

FUJITSU Software openUTM (BS2000)

Version 7.0A  
November 2019

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.

© 2019 Fujitsu Technology Solutions GmbH

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern. BS2000 ist eine Marke von Fujitsu Technology Solutions GmbH in Europa und in anderen Ländern.

<b>1</b>	<b>ALLGEMEINES</b>	<b>1</b>
1.1	<b>Bestellung</b>	<b>2</b>
1.1.1	Nutzungsrechte	2
1.1.2	Nutzungsrecht für CALLUTM	3
1.2	<b>Auslieferung</b>	<b>3</b>
1.2.1	Software auf Datenträger	3
1.2.2	Software zum Download openUTM WebAdmin	4
1.2.3	Software WS4UTM und BizXML2COBOL	5
1.3	<b>Dokumentation</b>	<b>5</b>
1.3.1	Verfügbarkeit	5
1.3.2	Dokumentation der Open Group-Schnittstellen	5
<b>2</b>	<b>SOFTWARE-ERWEITERUNGEN</b>	<b>6</b>
2.1	<b>Neue Funktionen in openUTM (BS2000) V7.0A</b>	<b>6</b>
2.1.1	UTM als HTTP-Server	6
2.1.2	Verschlüsselung	8
2.1.3	OSI-TP Kommunikation und Portnummern	8
2.1.4	Subnetze	8
2.1.5	Zugangsdaten für den XA-Datenbank-Anschluss	9
2.1.6	Reconnect für den XA-Datenbank-Anschluss	9
2.1.7	Sonstige Änderungen	9
2.2	<b>Realisierte Change-Requests (CR)</b>	<b>9</b>
<b>3</b>	<b>TECHNISCHE HINWEISE</b>	<b>9</b>
3.1	<b>Ressourcenbedarf</b>	<b>9</b>
3.2	<b>SW-Konfigurationen</b>	<b>10</b>
3.3	<b>Produkt-Installation</b>	<b>11</b>
3.3.1	Das Subsystem UTM	12
3.3.2	Koexistenz mehrerer UTM-Versionen (Parallelbetrieb)	12
3.3.3	Subsystem UTM-SM2	12
3.3.4	Subsystem KDCMON	13
3.4	<b>Produkt-Einsatz</b>	<b>14</b>
3.4.1	Hinweise zur Umstellung auf openUTM (BS2000) V7.0A	14
3.4.2	Hinweise zur Umstellung von /390 auf x86	14
3.4.3	Optionale Reps	15
3.5	<b>Entfallene und gekündigte Funktionen</b>	<b>15</b>
3.5.1	Entfallene Funktionen	15
3.5.2	Gekündigte Schnittstellen	15
3.6	<b>Inkompatible Änderungen</b>	<b>15</b>
3.7	<b>Einschränkungen</b>	<b>15</b>
3.8	<b>Verhalten im Fehlerfall</b>	<b>16</b>



# 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:

- Die grafische Oberfläche openUTM WinAdmin zur Administration von UTM-Anwendungen wird auf der WinAdmin-CD mitgeliefert. openUTM WinAdmin ist ablauffähig auf Windows- und Windows Server-Systemen, sowie auf Linux-Systemen.
- Das webbasierte Tool openUTM WebAdmin steht für die Administration von UTM-Anwendungen über einen Internet-Browser zur Verfügung.
- Mit WS4UTM wird ein Tool zur Verfügung gestellt, welches auf komfortable Weise ermöglicht, Teilprogramme einer UTM-Anwendung als Web Service verfügbar zu machen. Aus einer in Java implementierten Web-Anwendung heraus kann ein UTM-Service als Webservice aufgerufen werden. Damit kann UTM in SOA-Lösungen integriert werden (SOA = Service Oriented Architecture).
- 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.
- Mit BizXML2Cobol ist es möglich, aus vorhandenen Service-Definitionen (WSDL-Beschreibung oder XML Schema) Cobol Datenstrukturen und Code zu erzeugen, der sich in vorhandene transaktionale Cobol-Applikationen integrieren lässt, so dass diese den vordefinierten Service realisieren. Damit wird der Top-Down-Ansatz (von der geschäftsrelevanten Definition zur Implementierung) in SOA Projekten auch für den Einbezug existierender Programmlogik unterstützt.

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.
- Für die Kopplung von UTM-Anwendungen mit Java 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:

<http://de.ts.fujitsu.com/openUTM>

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 (BS2000) V7.0A sowie alle nötigen Informationen für den Umstieg von der Vorgängerversion openUTM (BS2000) V6.5A. Nachträglich bekannt gewordene Änderungen werden in dieser Datei aktualisiert und nachgeliefert.

Der Inhalt entspricht dem Freigabestand: November 2019.

Diese Freigabemitteilung ist, wie alle Freigabemitteilungen für BS2000-Produkte, 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.

## **1.1 Bestellung**

openUTM (BS2000) V7.0A kann über Ihre zuständige Vertriebsgesellschaft bezogen werden. Für openUTM (BS2000) V7.0A gelten die allgemeinen Bedingungen zum Kauf-, Wartungs-, Softwareüberlassungs- und Softwarebetreuungsvertrag.

### **1.1.1 Nutzungsrechte**

Die Nutzungsrechte sind Bestandteil des Vertrages, den Sie beim Bezug von openUTM (BS2000) mit Fujitsu Technology Solutions abgeschlossen haben.

openUTM (BS2000) V7.0A ist ein Lizenzprodukt. Für den Einsatz sind Nutzungsrechte notwendig. Sie werden in Varianten je nach Leistungsklasse des Systems inkl. Nutzungsrecht pro User-Klasse angeboten.

Für Verteilte Transaktionsverarbeitung gibt es das zusätzliche Nutzungsrecht UTM-D pro System (in Varianten je nach Leistungsklasse des Systems inkl. Nutzungsrecht pro User-Klasse).

openUTM (BS2000) unterliegt Exportbeschränkungen.

### 1.1.2 Nutzungsrecht für CALLUTM

Mit openUTM auf BS2000-Systemen wird das Programm CALLUTM ausgeliefert, das es erlaubt, aus einer beliebigen BS2000-Batch- oder Dialog-Task heraus UTM-Services aufzurufen. Das Programm bietet eine SDF-Oberfläche und kann aus dem BS2000-Kommandomodus aufgerufen werden.

Zum Betrieb des mit openUTM (BS2000) V7.0A ausgelieferten CALLUTM V7.0A wird eine Runtime-Lizenz von openUTM-Client (BS2000) V7.0A benötigt, da die UPIC Bibliotheken beim Aufruf von CALLUTM dynamisch nachgeladen werden. Das dynamische Nachladen hat den Vorteil, dass man wählen kann, ob die Kommunikation über Socket oder CMX erfolgen soll-

## 1.2 Auslieferung

### 1.2.1 Software auf Datenträger

Die Lieferung der Dateien zu openUTM (BS2000) V7.0A erfolgt mit dem Lieferverfahren SOLIS. Im SOLIS2-Lieferanschreiben sind die jeweils gültigen Dateien und Datenträger-Merkmale von openUTM aufgeführt.

Die Lieferung umfasst alle Produktbestandteile für openUTM (BS2000) V7.0A (incl. openUTM-D).

### Lieferbestandteile von openUTM (BS2000) V7.0A

Im Folgenden werden die Lieferbestandteile von openUTM (BS2000) V7.0A aufgelistet:

<u>Datei</u>	<u>Inhalt</u>
<u>Modulbibliotheken</u>	
SYSLNK.UTM.070.TPR	UTM-Systemcode (TPR) für /390
SKMLNK.UTM.070.TPR	UTM-Systemcode (TPR) für x86
SYSLNK.UTM.070	openUTM-Modulbibliothek nur TU-Code
SYSLNK.UTM.070.SPLRTS	SPL-Laufzeitsystem für openUTM
SYSLNK.UTM.070.UTIL	Ladebibliothek für UTM-Dienstprogramme
SYSLIB.UTM.070.XOPEN	Bestandteile für X/OPEN-Schnittstellen XATMI, CPIC, TX
SYSLIB.UTM.070.SSL	UTM SSL-PROXY für HTTPS Zugang

### Includes/COPY-Elemente

SYSLIB.UTM.070.ASS	Assembler
SYSLIB.UTM.070.COB	COBOL
SYSLIB.UTM.070.C	C
SYSLIB.UTM.070.PLI1	PL/1
SYSLIB.UTM.070.PASC	PASCAL-XT
SYSLIB.UTM.070.FOR	FORTRAN
SYSLIB.UTM.070.SPL	SPL4

### openUTM-Meldungswesen

SYSMSH.UTM.070.MSGFILE	Meldungsdefinitionsdatei
SYSDAT.UTM.070.MTXTIN	Eingabe für UTM-Tool KDCMTXT

### BS2000-Meldungswesen

SYSMES.UTM.070      BS2000 Meldungsdatei

### Syntaxdatei

SYSSDF.UTM.070

### Installation

SYSSSC.UTM.070.190      SSD-Objekt für BS2000 OSD/BC V10.0 (/390 und x86)

SYSSSC.UTM.070.200      SSD-Objekt für BS2000 OSD/BC V11.0 (/390 und x86)

SYSNRF.UTM.070      NOREF-Datei

SYSRMS.UTM.070      RMS-Datei

SYSSII.UTM.070      Strukturinformation für IMON

### openUTM-Tools

SYSLNK.UTM.070.CALLUTM      UPIC-Anwendung für Administration und Client-Server-Kommunikation über eine SDF-Kommandooberfläche

SYSSDF.UTM.070.CALLUTM      SDF-Syntaxdatei für das Client-Programm CALLUTM

### Beispiele

SYSPRC.UTM.070      Bibliothek mit precompilierten Beispielprozeduren, die auf Anlagen ohne SDF-P ablauffähig sind

SYSLIB.UTM.070.EXAMPLE      Bibliothek mit Beispielprogrammen und Beispielprozeduren

### Information zu Open Source Software

SYSDOC.UTM.070.OSS      Information zu den Lizenzen der ausgelieferten Open Software Komponenten

### Dokumentation

SYSGM.UTM.070.D      Freigabemitteilung deutsch

SYSGM.UTM.070.E      Freigabemitteilung englisch

### **openUTM WinAdmin**

Mit openUTM (BS2000) V7.0A wird auch das Produkt FUJITSU Software openUTM WinAdmin V7.0A auf einer eigenen CD ausgeliefert, mit dem Sie UTM-Anwendungen über eine komfortable grafische Oberfläche administrieren können.

#### **1.2.2 Software zum Download openUTM WebAdmin**

Zusammen mit openUTM (BS2000) 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.

openUTM WebAdmin wird auf der openUTM Website

<http://de.ts.fujitsu.com/openUTM> zum Download zur Verfügung gestellt.

openUTM WebAdmin V7.0A wird als ein add-on zum SE Manager auf einer CD/DVD ausgeliefert und wird auch als ISO image unter dem oben spezifizierten link bereitgestellt.

### **XML für openUTM**

Die Software "XML für openUTM" ist ein kostenloses Zusatzprodukt zu openUTM, für das Fujitsu Technology Solutions keine Fehlerbehebung übernimmt.



Die Software ist beziehbar über  
<http://de.ts.fujitsu.com/openutm>

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 unter

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

### 1.2.3 Software WS4UTM und BizXML2COBOL

Consulting Projektpakete, in deren Rahmen die Software „WebServices for openUTM“ (WS4UTM) bzw. BizXML2Cobol zur Verfügung gestellt wird, können erworben werden. Für detaillierte Informationen wenden Sie sich bitte an den für Sie zuständigen Sitz der Region von Fujitsu.

## 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 (BS2000) 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.
  - KDCDEF-Anweisung CHAR-SET:
    - Mit dieser Anweisung können jeder der von openUTM gestellten vier Code-Konvertierungen jeweils bis zu vier Character-Set Namen zugeordnet 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 (*KCHTT-PINF/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 der HTTP-Response.
- Function *kcHttpPutStatus()*  
Diese Funktion übergibt einen HTTP-Statuscode für die HTTP-Response.
- Kommunikation über den Secure Socket Layer (SSL)  
Ist für eine UTM-Anwendung ein BCAMAPPL mit T-PROT=(SOCKET,....,SECURE) generiert, dann wird beim Anwendungsstart von UTM eine zusätzliche Task mit einem Reverse Proxy gestartet, der für die Anwendung als TLS Termination Proxy fungiert und über den sämtliche SSL-Kommunikation abgewickelt wird.

### 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 (BS2000) V7.0 ist dies nicht mehr erforderlich. Mit openUTM (BS2000) 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!

### 2.1.3 OSI-TP Kommunikation und Portnummern

- KDCDEF-Anweisung OSI-CON  
Der Operand LISTENER-PORT kann auch auf BS2000-Systemen angegeben werden.
- Administrationsschnittstelle KDCADMI  
In der Struktur *kc\_osi\_con\_str* wird auch auf BS2000-Systemen im Feld *listener-port* die Portnummer angezeigt.

### 2.1.4 Subnetze

In einer UTM-Anwendung können auch auf BS2000-Systemen Subnetze generiert werden, um den Zugang zu UTM-Anwendungen auf definierte IP-Adressbereiche beschränken zu können. Zusätzlich kann die Namensauflösung per DNS gesteuert werden.

Dazu wurden folgende Schnittstellen geändert:

- Generierung
  - KDCDEF-Anweisung SUBNET:  
Die SUBNET-Anweisung kann auch auf BS2000-Systemen angegeben werden.
  - 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*.

### 2.1.5 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*.

### 2.1.6 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.

### 2.1.7 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).

## 2.2 Realisierte Change-Requests (CR)

keine

# 3 Technische Hinweise

## 3.1 Ressourcenbedarf

### Virtueller Adressraum

Für openUTM (BS2000) V7.0A ergeben sich folgende Speicherbedarfswerte:

Systemcode: 4 MB

Jede UTM-Task belegt zusätzlich Speicher in Klasse 5 und 6, dessen Größe von der Konfiguration der Anwendung und von den Anwendungsprogrammen abhängt.

### Statischer Plattenspeicher

Die Produktdateien von openUTM (BS2000) V7.0A belegen ca. 35.000 – 40.000 PAM-Seiten auf /390-Rechnern und x86-Rechnern.

### Dynamischer Plattenspeicher

Der benötigte Platz für KDCFILE-, USLOG-, SYSLOG-Dateien und Anwendungsprogramm ist abhängig von der Größe der Konfiguration.

## 3.2 SW-Konfigurationen

OSD/BC	ab V10.0	für /390
OSD/XC	ab V10.0	für x86
DSSM	ab V4.3B	
openNet Server	ab V3.6A	enthält BCAM, CMX, DCAM, ONETSERV, SOCKETS, VTSU-B, XHCS für Unicode-Unterstützung u.a.
OSS	ab V4.1D	für OSI-TP-Anwendungen
FHS	ab V8.3B	für die Nutzung von Formaten mit CCSNAME für asynchrone Formatierung und für Unicode-Unterstützung
IFG	ab V8.3A	
JV	ab V15.0A	
RSO	ab V3.6A	
OMNIS	ab V8.4A	
OMNIS-MENU	ab V3.4A	
DRIVE	ab V3.1A	
UDS/SQL	ab V2.9A	
SESAM/SQL	ab V9.0A	
LEASY	ab V6.2A	
ORACLE	ab 11.2	nur XA-Anschluss
CRTE	ab V10.0A	
SPL4	ab V2.9A	
C/C++	ab V3.2A	
COBOL85	ab V2.3A	
COBOL2000	ab V1.5A	
ASSEMBH	ab V1.3	
FOR1	ab V2.2C	
FORTTRAN90	ab V1.0A	
PASCAL-XT	ab V2.2B	
PLI1	V4.2A	
FormplagDoors	ab V3.0B	
SECOS	ab V5.2B	notwendig für Kerberos

Kommt es bei Einsatz älterer Compilerversionen und Laufzeitsysteme, für die seitens Fujitsu Technology Solutions keine Wartungsverpflichtung mehr besteht, zu Problemen, besteht kein Anspruch auf Gewährleistung oder Korrekturen.

In diesem Fall wird empfohlen, auf Compiler und/oder Laufzeitsysteme einer neueren Version umzustellen.

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

openUTM (BS2000)	ab V6.4
openUTM (Unix-, Linux- und Windows-Systeme )	ab V6.4

Wenn es bei der Kopplung mit Versionen kleiner V6.4 Probleme gibt, dann besteht keine Wartungsverpflichtung seitens Fujitsu Technology Solutions; in diesem Fall sollte ein Kunde die Partneranwendungen auf UTM-Versionen größer/gleich V6.4 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 6.1.2

Solaris: SNAP-IX ab Version 7.0.2.4; von Metaswitch Networks

Für verteilte Transaktionsverarbeitung mit Java EE Anwendungen werden folgende Versionen unterstützt:

BeanConnect ab V3.0B

Für Client-Server-Kommunikation wahlweise:

openUTM-Client (Unix-, Linux- und Windows-Systeme) ab V6.4A  
(enthält Trägersysteme UPIC V6.4; OpenCPIC V4.0)

openUTM-Client (Linux- und Windows-Systeme) ab V7.0A  
bei Verwendung von Verschlüsselung  
(enthält Trägersysteme UPIC V7.0; OpenCPIC V4.0)

openUTM-Client (BS2000) ab V6.4A

BeanConnect (Unix-, Linux- und Windows-Systeme) ab V3.0B  
(enthält die Komponente openUTM JConnect)

CALLUTM (BS2000) ab V6.4

**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 WinAdmin V7.0A

openUTM WebAdmin V7.0A

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

Die Installation des Produktes openUTM (BS2000) muss mit dem Installationsmonitor IMON durchgeführt werden. Die Informationen zur Installation im Lieferanschreiben sowie im Handbuch des Produktes müssen berücksichtigt werden.

Die erforderlichen Eingaben sowie der Ablauf der Installation mit IMON sind im Handbuch zu IMON beschrieben

Für die folgenden Produktdateien ist in den SSD-Objekten \*DEFAULT-USERID als Installationskennung voreingestellt:

SYSLNK.UTM.070.TPR Ladebibliothek für /390-Hardware

SKMLNK.UTM.070.TPR Ladebibliothek für x86

### 3.3.1 Das Subsystem UTM

Der gesamte UTM-Systemcode, inkl. Abbildungsmodul zum BS2000-System und UTM-D-Systemcode, wird vom DSSM als ein Subsystem geladen und verwaltet. In den DSSM-Anweisungen für das Subsystem UTM wird der Zeitpunkt festgelegt, zu dem der UTM-Systemcode geladen werden soll.

Darin ist `CREATIM=AT-CREATION-REQUEST` voreingestellt, das bedeutet, dass das Subsystem UTM per Kommando gestartet werden muss.

Bei der Standardinstallation erzeugt SOLIS automatisch einen Eintrag im Subsystemkatalog.

Für die Aufnahme in den Subsystemkatalog werden folgende SSD-Objekte ausgeliefert:

`SYSSSC.UTM.070.190` für BS2000 OSD/BC V10.0 für /390 und x86

`SYSSSC.UTM.070.200` für BS2000 OSD/BC V11.0 für /390 und x86

### 3.3.2 Koexistenz mehrerer UTM-Versionen (Parallelbetrieb)

Es ist möglich, im gleichen BS2000-System mehrere UTM-Versionen zu laden und im Produktivbetrieb gleichzeitig zu nutzen.

Der Parallelbetrieb ist mit allen UTM-Versionen möglich, die für das entsprechende BS2000-System freigegeben sind.

Generell ist dabei Folgendes zu beachten:

- Sollen ältere Versionen erhalten bleiben, so muss dies bei der Installation in IMON explizit angegeben werden. Standardfall ist, dass diese deinstalliert werden.
- Die Subsysteme müssen mit dem Parameter `VERSION-PARALLELISM=*COEXISTENCE-MODE` beim Kommando `/START-SUBSYSTEM` gestartet werden.
- Zwei UTM-Anwendungen mit gleichem Anwendungsnamen können nicht in verschiedenen UTM-Versionen zugleich laufen. `openUTM` verhindert dann den Start der zweiten Anwendung.

### 3.3.3 Subsystem UTM-SM2

Mit dem Messmonitor SM2 des BS2000-Systems kann man Werte zur aktuellen Leistung in UTM-Anwendungen auf dem Bildschirm anzeigen und für spätere Auswertungen in der SM2-Messwertdatei sammeln.

Zur Kommunikation zwischen `openSM2` und `openUTM` (BS2000) dient der Baustein UTM-SM2. Er ist als eigenständiges BS2000-Subsystem realisiert.

Die Dateien für das Subsystem UTM-SM2 werden mit BS2000-GA ausgeliefert. Bei der Installation von BS2000 wird automatisch auch der Eintrag für UTM-SM2 im Subsystemkatalog erzeugt.

Folgende Versionen von UTM-SM2 sind einzusetzen:

`UTM-SM2 V19.0` für BS2000 OSD/BC V10.0 für /390 und x86

`UTM-SM2 V20.0` für BS2000 OSD/BC V11.0 für /390 und x86

Bei der Standardinstallation erzeugt SOLIS automatisch den Eintrag im Subsystemkatalog für UTM-SM2. Zum Einsatz von UTM-SM2 sind folgende Dateien notwendig:

Ladebibliotheken:

`SYSLNK.UTM-SM2.190` für BS2000 OSD/BC V10.0 für /390

`SKMLNK.UTM-SM2.190` für BS2000 OSD/BC V10.0 für x86

`SYSLNK.UTM-SM2.200` für BS2000 OSD/BC V11.0 für /390

`SKMLNK.UTM-SM2.200` für BS2000 OSD/BC V11.0 für x86

REP-Datei:

`SYSREP.UTM-SM2.nnn`



#### SSD-Objekte:

SYSSSC.UTM-SM2.190 für BS2000 OSD/BC V10.0 für /390 und x86  
SYSSSC.UTM-SM2.200 für BS2000 OSD/BC V11.0 für /390 und x86

Im SSD-Objekt sind folgende Standardwerte voreingestellt:

- Ladezeitpunkt : Laden mit DSSM-Kommando erforderlich
- Ladebibliothek: \$TSOS.SYSLNK.UTM-SM2.nnn bzw.  
\$TSOS.SKMLNK.UTM-SM2.nnn
- REP-Datei: \$TSOS.SYSREP.UTM-SM2.nnn

Suffix nnn = 190 für UTM-SM2 V19.0A

Suffix nnn = 200 für UTM-SM2 V20.0A

Der Systemverwalter darf den voreingestellten Ladezeitpunkt nicht ändern!

### **3.3.4 Subsystem KDCMON**

KDCMON ist ein Werkzeug zum Vermessen von UTM-Anwendungen. Es ist als eigenes Subsystem realisiert.

Mit KDCMON kann man UTM-Anwendungen vermessen, die in einem BS2000-System unter verschiedenen UTM-Versionen ablaufen. Die Dateien für das Subsystem KDCMON werden mit BS2000-GA ausgeliefert. Bei der Installation von BS2000 wird automatisch auch der Eintrag für KDCMON im Subsystemkatalog erzeugt.

Folgende KDCMON-Versionen sind einzusetzen:

KDCMON V19.0A in BS2000 OSD/BC V10.0 für /390 und x86

KDCMON V20.0A in BS2000 OSD/BC V11.0 für /390 und x86

Bei der Standardinstallation erzeugt SOLIS automatisch den Eintrag im Subsystemkatalog für KDCMON. Zum Einsatz von KDCMON sind folgende Dateien notwendig:

#### Ladebibliotheken:

SYSLNK.KDCMON.190 für BS2000 OSD/BC V10.0 für /390  
SKMLNK.KDCMON.190 für BS2000 OSD/BC V10.0 für x86  
SYSLNK.KDCMON.200 für BS2000 OSD/BC V11.0 für /390  
SKMLNK.KDCMON.200 für BS2000 OSD/BC V11.0 für x86

#### REP-Datei:

SYSREP.KDCMON.nnn

#### SSD-Objekte:

SYSSSC.KDCMON.190 für BS2000 OSD/BC V10.0 für /390 und x86  
SYSSSC.KDCMON.200 für BS2000 OSD/BC V11.0 für /390 und x86

#### Ausführbare Programme:

SYSPRG.KDCMON.nnn Programm zum Starten von KDCMON  
SYSPRG.KDCMON.nnn.KDCPMSM Programm zum Umwandeln d. Messdaten

#### Meldungsdatei:

SYSMES.KDCMON.nnn BS2000 Meldungsdatei

Im SSD-Objekt sind folgende Standardwerte voreingestellt:

- Ladezeitpunkt : implizit beim ersten Aufruf
- Ladebibliothek : \$TSOS.SYSLNK.KDCMON.nnn bzw.  
\$TSOS.SKMLNK.KDCMON.nnn
- REP-Datei: \$TSOS.SYSREP.KDCMON.nnn

Suffix nnn = 190 für KDCMON V19.0A

Suffix nnn = 200 für KDCMON V20.0A

Der Systemverwalter darf die Voreinstellung für den Ladezeitpunkt nicht ändern!

KDCMON wird geladen, indem man das Programm SYSPRG.KDCMON.nnn unter \$TSOS startet.

### 3.4 Produkt-Einsatz

#### 3.4.1 Hinweise zur Umstellung auf openUTM (BS2000) V7.0A

Bestandteile von verschiedenen UTM-Versionen dürfen nicht gemischt werden. Der Versionscheck von openUTM prüft, ob die Versionen der Produktbestandteile übereinstimmen.

Bei der Umstellung von einer Vorgängerversion auf openUTM (BS2000) V7.0A sind folgende Schritte durchzuführen:

- Generierung von KDCFILE und KDCROOT mit dem KDCDEF von openUTM (BS2000) V7.0A (evtl. KDCDEF-Änderungen berücksichtigen)
- Übersetzung des ROOT-Source
- Übersetzen eigener Administrationsprogramme
- Übersetzen des MSGTAC-Teilprogramms, falls vorhanden und ggfs. eigener Asynchron Programme, falls deren TAC über die KDCDEF-Anweisung MSG-DEST als Meldungsziel definiert wurde
- werden eigene Meldungsmodul verwendet, so muss der Source für den Meldungsmodul mit KDCMMOD neu erzeugt und übersetzt werden (entfallene Meldungen und geänderte Meldungsziele sind zu berücksichtigen)
- Binden des Anwendungsprogramms
- Übertragen der Anwenderdaten mit dem KDCUPD von openUTM (BS2000) V7.0A in die neue KDCFILE, wenn die Daten relevant sind

KDCUPD von openUTM (BS2000) V7.0A unterstützt folgende Versionsübergänge:

openUTM (BS2000) V6.4A ==> openUTM (BS2000) V7.0A

openUTM (BS2000) V6.5A ==> openUTM (BS2000) V7.0A

openUTM (BS2000) V7.0A ==> openUTM (BS2000) V7.0A

Mit openUTM (BS2000) V7.0 wird nur noch eine Variante ausgeliefert. In dieser Variante steht die Verschlüsselungsfunktionalität standardmäßig zur Verfügung.

Über die Generierung bzw. zum Ablaufzeitpunkt kann entschieden werden, ob die Verschlüsselungsfunktionalität zum Einsatz kommen soll oder nicht.

#### 3.4.2 Hinweise zur Umstellung von /390 auf x86

Wird eine UTM-Anwendung von einem BS2000-System /390 auf ein BS2000-System x86 verlagert, dann sind evtl. die für diese Anwendung generierten Timer-Werte anzupassen.

Dies gilt insbesondere für die Timer zur Überwachung der Laufzeit von Teilprogrammen, die mit dem Parameter TAC TIME generiert sind. Nach einer Umstellung sollten die dort generierten Werte zur CPU- und zur Realzeitüberwachung kontrolliert und evtl. angepasst werden.

### 3.4.3 Optionale Reps

Mit openUTM (BS2000) V7.0A werden z.Z. keine optionalen REPs ausgeliefert.

## 3.5 Entfallene und gekündigte Funktionen

### 3.5.1 Entfallene Funktionen

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:

- 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*

### 3.5.2 Gekündigte Schnittstellen

- UTM-Cluster  
Auf BS2000-Systemen werden UTM-Cluster-Anwendungen nicht mehr unterstützt.
- Fujitsu Software openUTM WinAdmin  
Mit dieser Version wird openUTM WinAdmin letztmalig unterstützt.

## 3.6 Inkompatible Änderungen

Rückgabelänge KCRLM bei MGET mit KCSRCC 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

## 3.7 Einschränkungen

keine

### 3.8 Verhalten im Fehlerfall

Im Fehlerfall werden folgende Diagnoseunterlagen benötigt:

- Genaue Beschreibung der Fehlersituation und Angabe, ob und wie der Fehler reproduzierbar ist
- BS2000-Version mit Korrekturstand und Laderkennzeichen
- openUTM-spezifische Unterlagen:
  - UTM-Version mit Korrekturstand
  - Liste der Steueranweisungen für KDCDEF
  - bei Warmstartfehlern bzw. Fehlern in KDCUPD und bei fehlerhaftem Anwendungsende mit PMIO20/PMIO22 werden die KDCFILE-Dateien benötigt
  - Binderlisten des Anwendungsprogramms
  - DBL-Liste der Anwendung (dynamischer Bindelader)
  - Übersetzungslisten betroffener Teilprogramme
  - beim Reproduzieren von Fehlern TESTMODE=ON einschalten, wobei TRACE-REC mindestens auf 3000 (10000 bei UTM-D-Einsatz) eingestellt sein sollte
  - UTM-Dumps, wobei darauf zu achten ist, dass die Dumps von allen Tasks, die zu einem Fehlerzeitpunkt tätig waren, mitgeliefert werden
  - SYSLOG-Datei
  - Userdumps bei z.B. XT58/XT5C etc.
  - SYSLST-/SYSOUT-Protokolle
- Bei Fehlern in Zusammenhang mit FHS zusätzlich:
  - Angabe der eingesetzten FHS-Version
  - Formatausdruck mit IFG
  - einsatzvorbereitetes Formatmodul in Dateiform bzw. Ausdruck des Formats mit LMS oder DPAGE
  - Benutzerdump zum Fehlerzeitpunkt in Dateiform
- Bei Fehlern in Zusammenarbeit mit Datenbanken siehe entsprechende Freigabemittlung des Datenbanksystems
- Bei Fehlern mit KDCUPD:
  - SYSOUT-/SYSLST-Protokoll zum KDCUPD-Lauf
  - UTM-Dump, wenn einer erzeugt wurde
  - KDCDEF-Steueranweisungen für alte und neue KDCFILE wenn aus Datenschutzgründen möglich
  - alte KDCFILE
  - neue KDCFILE, im Zustand vor dem KDCUPD-Lauf, oder KDCDEF-Eingabedaten für neue KDCFILE als Datei
- Bei Fehlern in Zusammenarbeit mit RSO sind die Hinweise zur OLTP-Schnittstelle im Handbuch sowie der Freigabemittlung der verwendeten RSO-Version zu beachten
- Bei verteilter Verarbeitung über LU6.1:
  - Evtl. BCAM-Trace (PORT-Trace)
  - Evtl. UTM-BCAM-Trace (BTRACE)
- Bei heterogener Kopplung Generierungsinformation und Trace von openUTM LU6.2 und seiner SNA Komponenten und vom IBM-SNA System
- Bei Kommunikation über OSI TP:  
Einschalten des OSS-Trace in der UTM-Anwendung
- Bei Kommunikation mit HTTPS-Clients:
  - User-Dumps des UTM SSL-Proxy Prozesses
  - SYSLST- und SYSOUT-Protokoll des UTM SSL-Proxy Prozesses

## 4 Hardware-Unterstützung

openUTM (BS2000) V7.0A ist auf allen Zentraleinheiten einsetzbar, die durch folgende BS2000-Versionen unterstützt werden:

BS2000 OSD/BC ab V10.0 und V11.0 (/390- und x86-Architektur)

Eine Liste aller von openUTM (BS2000) V7.0A unterstützten Geräte finden Sie im Handbuch "openUTM V7.0 Anwendungen generieren".