-Einsatz	13
3.4.1	Hinweise zur Umstellung auf openUTM 7.0A	13
3.4.2	Hinweise zur Umstellung von openUTM (Linux, Windows-Systeme) V7.0A00 auf V7.0A23	13
3.4.3	Hinweise zur Umstellung von openUTM V7.0A23 auf openUTM V7.0A35	14
3.4.4	Einsatz des Tools kdcdef unter RHEL 10	14
3.4.5	Umstellung einer stand-alone Anwendung in eine UTM-Cluster-Anwendung	14
3.4.6	Client-Programme mit lokaler Anbindung an openUTM	14
3.4.7	Binden von UTM OSI-TP Anwendungen auf Linux-Plattformen	14
3.4.8	Binden und Starten von UTM-Anwendungen auf Linux-Systemen	14
3.4.9	Hinweise zur Beispielanwendung für UTM-Cluster-Anwendungen	14
3.5	Entfallene und gekündigte Funktionen	15
3.5.1	Entfallene Funktionen	15
3.5.2	Gekündigte Funktionen	15
3.6	Inkompatible Änderungen	15
3.7	Einschränkungen	16
3.8	Verhalten im Fehlerfall	16
3.9	Ergänzungen zum Benutzerhandbuch	17
3.9.1	openUTM Meldungen, Test und Diagnose	17
3.9.2	openUTM Anwendungen administrieren	17
3.9.3	Zusätzliche Inserts in U315	17
4	Hardware-Unterstützung	18

1 Allgemeines

Der offene und universelle Transaktionsmonitor openUTM unterstützt die Erstellung und den Betrieb von Transaktionsanwendungen im Fujitsu Betriebssystem BS2000, auf Linux- und Windows-Systemen, sowie auf Anfrage auch auf UNIX-Systemen.

Fujitsu Software openUTM ermöglicht die Realisierung von Client-/Server-Architekturen sowie die Erstellung verteilter Transaktionsanwendungen in homogenen sowie auch heterogenen IT-Infrastrukturen (Multi-Tier-Architektur).

Im Produktumfang von openUTM sind bereits einige Tools kostenfrei enthalten:

- Das webbasierte Tool openUTM WebAdmin steht für die Administration von UTM-Anwendungen über einen Internet-Browser zur Verfügung.
- Mit der Programmschnittstelle "XML für openUTM", kurz UTM-XML, besitzt openUTM eine komfortable Schnittstelle zum Erstellen und Bearbeiten von Daten in Form von XML-Dokumenten.

openUTM kann zusätzlich mit weiteren kostenpflichtigen Produkten kombiniert werden:

- Das Produkt Fujitsu Software openUTM-Client, welches Client-Server-Kommunikation mit openUTM-Server-Anwendungen ermöglicht, steht für das Fujitsu Betriebssystem BS2000 als eigenes Produkt ebenso wie für Linux- und Windows-Systeme zur Verfügung und auf Anfrage auch für UNIX-Systeme.
- Für die Kopplung mit CICS/IMS-Anwendungen auf IBM-Systemen über SNA LU6.2 gibt es das Zusatzprodukt Fujitsu Software openUTM LU62 (Unix-, Linux- und Windows-Systeme) V5.1.
- *2 • Für die Kopplung von UTM-Anwendungen mit Jakarta EE Application Servern gibt es das Produkt Fujitsu Software BeanConnect. BeanConnect beinhaltet auch eine Komponente zur Erstellung von JAVA-Clients für openUTM.

Für alle Tools gibt es eigene Freigabemitteilungen.

Weitere Informationen zu den oben aufgeführten Produkten sowie die Datenblätter finden Sie unter:

- *1 [FUJITSU Software openUTM - Transaction Monitor : Fujitsu Deutschland](#)

openUTM ist Teil des umfassenden Angebots der Middleware-Suite openSEAS.

Die Middleware-Suite openSEAS ermöglicht die einfache Integration einzelner Anwendungen und das Zusammenwirken von vorhandenen und neuen Anwendungen. openSEAS ergänzt sich perfekt mit der Fusion Middleware unseres strategischen Partners Oracle. Oracle Fusion Middleware ist eine marktführende Familie von vorintegrierten Produkten, die von Anwendungsinfrastruktur und Entwickleranwendungen bis zu Business Intelligence reicht.

Zusammen bietet dies die Plattform für:

- Individuelle geschäftskritische IT-Lösungen
- Unternehmensportale
- Integrationslösungen für Service Orientierte Architekturen
- Web-Anwendungen und Portale
- Geschäftsprozesse

Diese Freigabemitteilung enthält in Kurzform die wesentlichen Erweiterungen, Abhängigkeiten und Betriebshinweise zu openUTM (Linux- und Windows-Systeme) V7.0A sowie alle nötigen Informationen für den Umstieg von der Vorgängerversion openUTM (Linux- und Windows-Systeme) V6.5A.

Nachträglich bekannt gewordene Änderungen werden in dieser Datei aktualisiert und nachgeliefert.

*2 Der Inhalt entspricht dem Freigabestand: Oktober 2025.

*1 Änderungen gegenüber openUTM V7.0A vom November 2019 sind mit *1 gekennzeichnet.

*2 Änderungen gegenüber openUTM V7.0A23 vom Juni 2022 sind mit *2 gekennzeichnet.

*2 Der Inhalt wurde gegenüber dem Freigabestand V7.0A23 modifiziert.

*2 Er entspricht dem Freigabestand V7.0A35: Oktober 2025

Diese Freigabemitteilung ist, wie alle Freigabemitteilungen, über das Internet verfügbar, und zwar unter der URL:

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

Werden mit dem Einsatz der vorliegenden Produktversion eine oder mehrere Vorgängerversionen übersprungen, so sind auch die Hinweise aus den Freigabemitteilungen (bzw. readme-Dateien) der Vorgängerversionen zu berücksichtigen.

Die Wiedergabe von Namen, Warenbezeichnungen und dgl. in dieser Freigabemitteilung berechtigt nicht zu der Annahme, dass diese Namen/Bezeichnungen ohne weiteres von jedermann benutzt werden dürfen; oft handelt es sich um gesetzlich oder vertraglich geschützte Namen/Bezeichnungen, auch wenn sie nicht als solche gekennzeichnet sind.

openUTM (Linux- und Windows-Systeme) ist die Server-Komponente in dem Client/Server-Paket openUTM Enterprise Edition.

In diesem Paket sind ferner enthalten:

- openUTM-Client (Client/Server-Kommunikation),
- openUTM WebAdmin (grafische Administrationsoberflächen).

Auf den verschiedenen Plattformen sind nicht immer alle Produkte verfügbar.

Das Trägersystem UPIC für UTM-Client-Anwendungen mit lokaler Anbindung (UPIC-Lokal) ist Bestandteil von openUTM.

Für openUTM-Clients auf Unix-, Linux- und Windows-Plattformen, die remote über ein Netzwerk angeschlossen sind, stehen zwei Trägersysteme (UPIC und OpenCPIC) mit unterschiedlicher Mächtigkeit zur Auswahl.

1.1 Bestellung

openUTM Enterprise Edition V7.0A kann über die für Sie zuständige Vertriebsgesellschaft bezogen werden.

Für openUTM Enterprise Edition V7.0A gelten die allgemeinen Bedingungen zum Kauf-, Wartungs-, Softwareüberlassungs- und Softwarebetreuungsvertrag.

1.1.1 Nutzungsrechte

*2 Die Nutzungsrechte sind Bestandteil des Vertrages, den Sie beim Bezug von openUTM Enterprise Edition mit Fujitsu Germany GmbH abgeschlossen haben.

openUTM Enterprise Edition V7.0A ist ein Lizenzprodukt. Für den Einsatz sind Nutzungsrechte notwendig. Sie werden als Basis-Nutzungsrecht pro System bzw. Partition plus Nutzungsrecht pro User angeboten.

Für Verteilte Transaktionsverarbeitung gibt es zusätzlich Nutzungsrechte in 5 verschiedenen Ausprägungen je nach Anzahl paralleler Verbindungen zwischen den Anwendungen.

Zur Entwicklung gibt es Entwicklungspakete mit dem Basisnutzungsrecht je System bzw. Partition und je 2 Usern für die Entwicklung von UTM-Anwendungen. Nutzungsrechte für weitere Entwickler können einzeln bezogen werden.

Die User-Nutzungsrechte von openUTM-Enterprise Edition beinhalten Client- und Server-User. Nutzungsrechte für UTM-Cluster-Anwendungen:
Für die Nutzung der UTM Cluster Funktionalität gibt es keine spezielle Cluster Lizenz. Sie müssen openUTM auf jedem Knoten eigens lizenzieren, unabhängig davon, ob Sie auf diesem Rechner eine stand-alone Anwendung oder eine Knoten-Anwendung betreiben.

1.2 Auslieferung

1.2.1 Software auf Datenträger

openUTM

Die Auslieferung der Dateien zu openUTM V7.0A erfolgt auf dem CD-Datenträger openUTM EED-CD V7.0.

Auf diesem Datenträger sind weitere Softwareprodukte der openUTM-Produktfamilie sowie für die Kommunikation gespeichert.

openUTM WinAdmin

*2 Ab openUTM (Linux- und Windows-Systeme) V7.0A35 wird das Produkt Fujitsu Software
*2 openUTM WinAdmin nicht mehr ausgeliefert.

*1 Hinweis:

*1 Sollte bei Ihnen derzeit openUTM WinAdmin im Einsatz sein, wechseln Sie bitte auf das Produkt
*1 Fujitsu Software openUTM WebAdmin. openUTM WebAdmin bietet Ihnen ebenfalls eine
*1 grafische Oberfläche zur Administration von UTM-Anwendungen.

1.2.2 Software zum Download

openUTM WebAdmin

Zusammen mit openUTM V7.0A wird zusätzlich das Produkt Fujitsu Software openUTM WebAdmin V7.0A freigegeben.

openUTM WebAdmin bietet eine weitere grafische Oberfläche zur Administration von UTM-Anwendungen, die im Unterschied zu WinAdmin auf einem Web-Server abläuft.

*1 openUTM WebAdmin wird auf der openUTM Website
[FUJITSU Software openUTM - Transaction Monitor : Fujitsu Deutschland](#)
zum Download zur Verfügung gestellt.

XML für openUTM

*2 Die Software "XML für openUTM" ist ein kostenloses Zusatzprodukt zu openUTM, für das Fujitsu
*2 Germany GmbH keine Fehlerbehebung übernimmt.

*1 Die Software ist beziehbar über
[FUJITSU Software openUTM - Transaction Monitor : Fujitsu Deutschland](#)

Diese Komponente enthält den OpenSource GNOME XML-Parser, der zu den Bedingungen der MIT Lizenz bereitgestellt wird. Diese wird im Anwendungspaket ('rt') mit ausgeliefert und ist auch im Web unter

<http://www.opensource.org/licenses/mit-license.html>
zu finden.

1.3 Dokumentation

1.3.1 Verfügbarkeit

Die Dokumentation ist im Internet unter <https://bs2manuals.ts.fujitsu.com> verfügbar.

Für die Kopplung über das OSI-TP-Protokoll wird zusätzlich die Dokumentation zu OSS und CMX empfohlen.

1.3.2 Dokumentation der Open Group-Schnittstellen

<u>Titel:</u>	<u>Bestellnummer:</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

Die X/Open-Handbücher sind im Internet (siehe oben) verfügbar.
Sie können gedruckt im Buchhandel unter der ISBN bezogen werden.

2 Software-Erweiterungen

2.1 Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A

2.1.1 UTM als HTTP-Server

Eine UTM Anwendung kann auch als HTTP-Server fungieren.

Als Methoden werden GET, PUT, POST und DELETE unterstützt. Neben HTTP wird auch der Zugang über HTTPS unterstützt.

Dazu wurden folgende Schnittstellen geändert:

- Generierung
 - KDCDEF-Anweisung BCAMAPPL:
 - Beim Operand T-PROT= mit Wert SOCKET gibt es eine zusätzliche Angabe zum Transportprotokoll:
 - *USP: Auf Verbindungen dieses Zugriffspunktes soll das UTM-Socket-Protokoll verwendet werden.
 - *HTTP: Auf Verbindungen dieses Zugriffspunktes soll das HTTP-Protokoll verwendet werden.
 - *ANY: Auf Verbindungen dieses Zugriffspunktes werden sowohl das UTM-Socket-Protokoll als auch das HTTP-Protokoll unterstützt.
 - Beim Operand T-PROT= mit Wert SOCKET gibt es zusätzlich die Angabe zur Verschlüsselung:
 - SECURE: Auf Verbindungen dieses Zugriffspunktes erfolgt die Kommunikation unter Verwendung von Transport Layer Security (TLS).
 - Neuer Operand USER-AUTH = *NONE | *BASIC:
 - Hiermit kann angegeben werden, welchen Authentisierungsmechanismus HTTP-Clients für diesen Zugangspunkt verwenden müssen.
 - KDCDEF-Anweisung HTTP-DESCRIPTOR:
 - Mit dieser Anweisung wird eine Abbildung des in einem HTTP-Request empfangenen Path auf einen TAC definiert und es können zusätzliche Verarbeitungsparameter angegeben werden.
- Programmierung
 - KDCS-Kommunikationsbereich (KB):
 - Im Kopf des KDCS-Kommunikationsbereichs gibt es im Feld *kccp/KCCP* neue Werte für die Client-Protokolle HTTP, USP-SECURE und HTTPS.
 - KDCS-Aufruf INIT PU:
 - Die Version der Schnittstelle wurde auf 7 erhöht.
 - Um die verfügbare Information vollständig zu erhalten, muss im Feld KCLI der Wert 372 angegeben werden.
 - Neue Felder zur Anforderung (*KCHTTP/http_info*) und Rückgabe (*KCHTTPINF/httpInfo*) von HTTP-spezifischen Informationen.
- Administrationsschnittstelle KDCADMI
 - Die Datenstrukturversion von KDCADMI wurde auf Version 11 geändert (Feld *version_data* im Parameterbereich).
 - Neue Struktur *kc_http_descriptor_str* im Identifikationsbereich für die Unterstützung des HTTP Deskriptors.
 - Neue Struktur *kc_character_set_str* im Identifikationsbereich für die Unterstützung des HTTP Charactersets.
 - Neue Felder *secure_soc* und *user_auth* in Struktur *kc_bcamappl_str* für die Unterstützung von HTTP Zugangspunkten.

- Programmschnittstelle UTM-HTTP
Zusätzlich zum KDCS-Interface bietet UTM ein Interface zum Lesen und Schreiben von HTTP Protokollinformationen und zur Behandlung des HTTP Message Body.
Im Folgenden werden die Funktionen des Interface kurz aufgelistet:
 - Funktion *kcHttpGetHeaderByIndex()*
Diese Funktion liefert den Namen und Wert des HTTP-Header-Feldes für den angegebenen Index zurück.
 - Funktion *kcHttpGetHeaderByName()*
Die Funktion liefert den Wert des über den Namen spezifizierten HTTP-Header-Feldes zurück.
 - Funktion *kcHttpGetHeaderCount()*
Diese Funktion liefert die Anzahl der in dem HTTP-Request enthaltenen Header-Felder zurück, die vom Teilprogramm gelesen werden können.
 - Funktion *kcHttpGetMethod()*
Diese Funktion liefert die HTTP-Methode des HTTP-Requests zurück.
 - Funktion *kcHttpGetMputMsg()*
Diese Funktion liefert die vom Teilprogramm erzeugte MPUT-Nachricht zurück.
 - Funktion *kcHttpGetPath()*
Diese Funktion liefert den mit KC_HTTP_NORM_UNRESERVED normierten HTTP-Path des HTTP-Requests zurück.
 - Funktion *kcHttpGetQuery()*
Diese Funktion liefert die mit KC_HTTP_NORM_UNRESERVED normierte HTTP-Query des HTTP-Requests zurück.
 - Funktion *kcHttpGetRc2String()*
Hilfsfunktion um ein Funktionsergebnis vom Typ enum in einen abdruckbaren null-terminierten String umzuwandeln.
 - Funktion *kcHttpGetReqMsgBody()*
Diese Funktion liefert den Message Body des HTTP-Requests zurück.
 - Funktion *kcHttpGetScheme()*
Diese Funktion liefert das Schema des HTTP-Requests zurück.
 - Funktion *kcHttpGetVersion()*
Diese Funktion liefert die Version des HTTP-Requests zurück.
 - Funktion *kcHttpPercentDecode()*
Funktion zur Umwandlung von Zeichen in Prozent-Darstellung in Zeichenfolgen in ihre normale Ein-Zeichen-Darstellung.
 - Funktion *kcHttpPutHeader()*
Diese Funktion übergibt einen HTTP-Header für die HTTP-Response.
 - Funktion *kcHttpPutMgetMsg()*
Diese Funktion übergibt eine Nachricht für das Teilprogramm, die mit MGET gelesen werden kann.
 - Funktion *kcHttpPutRspMsgBody()*
Diese Funktion übergibt eine Nachricht für den Message Body für die HTTP-Response.
 - Funktion *kcHttpPutStatus()*
Diese Funktion übergibt einen HTTP-Statuscode für die HTTP-Response.
- Kommunikation über den Secure Socket Layer (SSL)
Für einen sicheren Zugang über TLS steht ein weiterer Netzprozess vom Typ *utmnetssl* zur Verfügung.
Ist für eine UTM-Anwendung BCAMAPPL mit T-PROT=(SOCKET,...,SECURE) generiert, dann wird beim Anwendungsstart von UTM eine Anzahl von *utmnetssl* Prozessen gestartet. Die Anzahl dieser Prozesse ist abhängig vom Wert LISTENER-ID dieser BCAMAPPL Objekte. In einem *utmnetssl* Prozess wird für die zugeordneten BCAMAPPL Portnummern die gesamte TLS-Kommunikation abgewickelt.

2.1.2 Verschlüsselung

Die Verschlüsselungsfunktionalität in UTM zwischen einer UTM-Anwendung und einem UPIC-Client wurde überarbeitet. Dabei wurden Sicherheitslücken geschlossen, moderne Methoden aufgenommen und die Auslieferung wie folgt vereinfacht:

- UTM-CRYPT Variante
Bisher stand die Verschlüsselungsfunktionalität in UTM nur zur Verfügung, wenn man das Produkt UTM-CRYPT installiert hatte. Mit openUTM V7.0 ist dies nicht mehr erforderlich. Ab dieser Version wird über die Generierung bzw. zum Anwendungsstart entschieden, ob die Verschlüsselungsfunktionalität zum Einsatz kommt oder nicht.
- Security
Bei der Kommunikation zwischen einer UTM-Anwendung und einem UPIC-Client wurde eine Sicherheitslücke behoben.
Achtung: Das hat zur Folge, dass verschlüsselte Kommunikation einer UTM-Anwendung V7.0 nur zusammen mit UPIC-Client Anwendungen ab UPIC V7.0 möglich ist!
- Verschlüsselung Level 5
KDCDEF-Anweisungen PTERM, TAC und TPOOL
Beim Operanden ENCRYPTION-LEVEL gibt es einen zusätzlichen Level 5. Dabei wird zur Vereinbarung des Session-Keys das auf Elliptic Curves basierende Diffie-Hellman Verfahren verwendet und Ein-/Ausgabe-Nachrichten werden mit dem AES-GCM Algorithmus verschlüsselt.

2.1.3 Subnetze

Bei der Generierung von Subnetzen kann die Namensauflösung per DNS gesteuert werden. Dazu wurden folgende Schnittstellen geändert:

- Generierung
 - KDCDEF-Anweisung SUBNET:
Mit RESOLVE-NAMES=YES/NO kann angegeben werden, ob nach einem Verbindungsaufbau eine Namensauflösung per DNS stattfinden soll oder nicht.
Falls eine Namensauflösung erfolgt, dann wird über die Administrationsschnittstelle und in Meldungen der echte Prozessname des Kommunikationspartners angezeigt. Andernfalls wird als Prozessname die IP-Adresse des Kommunikationspartners sowie der Name des in der Generierung definierten Subnetzes angezeigt.
- Administrationsschnittstelle KDCADMI
Die Strukturen *kc_subnet_str* und *kc_tpool_str* enthalten ein neues Feld *resolve_names*.

2.1.4 Zugangsdaten für den XA-Datenbank-Anschluss

Ein modifizierter aber noch nicht aktivierter Benutzername für den XA-Datenbank-Anschluss kann per Administration (KDCADMI) gelesen werden:

- Operationscode KC_GET_OBJECT:
Datenstruktur *kc_db_info_str*: Neues Feld *db_new_userid*.

2.1.5 Reconnect für den XA-Datenbank-Anschluss

Wird bei einer XA Aktion zur Steuerung der Transaktion entdeckt, dass die Verbindung zur Datenbank nicht mehr besteht, wird versucht die Verbindung zu erneuern und die XA Aktion zu wiederholen.

Nur falls dies nicht erfolgreich ist, werden der betroffene UTM Prozess und die UTM-Anwendung abnormal beendet. Bisher wurde bei jedem Verbindungsverlust zur XA Datenbank unmittelbar ohne erneuten Verbindungs-Versuch die UTM-Anwendung abnormal beendet.

2.1.6 Sonstige Änderungen

- XA-Meldungen
Die Meldungen bzgl. der XA-Schnittstelle wurden jeweils um die Inserts UTM-Userid und TAC erweitert. Betroffen sind die Meldungen K204-K207, K212-K215 und K217-K218.
- UTM-Tool KDCEVAL
Im TRACE 2 Satz von KDCEVAL wurde im WAITEND Record der Typ des letzten Auftrags (Börsen-Announcements) aufgenommen (ersten beiden Bytes abdruckbar).

*1 2.2 Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A23

- *1 • Ablösung von PCMX durch eigene Implementierung
*1 Ab openUTM (Linux- und Windows-Systeme) V7.0A23 erfolgt die Netzanbindung nicht mehr
*1 mittels der CMX-Bibliothek, sondern durch eine UTM eigene Implementierung.
*1 Das heißt für den Betrieb von UTM-Anwendungen auf Linux- und Windows-Systemen ist
*1 zukünftig das Software Paket PCMX nicht mehr notwendig.
- *1 • Bis zu 20.000 Verbindungen pro UTM Netzprozess (Linux-Systeme)
*1 Die maximale Anzahl der Verbindungen pro UTM Netzprozess wurde auf Linux-Systemen auf
*1 20.000 erhöht.
*1 Auf Windows-Systemen bleibt es bei der Beschränkung auf 1000 Verbindungen pro UTM-
*1 Netzprozess.
- *1 • Unterstützung Micro Focus COBOL Compiler V5, V6, V7
*1 Mit openUTM (Linux- und Windows-Systeme) V7.0A23 werden die Versionen 5, 6 und 7 des
*1 Micro Focus COBOL Compilers unterstützt.
- *1 • Anwenderspezifische Fehlerbehandlung
*1 Nachdem Sie mit KCX_SET_RELOAD_FLAG den Austausch des utmwork-Prozesses
*1 angefordert haben, können Sie ab openUTM (Linux- und Windows-Systeme) V7.0A23 die
*1 Transaktion mit allen PGWT- bzw. den PEND-Aufrufen KP, RE, FI, FC, RS, ER und FR
*1 beenden. Die Aufrufe PEND PA/PR und SP sind in dieser Situation nicht zulässig.
- *1 • Performance Optimierung
*1 Es wurden mehrere Änderungen vorgenommen, um vor Allem auf Linux-Systemen, die
*1 Performance zu erhöhen.
- *1 • Neue Meldung K523
*1 Das Dienstprogramm KDCDEF führt eine Prüfung bzgl. des Tupels LISTENER-PORT /
*1 LISTENER-ID durch. Im Fehlerfall wird die Meldung K523 ausgegeben und der
*1 Generierungsablauf fehlerhaft beendet.
- *1 • Neue Meldung U321
*1 Falls ein Verbindungsaufbauwunsch von UTM abgelehnt wird, weil der Transportselektor
*1 (TSEL) nicht zum Tupel Portnummer / TSEL passt, der in UTM konfiguriert ist, wird die Meldung
*1 U321 ausgegeben.
- *1 • Neue Meldung K078 SSL 03
*1 Falls beim Ablauf von KDCDEF bzw. beim Start der UTM-Anwendung erkannt wird, dass eine
*1 OpenSSL Version < V1.1.1 zum Einsatz kommt, wird der Anwender darüber informiert. Es wird
*1 eine Meldung „K078 SSL 03 OpenSSL is deprecated“ nach stderr ausgegeben.

***2 2.3 Neue Funktionen in openUTM (Linux- und Windows-Systeme) V7.0A35**

- *2 • Unterstützung Rocket Visual COBOL V8, V9 und V10
*2 Mit openUTM (Linux- und Windows-Systeme) V7.0A35 werden die Versionen 8, 9 und 10 des
*2 Rocket Visual COBOL Compilers unterstützt.

- *2 • Meldungsziel OSXLOG
*2 Mit openUTM (Linux- und Windows-Systeme) V7.0A35 ist es möglich, als Meldungsziel die
*2 Linux System-Protokolldatei SYSLOG anzugeben. Für detaillierte Informationen wenden Sie
*2 sich bitte an den für Sie zuständigen Sitz der Region von Fujitsu.

- *2 • Fehlerkorrekturen
*2 Mit openUTM (Linux- und Windows-Systeme) V7.0A35 wurden einige Fehler behoben.

2.4 Realisierte Change-Requests (CR)

Keine

3 Technische Hinweise

3.1 Ressourcenbedarf

Plattenspeicherbedarf für die Installation:

openUTM:

*1	Linux	< 100 MB
	Windows	< 100 MB

Der Hauptspeicherbedarf für eine ablauffähige UTM-Anwendung ist zum einen abhängig von der Art der verwendeten Produkte, wie Sprachanschluss, Datenbank, X/Open-Schnittstelle und verteilte Verarbeitung, zum anderen von Menge und Umfang der Anwenderprogramme.

Der dynamische Speicherbedarf für UTM-Betriebsmittel zur Prozesskommunikation (Shared Memory) ist abhängig von Umfang und Parametrisierung der KDCDEF-Generierung.

3.2 SW-Konfigurationen

*2	Linux (SuSE)	64 Bit	ab SLES 15.6
*2	Linux (RedHat)	64 Bit	ab RHEL 8.10
	weitere Unix-Systeme auf Anfrage		

*2	Windows	64 Bit	ab Windows 11
*2	Windows Server	64 Bit	ab Windows Server 2022

UTM-Cluster-Anwendungen werden auf der Plattform Linux unterstützt. Auf Windows-Systemen sind UTM-Cluster-Anwendungen nicht allgemein freigegeben.

*1	Die Knoten-Anwendungen einer UTM-Cluster-Anwendung müssen auf identischen Plattformen ablaufen.		
	Für den Zugriff auf die globalen Cluster-Dateien von openUTM-Cluster-Anwendungen wird NFS V4 benötigt.		

Zur Nutzung der openSM2-Funktionalität gilt:
openSM2 V9

Für verteilte Transaktionsverarbeitung mit anderen UTM-Anwendungen werden folgende Versionen unterstützt:

*1	openUTM (BS2000)	ab V7.0A
*1	openUTM (Unix-, Linux-, Windows- Systeme)	ab V7.0A

*1	Wenn es bei der Kopplung mit Versionen kleiner V7.0 Probleme gibt, dann besteht keine	
*2	Wartungsverpflichtung seitens Fujitsu Germany GmbH; in diesem Fall sollte ein Kunde die	
*1	Partneranwendungen auf UTM-Versionen größer/gleich V7.0 umstellen.	

Für die Kommunikation mit IBM-SNA-Systemen über LU6.2 können wahlweise folgende Produkte eingesetzt werden:

	openUTM-LU62 (Unix-, Linux- und Windows-Systeme)	ab V5.1
	und abhängig vom Betriebssystem noch die Fremdprodukte:	
	Linux: IBM Communications Server für Linux,	ab Version 6.4
	Windows: IBM Communications Server für Windows,	ab Version 6.4

*2	Für verteilte Transaktionsverarbeitung mit Jakarta EE Anwendungen werden folgende Versionen	
	unterstützt:	
*2	BeanConnect	ab V7.0A

	Für Client-Server-Kommunikation wahlweise:	
*1	openUTM-Client (Unix-, Linux- und Windows-Systeme)	ab V7.0A
	(enthält Trägersysteme UPIC V7.0; OpenCPIC V4.0)	
*1	openUTM-Client (BS2000)	ab V7.0A

*2 BeanConnect (Unix-, Linux- und Windows-Systeme) ab V7.0A
(enthält die Komponente openUTM-JConnect)

Achtung: Verschlüsselte Kommunikation einer UTM-Anwendung V7.0 ist nur zusammen mit UPIC-Client Anwendungen ab V7.0 möglich!

Für die Administration von UTM-Anwendungen über grafische Oberflächen:
openUTM WebAdmin V7.0A

*1 Hinweis:

*1 Mit Fujitsu Software openUTM (Linux- und Windows-Systeme) V7.0A23 wird openUTM WinAdmin
*1 **letztmalig** unterstützt.

Für alle Linux- und Windows-Systeme ist ein C/C++-Compiler erforderlich, für Windows Visual Studio 2010 oder höher.

Für den Einsatz des QuickStart Kit auf Windows wird Visual Studio 2010 oder höher benötigt.

COBOL-Compiler:

*2 Zum Entwickeln von COBOL UTM-Anwendungen:
Rocket Visual COBOL V8.0, V9.0, V10.0

Zum Ablauf:

COBOL Runtime Lizenzen von Rocket

Hinweis: durch die Einführung der UTM-System-Prozesse kann es zu Problemen bei den COBOL-Laufzeitlizenzen kommen, wenn diese zu knapp kalkuliert wurden. Bei der Berechnung der benötigten Anzahl Lizenzen müssen bis zu 3 zusätzliche Lizenzen mitberücksichtigt werden.

*2 Für den Einsatz der Verschlüsselungsfunktionalität auf Linux- und Windows-Systemen muss vom
*2 Kunden OpenSSL zur Verfügung gestellt werden:

*2 OpenSSL ab V3

*1 Auf UNIX-, Linux- und Windows-Systemen kann folgendes Datenbanksystem eingesetzt werden:
ORACLE ab V19c

Hinweis:

Die Softwarekonfiguration enthält z.T. Produkte, die technisch einsetzbar sind, von denen es aber bereits neuere Versionen gibt. Es sollten nach Möglichkeit immer die aktuellen Versionen eingesetzt werden.

3.3 Produkt-Installation

3.3.1 Installation auf Linux-Plattformen

*2 Vor Installation der Software für openUTM importieren Sie bitte den OpenSEAS Schlüssel mit
*2 folgendem Kommando:

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

*2 Den Schlüssel *openseas-buildkey.key* finden Sie auf der CD im Verzeichnis OpenSEAS.

Auf **Linux** wird das Standardverfahren „rpm“ verwendet:

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

Zum Installieren von OUTM-LU62 lesen Sie bitte die dazugehörige readme-Datei.

*1 **ACHTUNG:** Der Einsatz der 32-Bit Variante wird mit dieser Version seitens Fujitsu nicht mehr
*1 unterstützt.

3.3.2 Installation auf Windows-Plattformen

*1 Zum Installieren wechseln Sie in das Produkt-Dateiverzeichnis und rufen unter
*1 Administrationsberechtigung „utm-64.msi“ auf.

Die empfohlene Installationsreihenfolge ist:

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

*1 openUTM darf nicht in ein Verzeichnis, welches Leerzeichen enthält, wie z.B. „Program Files“,
installiert werden. Es muss ein anderes Verzeichnis verwendet werden.
Der Installationspfad darf eine Länge von 39 Zeichen nicht überschreiten.

*1 **ACHTUNG:** Der Einsatz der 32-Bit Variante wird mit dieser Version seitens Fujitsu nicht mehr
*1 unterstützt.

3.3.3 Installation zusammen mit C++ Umgebung auf Linux-Plattformen

Bei der Verwendung von C++-Teilprogrammen muss darauf geachtet werden, dass die C++
Entwicklungsumgebung von openUTM bei der Installation aktiviert werden kann.

Dazu gilt für den C++ Compiler folgendes:

Linux: type g++ muss als Ergebnis den C++ Compiler liefern

Bei der Installation von openUTM wird dann mainutmCC.o erstellt.

mainutmCC.o kann auch (falls erforderlich) nach der Installation von openUTM durch folgende
Eingaben erstellt werden:

*1 <utmpfad>/shsc/CCmainutm

Näheres siehe Handbuch "Einsatz von UTM-Anwendungen".

3.3.4 Installationsprotokoll

Bei der Installation wird auf Linux-Systemen eine Datei <utmpfad>/utm.log erzeugt. Hier wird die
Information abgelegt, die während der Installation auf das Terminal ausgegeben wird.

3.3.5 Hinweise: Kernelparameter für openUTM

Zum Betreiben einer UTM-Anwendung kann es notwendig sein, einige Linux-Kernelparameter zu
vergrößern. Hiervon sind die folgenden Kernelparameter betroffen (ist mehr als ein
Kernelparameter pro Zeile angegeben, so sind alle angegebenen Parameter auf denselben Wert
zu ändern):

- MAXUP max. Anzahl Prozesse pro User
Generell kann es für einzelne Benutzer zu einem Prozessengpass kommen
(Fehlermeldung: fork failed), wenn sie sehr viele Prozesse starten müssen.
In diesem Fall ist MAXUP anzupassen.
- SEMMNS, SEMMNU max. Anzahl von Semaphoren und Semaphor-Strukturen
Die maximal benötigte Anzahl von Semaphoren lässt sich anhand der Beschreibung von MAX
SEMKEY im Handbuch "Anwendungen generieren" berechnen.
*1 Es ist zu berücksichtigen, dass es eventuell weitere Produkte auf dem Rechner gibt, die
ebenfalls Semaphore benötigen.
- SEMMAP, SEMMNI max. Anzahl von Semaphor-Identifizier
Die Semaphore werden zu Semaphor-Feldern zusammengefasst und jedem Semaphor-Feld ist
*1 ein (eindeutiger) Semaphor-Identifizier zugeordnet.
openUTM fasst 20 Semaphore zu einem Semaphor-Feld zusammen.

- SFSZLIM, HFSZLIM max. Größe einer Datei (in Byte)
Werte ab 48 MB (0x3000000) sind in der Regel auch für große UTM-DUMPs ausreichend.
In der Datei /etc/default/login ist der Eintrag ULIMIT=... ebenfalls an den neuen Wert anzupassen.
ULIMIT gibt die maximale Dateigröße (für normale User) in Blocks (1 Block = 512 Bytes) an.
- SDATLIM, HDATLIM max. Swap-Space pro Prozess (in Byte)
- SVMMLIM, HVMMLIM max. virtueller Speicher pro Prozess (in Byte)
Damit größere UTM-Anwendungen mit COBOL-Teilprogrammen noch gebunden werden können, müssen diese Werte auf mindestens 48 MB(0x3000000) erhöht werden, wobei darauf zu achten ist, dass der Swap-Bereich der Anlage >= 48 MB ist.

Normalerweise ist es nicht erforderlich die Kernelparameter zu ändern. In welchen Dateien die Werte geändert werden können und wie Sie einen neuen Kernel erzeugen, entnehmen Sie bitte den Unterlagen Ihres Linux-Distributors.

3.4 Produkt-Einsatz

3.4.1 Hinweise zur Umstellung auf openUTM 7.0A

Bei Umstellung von der Vorgängerversion müssen

- KDCFILE und ROOT-Source erzeugt werden mit KDCDEF von V7.0A
- die ROOT-Source übersetzt werden
- das MSGTAC-Teilprogramm übersetzt werden, falls vorhanden
- die Source für das Meldungsmodul mit KDCMMOD erzeugt und dann übersetzt werden, falls ein eigener Meldungsmodul genutzt wird
- *1 - Bibliothek *libutmcrypt* aus der Bindeanweisung für das Anwendungsprogramm entfernen
- das Anwendungsprogramm neu gebunden werden
- *1 - die Anwenderdaten mit dem KDCUPD in die neue KDCFILE übertragen werden, wenn die Daten relevant sind

KDCUPD unterstützt folgende Versionsübergänge:

openUTM (Linux- und Windows-Systeme) V6.4A auf openUTM (Linux- und Windows-Systeme) V7.0A

openUTM (Linux- und Windows-Systeme) V6.5A auf openUTM (Linux- und Windows-Systeme) V7.0A

openUTM (Linux- und Windows-Systeme) V7.0A auf openUTM (Linux- und Windows-Systeme) V7.0A

Hinweis:

Eine Übertragung der Anwenderdaten mit KDCUPD vom 32-Bit-Modus in den 64-Bit-Modus ist immer nur ohne zusätzlichen Versionsübergang möglich, d.h. nur innerhalb der aktuellen Version.

*1 3.4.2 Hinweise zur Umstellung von openUTM (Linux, Windows-Systeme) V7.0A00 auf V7.0A23

*1 openUTM V7.0A23 ist kompatibel zu openUTM V7.0A00. UTM-Anwendungen und KDCFILE, die
*1 in V7.0A00 im Einsatz waren, können unverändert auch in V7.0A23 eingesetzt werden.
*1 Anwender, die alle Korrekturen nutzen wollen, müssen alle Anwendungsbestandteile neu
*1 erzeugen.

Einige Korrekturen werden erst nach einem erneuten Binden der Anwendung wirksam.

***2 3.4.3 Hinweise zur Umstellung von openUTM V7.0A23 auf openUTM V7.0A35**

*2 openUTM V7.0A35 ist kompatibel zu openUTM V7.0A23. UTM-Anwendungen und KDCFILE, die
*2 in V7.0A23 im Einsatz waren, können unverändert auch in V7.0A35 eingesetzt werden.
*2 Anwender, die alle Korrekturen nutzen wollen, müssen alle Anwendungsbestandteile neu
*2 erzeugen.

***2 3.4.4 Einsatz des Tools kdcdef unter RHEL 10**

*2 Für den Einsatz des Tools kdcdef unter RHEL 10 ist ein zusätzlicher symbolischer Link
*2 notwendig. Bitte erstellen Sie unter /usr/lib64 den folgenden symbolischen Link:
*2 In -s /usr/lib64/libcrypt.so.2 libcrypt.so.1

3.4.5 Umstellung einer stand-alone Anwendung in eine UTM-Cluster-Anwendung

Bei einer UTM-Cluster-Anwendung sind folgende Umstellungen möglich:

- Umstellung einer stand-alone-Anwendung V7.0A auf eine UTM-Cluster-Anwendung V7.0A
 - Umstellung einer UTM-Cluster-Anwendung V7.0A auf eine stand-alone UTM-Anwendung V7.0A
- Eine detaillierte Beschreibung der Vorgehensweise finden Sie im Manual "Einsatz unter Unix-, Linux und Windows-Systemen".

3.4.6 Client-Programme mit lokaler Anbindung an openUTM

Client-Programme mit lokaler Anbindung an openUTM müssen neu übersetzt und gebunden werden.

3.4.7 Binden von UTM OSI-TP Anwendungen auf Linux-Plattformen

Bei der Angabe der OSS Bibliotheken ist unbedingt die folgende Reihenfolge einzuhalten:
Zuerst ist die Bibliothek libossutm.so und danach erst die Bibliothek liboss.so anzugeben.

Hält man die Reihenfolge nicht ein, dann führt das zu folgendem Start-Fehler:

```
P001 Fehler beim OSS Aufruf (o_create() ·): - 1, 300, 199, 0
```

```
K060 Der Anwendungslauf wurde abgebrochen; die Ursache ist XINI06.
```

3.4.8 Binden und Starten von UTM-Anwendungen auf Linux-Systemen

Beim Binden und beim Starten einer UTM-Anwendung auf Linux-Systemen muss die Umgebungsvariable LD_LIBRARY_PATH bzw. LD_LIBRARY_PATH_64 auf <utmpfad>/sys gesetzt sein.

3.4.9 Hinweise zur Beispielanwendung für UTM-Cluster-Anwendungen

Bei der Installation einer Beispielanwendung für eine UTM-Knoten-Anwendung muss folgendes berücksichtigt werden:

- *1
- Für eine Beispielanwendung kann eine installierte UTM-Knoten-Anwendung nicht in eine stand-alone Anwendung konvertiert werden.
 - Ein Filebase Verzeichnis für eine Knoten-Anwendung und ein Filebase Verzeichnis für eine stand-alone Anwendung dürfen im Dateibaum nicht unter demselben übergeordneten Dateiverzeichnis angelegt werden.

3.5 Entfallene und gekündigte Funktionen

3.5.1 Entfallene Funktionen

Dienstprogramm KDCDEF

Mehrere Funktionen wurden gestrichen und können nicht mehr in KDCDEF generiert werden. Wenn sie dennoch angegeben werden, wird dies im KDCDEF-Lauf mit einem Syntaxfehler abgelehnt.

Im Einzelnen wurde Folgendes gestrichen:

- TNS Betrieb
Beim Start einer UTM-Anwendung wird die TNS-Generierung nicht mehr gelesen. Die Adressierungsinformation muss vollständig bei der Konfiguration mit KDCDEF hinterlegt werden.
- KDCDEF-Anweisung PTERM
Operanden-Werte 1 und 2 für *ENCRYPTION-LEVEL*
- KDCDEF-Anweisung TPOOL
Operanden-Werte 1 und 2 für *ENCRYPTION-LEVEL*
- KDCDEF-Anweisung TAC
Operanden-Wert 1 für *ENCRYPTION-LEVEL*

*1 NetCOBOL

*1 Mit openUTM V7.0 wird der COBOL Compiler NetCOBOL aus technischen Gründen nicht mehr
*1 unterstützt.

3.5.2 Gekündigte Funktionen

- *1 • Fujitsu Software openUTM WinAdmin
*1 Mit dieser Version wird openUTM WinAdmin letztmalig unterstützt.

3.6 Inkompatible Änderungen

- Kompilieren von ROOT C SOURCE
Die für die Übersetzung des ROOT C Sources benötigten C-Headerfiles haben sich in openUTM (Linux- und Windows-Systeme) V7.0 geändert. Da der ROOT Source vom Dienstprogramm KDCDEF vollständig erzeugt wird, erfolgte diese Anpassung automatisch. Allerdings kann es sein, dass in der Übersetzungsprozedur (z. B. Makefile) als Abhängigkeit C-Headerfiles angegeben wurden, die ab UTM V7.0 nicht mehr zur Verfügung stehen. Diese Abhängigkeiten müssen beim Umstieg auf UTM V7.0 gelöscht werden.
- Rückgabelänge KCRLM bei MGET mit KCRCCC 02Z bei USP-Clients
Liest ein Teilprogramm ein Nachrichtenteil mit MGET und die Länge im Parameter KCLA ist kleiner als die tatsächliche Länge des (restlichen) Nachrichtenteils, so wird der KDCS-Returncode 02Z gesetzt und im Rückgabefeld KCRLM nun die tatsächliche Länge des (restlichen) Nachrichtenteils zurückgegeben und nicht mehr der Wert von KCLA.

Bsp. bei einem Nachrichtenteil mit Länge 20:

MGET KCLA=5 -> 02Z, KCRLM=20 (statt bisher KCRLM=5)

MGET KCLA=7 -> 02Z, KCRLM=15 (statt bisher KCRLM=7)

MGET KCLA=8 -> 000, KCRLM=8

- *1 • Anwendungsprogramm utmwork
*1 Durch den Wegfall der UTM-Crypt Variante wird ab openUTM (Linux- und Windows-Systeme)
*1 V7.0A die UTM Bibliothek libutmrcrypt nicht mehr benötigt. Bitte entfernen Sie diese Bibliothek
*1 aus Ihrer Bindeanweisung.

3.7 Einschränkungen

Hinweise zum Einsatz der Beispielanwendung:

Für die Beispielanwendung wird keine Gewährleistung und Service-Verpflichtung übernommen.

3.8 Verhalten im Fehlerfall

Zur Fehlerdiagnose sind folgende Angaben nötig:

- genaue Beschreibung der Fehlersituation
- Angabe der Versions-/Korrekturstände der beteiligten Software
- Genaue Angabe des Rechnertyps

Die Fehlerunterlagen sollten möglichst vollständig vorhanden sein. Als Fehlerunterlagen werden benötigt:

- UTM-Dumps aller Work-Prozesse sowie zugehörige gcores. Diese Dateien sollten als Binärdateien, d.h. in nicht aufbereiteter Form, vorliegen
- SYSLOG-Datei(en)
- stdout/stderr-Protokolle der UTM-Prozesse
- KDCDEF-Generierung, Startprozedur und Startparameter
- core-Dateien mit zugehörigen Phasen (utmwork) und shared objects. Die shared objects können ermittelt werden mit "ldd utmwork"
- Binder- und Übersetzungslisten

Bei Fehlern in einer UTM-Cluster-Anwendung zusätzlich:

- alle Cluster globale Dateien, Protokolle (und DUMPS) aller Knoten-Anwendungen
- bei Problemen, die durch das Zusammenspiel der Knoten-Anwendungen verursacht wurden, die Protokolldateien von allen anderen Knoten-Anwendungen
- die Startprozedur und die bei der Generierung als EMERGENCY-CMD und FAILURE-CMD angegebenen Prozeduren

Vorgehen bei Fehlern:

- PENDER-Dump auswerten
- Fehler reproduzieren unter Verwendung geeigneter Debugger wie z.B. dbx, gdb
- Aufrufhierarchie beim core mit Hilfe eines Debuggers ermitteln

Es soll versucht werden, die Fehler unter Verwendung statischer Bibliotheken zu reproduzieren.

Bei abnormalem Anwendungsabbruch müssen die UTM-Dumps und gcores aller Prozesse gesichert werden. Falls möglich sollte der Abbruch mit TESTMODE=ON (Startparameter) reproduziert werden, wobei TRACEREC und IPTRACE mindestens auf 10000 eingestellt sein sollten.

*2
*2

Abbruch mit Signalen:

Trat ein PENDER-Dump mit 70Z/XT10 oder XT11 oder Anwendungsabbruch mit SIG010/SIG011 (Signal SIGBUS/SIGSEGV) auf, sollte die UTM-Signalbehandlung mit dem Startparameter START STXIT=OFF ausgeschaltet werden, damit beim fehlerhaften Befehl sofort ein core-Dump erzeugt wird. Vor dem nächsten Neustart der Anwendung muss auf jeden Fall kdcrem aufgerufen werden, um evtl. noch vorhandene Semaphore und Shared Memories sowie Statusinformationen der Anwendung in der <utmpfad>/applifile zu löschen bzw. zurückzusetzen.

Bei Fehlern, die in Zusammenhang mit der openUTM-Netzanbindung stehen, können zusätzlich folgende Unterlagen erstellt werden:

- Meldungen der UTM-Netzprozesse in stdout/stderr

*1

- OSS-Traces
- UTM-BCAM-Trace
- Bei heterogener Kopplung Generierungsinformation und Trace von openUTM LU6.2 und seiner SNA Komponenten und vom IBM-SNA System

Erstellung der UTM-BCAM-Traces und OSS-Traces siehe Handbuch "Meldungen, Test und Diagnose".

Bei Fehlern, die in Zusammenhang mit Datenbanken stehen, können zusätzlich folgende Unterlagen erstellt werden:

- XA Debug Trace
- weitere Unterlagen siehe entsprechende Freigabemitteilung des Datenbanksystems

Nach Startfehlern (K049-Meldung) z.B. mit Insert 40 / 32 muss vor einem neuen Start das UTM-Tool kdcrem aufgerufen werden.

*1 3.9 Ergänzungen zum Benutzerhandbuch

*1 3.9.1 openUTM Meldungen, Test und Diagnose

*1 Mit der Freigabe von Fujitsu Software openUTM (Linux- und Windows-Systeme) V7.0A23 wurde
*1 auch das Handbuch „openUTM Meldungen, Test und Diagnose auf Unix-, Linux- und Windows-
*1 Systemen“ aktualisiert.

*1 Die neue Ausgabe ist im Internet unter

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

*1 verfügbar.

*1

*1 3.9.2 openUTM Anwendungen administrieren

*1

*1 Kapitel 11.3.1.25 kc_pterm_str – Clients und Drucker

*1

*1 t_prot

*1 ...

*1 ‚H‘ natives TCP/IP-Transportprotokoll für die Kommunikation mit dem Hypertext Transfer
*1 Protocol (http) über die Socket-Schnittstelle (SOCKET)

*1 ...

*2 3.9.3 Zusätzliche Inserts in U315

*2 Es gibt einige zusätzliche Inserts in der Meldung U315 ab openUTM V7.0A35.

8	Verbindungsabbau durch openUTM Netzprozess aufgrund einer Fehlersituation. Ursache siehe vorangegangene U Meldung.
9	Ablehnung eines Verbindungsaufbauwunsches durch den openUTM Netzprozess. Ursache siehe vorangegangene U Meldung.
17	Ablehnung eines Verbindungsaufbauwunsches durch den Partner

4 Hardware-Unterstützung

Es wird die Hardware unterstützt, auf der die genannten Betriebssystemversionen ablauffähig sind. Eingeschlossen sind alle auf der Intel-x86-Technologie basierenden Systeme z.B. Laptops, PCs, PRIMERGY-Systeme.

Weitere Plattformen auf Anfrage.

CPU mindestens 250 MHz, Hauptspeicher mind. 512 MB

UTM-Cluster-Anwendungen:

- Für den Betrieb von UTM-Cluster-Anwendungen ist ein Network File System/Service (NFS) notwendig.
- Knoten einer UTM-Cluster-Anwendung können Systeme mit unterschiedlichen Linux-Versionen sein. Knoten mit anderen Betriebssystemen (andere Linux-Distributionen) sind nicht möglich.

*1