

Deutsch



FUJITSU Software

openUTM V6.3

Meldungen, Test und Diagnose in Unix- und Windows-Systemen

Benutzerhandbuch

Ausgabe Januar 2015

Kritik... Anregungen... Korrekturen...

Die Redaktion ist interessiert an Ihren Kommentaren zu diesem Handbuch. Ihre Rückmeldungen helfen uns, die Dokumentation zu optimieren und auf Ihre Wünsche und Bedürfnisse abzustimmen.

Sie können uns Ihre Kommentare per E-Mail an manuals@ts.fujitsu.com senden.

Zertifizierte Dokumentation nach DIN EN ISO 9001:2008

Um eine gleichbleibend hohe Qualität und Anwenderfreundlichkeit zu gewährleisten, wurde diese Dokumentation nach den Vorgaben eines Qualitätsmanagementsystems erstellt, welches die Forderungen der DIN EN ISO 9001:2008 erfüllt.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright und Handelsmarken

Copyright © 2015 Fujitsu Technology Solutions GmbH.

Alle Rechte vorbehalten.

Liefermöglichkeiten und technische Änderungen vorbehalten.

Alle verwendeten Hard- und Softwarenamen sind Handelsnamen und/oder Warenzeichen der jeweiligen Hersteller.

Inhalt

1	Einleitung	9
1.1	Zielgruppe und Konzept des Handbuchs	11
1.2	Wegweiser durch die Dokumentation zu openUTM	12
1.2.1	openUTM-Dokumentation	12
1.2.2	Dokumentation zum openSEAS-Produktumfeld	17
1.2.3	Readme-Dateien	18
1.3	Neuerungen in openUTM V6.3	19
1.3.1	Neue Server-Funktionen	19
1.3.2	Last-Simulation mit "Workload Capture & Replay"	22
1.3.3	Neue Client-Funktion	23
1.3.4	Neue und geänderte Funktionen für openUTM WinAdmin	23
1.3.5	Neue Funktionen für openUTM WebAdmin	23
1.4	Darstellungsmittel	25
2	Test und Fehlerdiagnose	27
2.1	Testen von UTM-Anwendungen	27
2.1.1	Testen einer UTM-Anwendung unter Unix-Systemen	29
2.1.1.1	Debuggen einer UTM-Anwendung unter Unix-Systemen	29
2.1.1.2	Starten einer UTM-Anwendung unter Unix-Systemen mit Debugger	30
2.1.2	Testen einer UTM-Anwendung unter Windows-Systemen	33
2.1.2.1	Debuggen einer UTM-Anwendung unter Windows-Systemen	33
2.1.2.2	Starten einer UTM-Anwendung unter Windows-Systemen mit Debugger	33
2.1.3	Ausgabe von Meldungen beim Start eines Prozesses	36
2.1.4	Arbeiten am Terminal im Testbetrieb	36
2.2	Fehlerdiagnose	39
2.2.1	Returncodes an der Programmschnittstelle	39
2.2.2	UTM-Meldungen bei Programmfehlern	40
2.2.3	Diagnose-Dump bei festgelegten Meldungen/Ereignissen	41
2.2.4	Fehlerunterlagen erstellen	44

2.2.5	Traces	46
2.2.5.1	Dynamischer openUTM -Trace über Umgebungsvariable	46
2.2.5.2	Protokollierung von Teilprogramm aufrufen	47
2.2.5.3	BCAM-Trace in openUTM	48
2.2.5.4	KTA-Trace in Datei	52
2.2.5.5	OSS-Trace	53
2.2.5.6	ADMI-Trace	55
2.2.5.7	Erzeugen eines core bei Anwendungsabbruch	55
2.2.5.8	Unterdrückung von gcore-Speicherabzügen	56
2.2.5.9	Tool KDCIPC	56
2.2.5.10	Tool KDCKAA	57
3	Der UTM-Dump	59
3.1	Die Dateien des UTM-Dumps	60
3.2	Das Tool KDCDUMP	62
3.2.1	KDCDUMP starten	62
3.2.2	KDCDUMP-Anweisungen	63
	! Systemkommando eingeben	64
	!! Zuletzt ausgeführtes Systemkommando wiederholen	65
	AFIND Adresse im Dump suchen	65
	Blätteranweisungen für Dialog-Aufbereitung	65
	DUMP UTM-Dump in den Speicher einlesen	67
	EDT Editor aufrufen	68
	END KDCDUMP beenden	69
	FGG Alle Dateien einer FGG aufbereiten	69
	FILE Einzelne Dump-Datei aufbereiten	70
	FIND Tabelleneintrag suchen und anzeigen	72
	HELP Hilfe zu KDCDUMP	73
	LIST Tabellenausschnitt aufbereiten	74
	SFIND String suchen	75
	SH und SYS KDCDUMP unterbrechen	77
	SYSLST Protokollierung ein/ausschalten	77
	TABLE Tabelle anzeigen	78
3.2.3	Meldungen von KDCDUMP	80
3.3	Inhalt des UTM-Dumps	81
3.3.1	Anwendungsglobaler Systemspeicher (KAA)	82
3.3.1.1	Die Tabelle CONS_ENTRIES	85
3.3.1.2	CACHE-Buffer	86
3.3.1.3	UTM SLOT POOLS	86
3.3.2	Anwendungsglobaler Systemspeicher von XAP-TP	87

3.3.3	Der Prozess-spezifische Systemspeicher (KTA)	88
3.3.4	Prozess-spezifischer Systemspeicher von XAP-TP	89
3.3.5	Der KDCROOT-Bereich	90
3.3.5.1	PROGRAM-Tabelle	94
3.3.5.2	LOAD-MODULE-Tabelle	95
3.3.5.3	UTM-DIAGAREA	97
3.3.5.4	DB-DIAGAREA	107
3.3.5.5	ADMI-DIAGAREA	112
3.3.5.6	ADMI-USERAREA	116
3.3.5.7	Der Kommunikationsbereich KB	117
3.3.6	Speicherbereiche in UTM-Cluster-Anwendungen	120
3.3.7	Summary	121
4	UTM-Meldungswesen	123
4.1	Meldungsmodul, Meldungsdefinitionsdatei	125
4.2	NLS-Meldungskataloge	126
4.2.1	Meldungskatalog-Sourcedatei für NLS	127
4.3	Meldungsziele	128
4.3.1	Ausgabeform der Meldungen	129
4.3.2	UTM-Meldungen an die Konsole	130
4.3.3	UTM-Meldungen an eine TS-Anwendung	130
4.3.4	UTM-Meldungen an Benutzer-spezifische Meldungsziele	131
4.3.5	UTM-Meldungen an MSGTAC	131
4.4	Aufbereitung der Meldungen durch openUTM	132
4.5	Gestaltung der Meldungsausgabe durch den Anwender	134
4.5.1	Meldungen in anderen Sprachen - Tool KDCMTXT	135
4.5.1.1	Aufruf von KDCMTXT	136
4.5.1.2	KDCMTXT-Steueranweisungen	136
4.5.1.3	Protokoll von KDCMTXT	140
4.5.2	Erstellen eines eigenen Meldungsmoduls mit KDCMMOD	141
4.5.2.1	Aufruf von KDCMMOD	144
4.5.2.2	Steueranweisungen von KDCMMOD	144
4.5.3	NLS-Meldungskataloge unter Windows-Systemen	152
4.6	UTM-Protokolldatei SYSLOG	153
4.6.1	SYSLOG-Datei auswerten	153
4.6.1.1	Das Tool KDCCSYSL - SYSLOG-Datei aufbereiten	154
4.6.1.2	Das Tool KDCPSYSL - Meldungstexte einfügen	155
4.6.1.3	Meldungen von KDCCSYSL und KDCPSYSL	155

4.7	Aufbau der UTM-Systemmeldungen	156
5	UTM-Meldungen	157
5.1	Meldungen des Transaktionsmonitors	157
5.2	Meldungen des XAP-TP-Providers	315
5.2.1	Allgemeine Inserts der XAP-TP-Meldungen	333
5.3	Meldungen des Generierungstools KDCDEF	342
5.4	Meldungen des UTM-Tools KDCPSYSL	349
5.5	Meldungen der UTM-Tools KDCMMOD / KDCMTXT	350
5.6	Meldungen des UTM-Tools KDCDUMP	352
5.7	Meldungstexte des UTM-Tools KDCUPD	355
5.8	U-Meldungen	357
5.8.1	Meldungen des Dialog-Terminalprozesses	357
5.8.2	Meldungen des Printerprozesses	359
5.8.3	Meldungen des utmlog-Prozesses	360
5.8.4	Allgemeine U-Meldungen	361
5.8.5	Meldungen des Timerprozesses	364
5.8.6	Meldungen des utmmain-Prozesses	365
5.8.7	Meldungen des Dienstprogramms kdcuslog und kdclog	369
5.8.8	Meldungen des Dienstprogramms kdccsysl	370
5.8.9	Meldungen des Netzprozesses	371
5.8.10	Meldungen von kdckaa	380
5.8.11	Meldungen des UTM-Tools kdcshut	380
5.8.12	Meldungen des UTM-Tools kdcrem	381
5.8.13	Meldungen des UTM-Tools kdcprog	381
5.9	Fehlercodes bei Dateibearbeitung (DMS-Fehler)	383
5.10	Standard-Meldungsdefinitionsdatei, Inserts	385
5.10.1	Konstanten der Standard-Meldungsdefinitionsdatei	385
5.10.2	Inserts in Meldungen	385
5.10.2.1	Inserts in K-und P-Meldungen	386
5.10.2.2	Inserts in U-Meldungen	395
5.11	Ziele der UTM-Meldungen	398
5.12	Windows Event Logging-Meldungen	416

6	UTM-Returncodes	419
6.1	KDCS-Returncodes in KCRCCC	419
6.2	Interner Returncode KCRCDC	422
	Fachwörter	445
	Abkürzungen	483
	Literatur	489
	Stichwörter	499

1 Einleitung

Moderne unternehmensweite IT-Umgebungen unterliegen zahlreichen Herausforderungen von zunehmender Brisanz. Dies wird verursacht durch

- heterogene Systemlandschaften
- unterschiedliche HW-Plattformen
- unterschiedliche Netze und Netzzugriffe (TCP/IP, SNA, ...)
- Verflechtung der Anwendungen mit den Unternehmen

Dadurch entwickeln sich Problemfelder, sei es bei Fusionen, durch Kooperationen oder auch nur durch Rationalisierungsmaßnahmen. Die Unternehmen fordern flexible und skalierbare Anwendungen, gleichzeitig soll die Transaktionssicherheit für Prozesse und Daten gewährleistet bleiben, obwohl die Geschäftsprozesse immer komplexer werden. Die wachsende Globalisierung geht selbstverständlich davon aus, dass Anwendungen im 7x24-Stunden-Betrieb laufen und hochverfügbar sind, um beispielsweise Internetzugriffe auf bestehende Anwendungen über Zeitzonen hinweg zu ermöglichen.

Die High-End-Plattform für Transaktionsverarbeitung openUTM bietet eine Ablaufumgebung, die all diesen Anforderungen moderner unternehmenskritischer Anwendungen gewachsen ist, denn openUTM verbindet alle Standards und Vorteile von transaktionsorientierten Middleware-Plattformen und Message Queuing Systemen:

- Konsistenz der Daten und der Verarbeitung
- Hohe Verfügbarkeit der Anwendungen (nicht nur der Hardware)
- Hohen Durchsatz auch bei großen Benutzerzahlen, d.h. höchste Skalierbarkeit
- Flexibilität bezüglich Änderungen und Anpassungen des IT-Systems

Eine UTM-Anwendung kann auf einem einzelnen Rechner als stand-alone UTM-Anwendung oder auf mehreren Rechnern gleichzeitig als UTM-Cluster-Anwendung betrieben werden.

openUTM ist Teil des umfassenden Angebots von **openSEAS**. Gemeinsam mit der Oracle Fusion Middleware bietet openSEAS die komplette Funktionalität für Anwendungsinnovation und moderne Anwendungsentwicklung. Im Rahmen des Produktangebots **openSEAS** nutzen innovative Produkte die ausgereifte Technologie von openUTM:

- BeanConnect ist ein Adapter gemäß der Java EE Connector Architecture (JCA) von Oracle/Sun und bietet den standardisierten Anschluss von UTM-Anwendungen an Java EE Application Server. Dadurch können bewährte Legacy-Anwendungen in neue Geschäftsprozesse integriert werden.
- Mit WebTransactions steht in openSEAS ein Produkt zur Verfügung, welches es ermöglicht, bewährte Host-Anwendungen flexibel in neuen Geschäftsprozessen und modernen Einsatzszenarien zu nutzen. Bestehende UTM-Anwendungen können unverändert ins Web übernommen werden.

1.1 Zielgruppe und Konzept des Handbuchs

Das openUTM-Handbuch „Meldungen, Test und Diagnose in Unix- und Windows-Systemen“ richtet sich an Anwender, Administratoren und Programmierer von UTM-Anwendungen.

Es beschreibt das Testen einer UTM-Anwendung unter Unix- und Windows-Systemen, den Aufbau des openUTM-Dumps, das Verhalten im Fehlerfall sowie alle von openUTM ausgegebenen Meldungen und Returncodes.

In Kapitel 4 ist außerdem das Meldungswesen von openUTM beschrieben mit den Möglichkeiten, Meldungen Benutzer-spezifisch in verschiedenen Sprachen auszugeben oder die von openUTM ausgelieferten Meldungen Anwendungs-spezifisch zu verändern oder an andere Meldungsziele auszugeben.

Voraussetzung für das Verständnis dieses Handbuchs sind Kenntnisse von openUTM und Unix-Betriebssystemen bzw. Windows-Betriebssystemen. Für BS2000-Systeme steht Ihnen ein eigenes openUTM-Handbuch „Meldungen, Test und Diagnose in BS2000/OSD“ zur Verfügung.



Wenn im Folgenden allgemein von Unix-System bzw. Unix-Plattform die Rede ist, dann ist darunter sowohl ein Unix-basiertes Betriebssystem wie z.B. Solaris oder HP-UX als auch eine Linux-Distribution wie z.B. SUSE oder Red Hat zu verstehen.

Wenn im Folgenden von Windows-System bzw. Windows-Plattform die Rede ist, dann sind damit alle Windows-Varianten gemeint, auf denen openUTM zum Ablauf kommt.

1.2 Wegweiser durch die Dokumentation zu openUTM

In diesem Abschnitt erhalten Sie einen Überblick über die Handbücher zu openUTM und zum Produktumfeld von openUTM.

1.2.1 openUTM-Dokumentation

Die openUTM-Dokumentation besteht aus Handbüchern, den Online-Hilfen für den grafischen Administrationsarbeitsplatz openUTM WinAdmin und das grafische Administrations-tool WebAdmin sowie einer Freigabemitteilung für jede Plattform, auf der openUTM freigegeben wird.

Es gibt Handbücher, die für alle Plattformen gültig sind, sowie Handbücher, die jeweils für BS2000-Systeme bzw. für Unix-Systeme und Windows-Systeme gelten.

Sämtliche Handbücher sind als PDF-Datei im Internet verfügbar unter der Adresse

<http://manuals.ts.fujitsu.com>

Geben Sie dort in das Feld **Produktsuche** den Suchbegriff "openUTM V6.3" ein, um sich alle openUTM-Handbücher der Version 6.3 anzeigen zu lassen.

Die Handbücher sind auf offenen Plattformen auf der Enterprise DVD enthalten und stehen für BS2000-Systeme auf der WinAdmin DVD zur Verfügung.

Die folgenden Abschnitte geben einen Aufgaben-bezogenen Überblick über die Dokumentation zu openUTM V6.3. Eine vollständige Liste der Dokumentation zu openUTM finden Sie im Literaturverzeichnis auf [Seite 489](#).

Einführung und Überblick

Das Handbuch **Konzepte und Funktionen** gibt einen zusammenhängenden Überblick über die wesentlichen Funktionen, Leistungen und Einsatzmöglichkeiten von openUTM. Es enthält alle Informationen, die Sie zum Planen des UTM-Einsatzes und zum Design einer UTM-Anwendung benötigen. Sie erfahren, was openUTM ist, wie man mit openUTM arbeitet und wie openUTM in die BS2000-, Unix- und Windows-Plattformen eingebettet ist.

Programmieren

- Zum Erstellen von Server-Anwendungen über die KDCS-Schnittstelle benötigen Sie das Handbuch **Anwendungen programmieren mit KDCS für COBOL, C und C++**, in dem die KDCS-Schnittstelle in der für COBOL, C und C++ gültigen Form beschrieben ist. Diese Schnittstelle umfasst sowohl die Basisfunktionen des universellen Transaktionsmonitors als auch die Aufrufe für verteilte Verarbeitung. Es wird auch die Zusammenarbeit mit Datenbanken beschrieben.
- Wollen Sie die X/Open-Schnittstellen nutzen, benötigen Sie das Handbuch **Anwendungen erstellen mit X/Open-Schnittstellen**. Es enthält die UTM-spezifischen Ergänzungen zu den X/Open-Programmschnittstellen TX, CPI-C und XATMI sowie Hinweise zu Konfiguration und Betrieb von UTM-Anwendungen, die X/Open-Schnittstellen nutzen. Ergänzend dazu benötigen Sie die X/Open-CAE-Specification für die jeweilige X/Open-Schnittstelle.
- Wenn Sie Daten auf Basis von XML austauschen wollen, benötigen Sie das Dokument **XML für openUTM**. Darin werden die C- und COBOL-Aufrufe beschrieben, die zum Bearbeiten von XML-Dokumenten benötigt werden.
- Für BS2000-Systeme gibt es Ergänzungsbände für die Programmiersprachen Assembler, Fortran, Pascal-XT und PL/1.

Konfigurieren

Zur Definition von Konfigurationen steht Ihnen das Handbuch **Anwendungen generieren** zur Verfügung. Darin ist beschrieben, wie Sie mit Hilfe des UTM-Tools KDCDEF sowohl für eine stand-alone UTM-Anwendung als auch für eine UTM-Cluster-Anwendung

- die Konfiguration definieren
- die KDCFILE erzeugen
- und im Falle einer UTM-Cluster-Anwendung die UTM-Cluster-Dateien erzeugen.

Zusätzlich wird gezeigt, wie Sie wichtige Verwaltungs- und Benutzerdaten mit Hilfe des Tools KDCUPD in eine neue KDCFILE übertragen, z.B. beim Umstieg auf eine neue Version von openUTM oder nach Änderungen in der Konfiguration. Für eine UTM-Cluster-Anwendung wird außerdem gezeigt, wie Sie diese Daten mit Hilfe des Tools KDCUPD in die neuen UTM-Cluster-Dateien übertragen.

Binden, Starten und Einsetzen

Um UTM-Anwendungen einsetzen zu können, benötigen Sie für das betreffende Betriebssystem (BS2000- bzw. Unix-/Windows-Systeme) das Handbuch **Einsatz von openUTM-Anwendungen**.

Dort ist beschrieben, wie man ein UTM-Anwendungsprogramm bindet und startet, wie man sich bei einer UTM-Anwendung an- und abmeldet und wie man Anwendungsprogramme strukturiert und im laufenden Betrieb austauscht. Außerdem enthält es die UTM-Kommandos, die dem Terminal-Benutzer zur Verfügung stehen. Zudem wird ausführlich auf die Punkte eingegangen, die beim Betrieb von UTM-Cluster-Anwendungen zu beachten sind.

Administrieren und Konfiguration dynamisch ändern

- Für das Administrieren von Anwendungen finden Sie die Beschreibung der Programm-schnittstelle zur Administration und die UTM-Administrationskommandos im Handbuch **Anwendungen administrieren**. Es informiert über die Erstellung eigener Administrationsprogramme für den Betrieb einer stand-alone UTM-Anwendung oder einer UTM-Cluster-Anwendung sowie über die Möglichkeiten, mehrere UTM-Anwendungen zentral zu administrieren. Darüber hinaus beschreibt es, wie Sie Message Queues und Drucker mit Hilfe der KDCS-Aufrufe DADM und PADM administrieren können.
- Wenn Sie den grafischen Administrationsarbeitsplatz **openUTM WinAdmin** oder die funktional vergleichbare Web-Anwendung **openUTM WebAdmin** einsetzen, dann steht Ihnen folgende Dokumentation zur Verfügung:
 - Die **WinAdmin-Beschreibung** und die **WebAdmin-Beschreibung** bieten einen umfassenden Überblick über den Funktionsumfang und das Handling von WinAdmin/WebAdmin. Die Dokumente werden jeweils mit der Software ausgeliefert und sind zusätzlich auch online als PDF-Datei verfügbar.
 - Das jeweilige **Online-Hilfesystem** beschreibt kontextsensitiv alle Dialogfelder und die zugehörigen Parameter, die die grafische Oberfläche bietet. Außerdem wird dargestellt, wie man WinAdmin bzw. WebAdmin konfiguriert, um stand-alone UTM-Anwendungen und UTM-Cluster-Anwendungen administrieren zu können.



Details zur Integration von openUTM WebAdmin in den SE Manager des SE Servers finden Sie im SE Server Handbuch **Bedienen und Verwalten**.

Testen und Fehler diagnostizieren

Für die o.g. Aufgaben benötigen Sie außerdem die Handbücher **Meldungen, Test und Diagnose** (jeweils ein Handbuch für Unix-/Windows-Systeme und für BS2000-Systeme). Sie beschreiben das Testen einer UTM-Anwendung, den Inhalt und die Auswertung eines UTM-Dumps, das Verhalten im Fehlerfall, das Meldungswesen von openUTM, sowie alle von openUTM ausgegebenen Meldungen und Returncodes.

openUTM-Clients erstellen

Wenn Sie Client-Anwendungen für die Kommunikation mit UTM-Anwendungen erstellen wollen, stehen Ihnen folgende Handbücher zur Verfügung:

- Das Handbuch **openUTM-Client für Trägersystem UPIC** beschreibt Erstellung und Einsatz von Client-Anwendungen, die auf UPIC basieren. Neben der Beschreibung der Schnittstellen CPI-C und XATMI erhalten Sie Informationen, wie Sie die C++-Klassen für die schnelle und einfache Programmerstellung nutzen können.
- Das Handbuch **openUTM-Client für Trägersystem OpenCPIC** beschreibt, wie man OpenCPIC installiert und konfiguriert. Es zeigt auf, was beim Programmieren einer CPI-C-Anwendung zu beachten ist und welche Einschränkungen es gegenüber der Programmschnittstelle X/Open CPI-C gibt.
- Für die mit **BeanConnect** ausgelieferten **JUpic-Java-Klassen** wird die Dokumentation mit der Software ausgeliefert. Diese Dokumentation besteht aus Word- und PDF-Dateien, die die Einführung und die Installation beschreiben, sowie aus einer Java-Dokumentation mit der Beschreibung der Java-Klassen.
- Das Handbuch **BizXML2Cobol** beschreibt, wie Sie bestehende Cobol-Programme einer UTM-Anwendung so erweitern können, dass sie als Standard-Web-Service auf XML-Basis genutzt werden können. Die Arbeit mit der grafischen Bedienoberfläche ist in der zugehörigen **Online-Hilfe** beschrieben.
- Wenn Sie UTM-Services auf einfache Weise ins Web stellen möchten, benötigen Sie das Handbuch **Web-Services für openUTM**. Das Handbuch beschreibt, wie Sie mit dem Software-Produkt WS4UTM (WebServices for openUTM) Services von UTM-Anwendungen als Web Services verfügbar machen. Die Arbeit mit der grafischen Bedienoberfläche ist in der zugehörigen **Online-Hilfe** beschrieben.

Kopplung mit der IBM-Welt

Wenn Sie aus Ihrer UTM-Anwendung mit Transaktionssystemen von IBM kommunizieren wollen, benötigen Sie außerdem das Handbuch **Verteilte Transaktionsverarbeitung zwischen openUTM und CICS-, IMS- und LU6.2-Anwendungen**. Es beschreibt die CICS-Kommandos, IMS-Makros und UTM-Aufrufe, die für die Kopplung von UTM-Anwendungen mit CICS- und IMS-Anwendungen benötigt werden. Die Kopplungsmöglichkeiten werden anhand ausführlicher Konfigurations- und Generierungsbeispiele erläutert. Außerdem beschreibt es die Kommunikation über openUTM-LU62, sowie dessen Installation, Generierung und Administration.

Dokumentation zu PCMX

Mit openUTM auf Unix- und Windows-Systemen wird die Kommunikationskomponente PCMX ausgeliefert. Die Funktionen von PCMX sind in folgenden Dokumenten beschrieben:

- Handbuch CMX (Unix-Systeme) "Betrieb und Administration" für Unix-Systeme
- Online-Hilfe zu PCMX für Windows-Systeme

1.2.2 Dokumentation zum openSEAS-Produktumfeld

Die Verbindung von openUTM zum openSEAS-Produktumfeld wird im openUTM-Handbuch **Konzepte und Funktionen** kurz dargestellt. Die folgenden Abschnitte zeigen, welche der openSEAS-Dokumentationen für openUTM von Bedeutung sind.

Integration von Java EE Application Servern und UTM-Anwendungen

Der Adapter BeanConnect gehört zur Produkt-Suite openSEAS. Der BeanConnect-Adapter realisiert die Verknüpfung zwischen klassischen Transaktionsmonitoren und Java EE Application Servern und ermöglicht damit die effiziente Integration von Legacy-Anwendungen in Java-Anwendungen.

- Das Handbuch **BeanConnect** beschreibt das Produkt BeanConnect, das einen JCA 1.5- und JCA 1.6-konformen Adapter bietet, der UTM-Anwendungen mit Anwendungen auf Basis von Java EE , z.B. mit dem Application Server von Oracle, verbindet.

Die Handbücher zum Application Server von Oracle sind bei Oracle beziehbar.

Web-Anbindung und Anwendungsintegration

Zum Anschließen neuer und bestehender UTM-Anwendungen an das Web mit dem Produkt WebTransactions benötigen Sie die Handbücher zu **WebTransactions**.

Die Dokumentation wird durch JavaDocs ergänzt.

1.2.3 Readme-Dateien

Funktionelle Änderungen und Nachträge der aktuellen Produktversion zu diesem Handbuch entnehmen Sie bitte ggf. den Produkt-spezifischen Readme-Dateien.

Readme-Dateien stehen Ihnen online bei dem jeweiligen Produkt zusätzlich zu den Produkthandbüchern unter <http://manuals.ts.fujitsu.com> zur Verfügung. Für die Plattform BS2000 finden Sie Readme-Dateien auch auf der Softbook-DVD.

Ergänzende Produkt-Informationen

Aktuelle Informationen, Versions-, Hardware-Abhängigkeiten und Hinweise für Installation und Einsatz einer Produktversion enthält die zugehörige Freigabemitteilung. Solche Freigabemitteilungen finden Sie online unter <http://manuals.ts.fujitsu.com>.

Readme-Datei unter Unix-Systemen

Die Readme-Datei und ggf. weitere Dateien wie z.B. eine Handbuchergänzungsdatei finden Sie im *utmpfad* unter */docs/sprache*.

Readme-Datei unter Windows-Systemen

Die Readme-Datei und ggf. weitere Dateien wie z.B. eine Handbuchergänzungsdatei finden Sie im *utmpfad* unter *\Docs\sprache*.

1.3 Neuerungen in openUTM V6.3

Die folgenden Abschnitte gehen näher auf die Neuerungen in den einzelnen Bereichen ein.

1.3.1 Neue Server-Funktionen

Zusätzliche UTM-System-Prozesse für interne Aufgaben

UTM startet zusätzlich zu den per Startparameter angegebenen Prozessen bis zu drei weitere Prozesse, die für interne Aufgaben von openUTM oder privilegierte Aufträge des Administrators freigehalten werden.

Dazu wurden die Generierungs- und die Administrationschnittstelle erweitert:

- Generierung, KDCDEF-Anweisung MAX
 - Neuer Operand PRIVILEGED-LTERM, um ein bestimmtes LTERM als privilegiert auszuzeichnen. Durch die Anmeldung eines Benutzer mit Administrationsberechtigung werden alle Aufträge des Benutzers zu privilegierten Aufträgen.
 - Operand TASKS: Der Maximalwert wurde wegen der zusätzlichen System-Prozesse auf 240 reduziert.
- Administrationsschnittstelle KDCADMI
 - Datenstruktur *kc_max_par_str*: Neues Feld *privileged_lterm* für das generierte privilegierte LTERM.
 - Datenstruktur *kc_tasks_par_str*: Neue Felder *gen_system_tasks* und *curr_system_tasks* für die System-Prozesse.
 - Datenstruktur *kc_curr_par_str*: Neues Feld *curr_system_tasks* für die System-Prozesse.

Höhere Auflösung der verbrauchten CPU-Zeit

Die verbrauchte CPU-Zeit wird für TACs jetzt in Mikrosekunden und für USERS in Millisekunden ausgegeben. Dazu wurden folgende Schnittstellen geändert:

- KDCADMI
 - Datenstruktur *kc_tac_str*: Neues Feld *taccpu_micro_sec* für die durchschnittlich verbrauchte CPU-Zeit in Mikrosekunden.
 - Datenstrukturen *kc_user_str* und *kc_user_dyn1_str*: Neues Feld *cputime_msec* für die verbrauchte CPU-Zeit in Millisekunden.

- Kommando-Schnittstelle KDCADM
 - KDCINF type=TAC: TACCPU gibt die durchschnittlich verbrauchte CPU-Zeit in Mikrosekunden aus.
 - KDCINF type=USER: CPUTIME gibt die verbrauchte CPU-Zeit in Millisekunden aus.
- KDCEVAL-Listen
 - In den KDCEVAL-Listen werden einige Zeiten jetzt in Mikrosekunden ausgegeben.

Neue Trace-Funktionen

Im laufenden Betrieb können zusätzliche Traces ein- und ausgeschaltet werden:

- ADMI Trace, d.h. Trace der Programmschnittstelle zur Administration (KDCADMI)
- X/Open Traces (CPI-C, TX, XATMI)

Dazu wurden folgende Schnittstellen erweitert:

- Startparameter:
 - Neue Startparameter ADMI-TRACE, CPIC-TRACE, TX-TRACE und XATMI-TRACE zum Einschalten der Traces.
- KDCADMI:
 - Datenstruktur *kc_diag_and_account_par_str*: Neue Felder *admi_trace*, *cpic_trace*, *tx_trace* und *xatmi_trace* zum Ein- und Ausschalten der Traces.

KDCDEF-Ein-/Ausgabe über LMS-Bibliothekselemente

In BS2000-Systemen können KDCDEF-Anweisungen aus LMS-Bibliothekselementen gelesen und beim inversen KDCDEF in LMS-Bibliothekselemente ausgegeben werden. Dazu wurden folgende Schnittstellen erweitert:

- Generierung
 - KDCDEF-Anweisung OPTION: Neuer Operandenwert LIBRARY-ELEMENT(...) beim Operanden DATA.
 - KDCDEF-Anweisung CREATE-CONTROL-STATEMENTS: Neuer Operandenwert LIBRARY-ELEMENT(...) beim Operanden TO-FILE.
- KDCADMI
 - Datenstruktur *kc_create_statements_str*: Neue Felder *lib_name*, *elem_name*, *vers*, *type*, *stmt_type* und *file_error_code*.

- Meldungen

Neue Meldungen K234, K519 und K520 beim Lesen von KDCDEF-Anweisungen aus LMS-Bibliothekselementen und beim Ausgeben von KDCDEF-Anweisungen in LMS-Bibliothekselemente.

Performanceverbesserungen

- UTM-Cache

Der UTM-Cache wurde optimiert, um die Performance bei intensiver Nutzung des UTM-Cache (z.B. bei sehr umfangreichen Vorgangsdaten) zu verbessern.

- UTM-Lock Algorithmus

Für konkurrierende Zugriffe auf UTM-interne Verwaltungsdaten wird auf offenen Plattformen durchgängig die vom Betriebssystem angebotene Compare&Swap Funktionalität verwendet.

- UTM-Netzanbindung

Die Netzanbindung auf offenen Plattformen wurde dahingehend verbessert, dass insbesondere bei Niederlast beim Senden von Daten an UTM-Partneranwendungen keine Verzögerungen mehr auftreten.

Sonstige Änderungen

- Meldungen

- Der Meldungsbereich für Systemmeldungen wurde vergrößert und umfasst jetzt den Bereich von K001 bis K399 (bisher bis K249). Damit haben sich folgende Meldungsbereiche verschoben:
 - Die Meldungsnummern für Meldungen, die ausschließlich von KDCUPD ausgegeben werden, belegen jetzt den Bereich von K800 bis K899 statt K250 bis K322.
Meldungen, die sowohl von KDCUPD als auch vom Online-Import ausgegeben werden, gelten als Systemmeldungen und bleiben unverändert.
 - Die Meldungsnummern für KDCCSYSL- und KDCCPSYSL-Meldungen belegen jetzt den Bereich K600 bis K649 statt K550 bis K599.
- Neue Meldung K235, falls die Namensauflösung für einen Rechner zu lange dauert.
- Bei den Meldungen K162 und K163 wurden die Standard-Meldungsziele geändert.

- KDCADMI
 - Die Felder *auto_connect* bei *kc_lpap_str* und *auto_connect_number* bei *kc_osi_lpap_str* besitzen die Eigenschaft GPD statt PD, d.h. Änderungen für diese Felder wirken nun Anwendungs-global. Eine administrative Änderung der Eigenschaften "Automatischer Verbindungsaufbau" bei LPAP und "Anzahl der Verbindungen" bei OSI-LPAP wirkt über das Anwendungsende hinaus.
 - Neues Feld *max_btrace_lth* bei *kc_diag_and_account_par_str* für die maximale Länge der Aufzeichnungsdaten bei eingeschalteter BCAM-Trace-Funktion.
- Für Plattformen, auf denen UTM im 64-Bit Mode laufen kann, ermöglicht KDCUPD den Umstieg von einer 32-Bit-Anwendungsumgebung auf eine 64-Bit-Anwendungsumgebung. Derzeit unterstützt UTM den 64-Bit-Modus nur auf Unix-Plattformen.
- Die Oracle User-Id kann bei den KDCDEF-Anweisungen DATABASE und RMXA auch in Kleinbuchstaben angegeben werden.
- Auf Windows-Systemen wird das Installationsverfahren InstallAware verwendet. Daher wird openUTM auf Windows-Systemen in Form von MSI-Dateien ausgeliefert.
- Neues Beispielprogramm ADJTCLT (ADJust Tac-CLass Table)

Mit dem C-Teilprogramm ADJTCLT kann der Anwender steuern, wie die Prozesse auf die TAC-Klassen aufgeteilt werden, und zwar abhängig von der aktuellen Anzahl aller Prozesse und der aktuellen Anzahl der Asynchron-Prozesse. Dazu erstellt der Anwender eine Tabelle mit den gewünschten Einstellungen. Die Einstellungen müssen so gewählt werden, dass immer mindestens ein Prozess frei ist, um andere Aufgaben, z.B. Transaktionsende-Verarbeitung von verteilten Transaktionen, zu erledigen.

1.3.2 Last-Simulation mit "Workload Capture & Replay"

Mit der neuen Funktion Workload Capture & Replay kann die Kommunikation von UTM-Anwendungen mit UPIC-Clients mitgeschnitten und anschließend mit einstellbaren Lastprofilen abgespielt werden. Damit lässt sich das Verhalten der UTM-Anwendung bei hoher Last unter realen Bedingungen testen.

Workload Capture & Replay besteht aus folgenden Komponenten:

- *UPIC Capture*: schneidet die Kommunikation mit dem UPIC-Client mit.

Zum Mitschneiden einer UPIC-Session (Capture) wird die Trace-Funktion BTRACE (BCAM-Trace) verwendet, die auf allen Server-Plattformen vorhanden ist.
- *UPIC Analyzer*: dient zur Analyse der mitgeschnittenen Kommunikation.
- *UPIC Replay*: dient zum Abspielen der mitgeschnittenen UPIC-Session mit unterschiedlichen Lastparametern (Geschwindigkeit, Client-Anzahl).

UPIC Analyzer und *UPIC Replay* stehen nur auf 64-Bit-Linux-Systemen zur Verfügung und sind Lieferbestandteil des openUTM-Client (UPIC).

Zusätzlich wird mit openUTM auf Unix- und Windows-Systemen das Dienstprogramm *kdcsort* ausgeliefert. Mit *kdcsort* können Sie die von BTRACE mitgeschnittene Kommunikation zeitlich sortieren, wenn die UTM-Anwendung beim Mitschneiden mit mehr als einem Prozess gelaufen ist und deshalb mehrere prozess-spezifische Dateien erzeugt wurden.

1.3.3 Neue Client-Funktion

Der UPIC-Client steht auf Windows-Systemen neben der 32-Bit Variante zusätzlich in einer 64-Bit Variante zur Verfügung.

1.3.4 Neue und geänderte Funktionen für openUTM WinAdmin

- WinAdmin unterstützt alle Neuerungen der UTM V6.3 bzgl. der Programmschnittstelle zur Administration. Dazu gehören z.B. die neuen Trace-Funktionen, das Schreiben von KDCDEF-Anweisungen in Bibliothekselemente beim Ablauf des inversen KDCDEF im BS2000 oder die Anzeige der verbrauchten CPU-Zeit eines Users in Millisekunden.
- Einführung einer Lebensdauer für Statistikwerte, um die Anzahl der in der Konfigurationsdatenbank gespeicherten Statistikwerte zu beschränken.

1.3.5 Neue Funktionen für openUTM WebAdmin

Zusatzfunktionen

WebAdmin bietet weitere Zusatzfunktionen, die über die Funktionalität der Administrationschnittstelle KDCADMI hinausgehen und bisher nur in WinAdmin zur Verfügung standen:

- Message Queues anzeigen (DADM-Funktionalität)
- Statistikkollektoren verwalten und deren Werte tabellarisch anzeigen (einschließlich der neuen Funktion "Lebensdauer für Statistikwerte")
- Statistiken in grafischer Form darstellen (Verlaufsgrafik)
- Schwellwert-Aktionen für Statistikkollektoren ausführen

Unterstützung der Neuerungen in openUTM V6.3

WebAdmin unterstützt alle Neuerungen von UTM V6.3 bzgl. der Programmschnittstelle zur Administration. Dazu gehören z.B. die neuen Trace-Funktionen, das Schreiben von KDCDEF-Anweisungen in Bibliothekselemente beim Ablauf des Inversen KDCDEF im BS2000 oder die Anzeige der verbrauchten CPU-Zeit eines Users in Millisekunden.

Integration in den SE Server

WebAdmin kann auf der Management Unit (SE Manager) eines SE Servers als Add-on installiert werden und bietet dann im Wesentlichen den selben Funktionsumfang wie bei einem Betrieb außerhalb des SE Managers.

1.4 Darstellungsmittel

Metasyntax

Die in diesem Handbuch verwendete Metasyntax können Sie der folgenden Tabelle entnehmen:

Formale Darstellung	Erläuterung	Beispiel
GROSSBUCHSTABEN	Großbuchstaben bezeichnen Konstanten (Namen von Aufrufen, Anweisungen, Feldnamen, Kommandos und Operanden etc.), die in dieser Form anzugeben sind.	LOAD-MODE=STARTUP
kleinbuchstaben	In Kleinbuchstaben sind in Syntaxdiagrammen und Operandenbeschreibung die Platzhalter für Operandenwerte dargestellt.	KDCFILE=filebase
<i>kleinbuchstaben</i>	Im Fließtext werden Variablen, Namen von Datenstrukturen und Feldern, Schlüsselwörter (z.B. C-Befehle, Unix- und Windows-Dateinamen etc.) in kursiven Kleinbuchstaben dargestellt.	Für COBOL sind die Datenstrukturen im COPY-Element KCINIC definiert, Datenstrukturen für C/C++ in der Include-Datei <i>kcini.h</i> .
Schreibmaschinenschrift	Im Fließtext werden Kommandos, Dateinamen, Meldungen und Beispiele in Schreibmaschinenschrift ausgezeichnet, die in genau dieser Form eingegeben werden müssen bzw. die genau diesen Namen oder diese Form besitzen.	Der Aufruf <code>tpcall</code>
{ } und	In geschweiften Klammern stehen alternative Angaben, von denen Sie eine auswählen müssen. Die zur Verfügung stehenden Alternativen werden jeweils durch einen Strich getrennt aufgelistet.	STATUS={ ON OFF }
[]	In eckigen Klammern stehen wahlfreie Angaben, die entfallen können.	KDCFILE=(filebase [, { SINGLE DOUBLE}])

Formale Darstellung	Erläuterung	Beispiel
()	Kann für einen Operanden eine Liste von Parametern angegeben werden, sind diese in runde Klammern einzuschließen und durch Kommata zu trennen. Wird nur ein Parameter angegeben, kann auf die Klammern verzichtet werden.	KEYS=(key1, key2, ... keyn)
<u>Unterstreichen</u>	Unterstreichen kennzeichnet den Standardwert.	CONNECT= { A/YES <u>NO</u> }
Kurzform	Die Standardkurzform für Anweisungen, Operanden und Operandenwerte wird „fett“ hervorgehoben. Die Kurzform kann alternativ angegeben werden.	TRANSPORT- SELECTOR =c 'C'
...	Punkte zeigen die Wiederholbarkeit einer syntaktischen Einheit an. Außerdem kennzeichnen die Punkte Ausschnitte aus einem Programm, einer Syntaxbeschreibung o.ä.	KDCDEF starten : : OPTION DATA=statement_file : END

Sonstige Symbole



für Hinweistexte.



für Warnhinweise.

utmpfad

bezeichnet unter Unix- und Windows-Systemen das Verzeichnis, unter dem openUTM installiert wurde.

2 Test und Fehlerdiagnose

In diesem Kapitel erfahren Sie

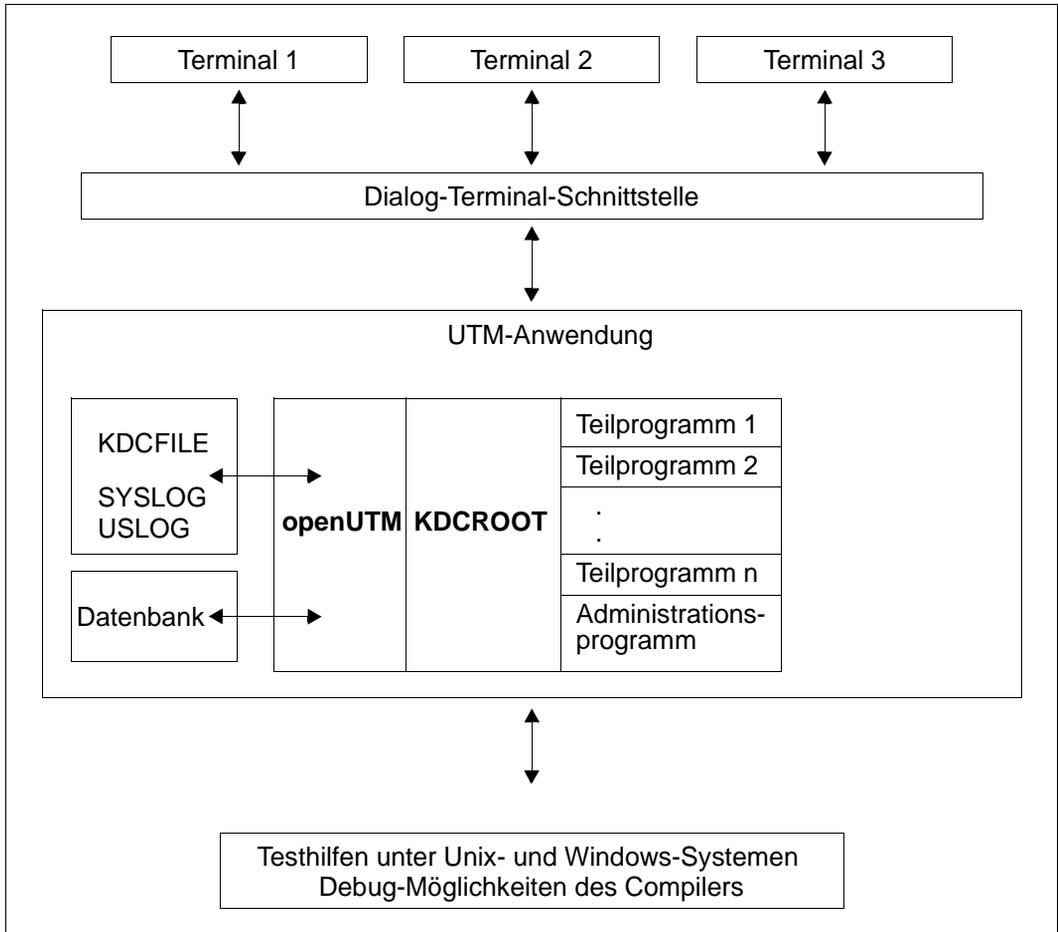
- wie Sie eine UTM-Anwendung testen
- wie Sie bei der Fehlerdiagnose vorgehen können
- welche Traces Sie zur Diagnose einsetzen können

2.1 Testen von UTM-Anwendungen

Für die meisten Testzwecke reicht es aus, die Anwendung ganz normal zu starten. Nur wenn Sie ein Teilprogramm debuggen wollen, das beim Start der Anwendung automatisch abläuft, z.B. Teilprogramme für einen Start-Exit oder das MSGTAC-Programm für Start-Meldungen, müssen Sie die Anwendung unter der Kontrolle eines Test-Tools starten. In beiden Fällen gilt:

- Zum Testen stehen Ihnen verschiedene Testtools zur Verfügung unter Unix-Systemen zum Beispiel *dbx*, *sdb*, *adb*, *xdb*, *debug*, *gdb*, Cobol-Testhilfe und unter Windows-Systemen der in Microsoft Visual Studio integrierte Debugger. Es ist möglich, sich mit dem Testtool an die Workprozesse der Anwendung anzuhängen oder mehrere Workprozesse unter der Kontrolle eines Test-Tools zu starten.
- Sie benötigen keine besondere Generierung dafür, d.h. Sie können das getestete Anwendungsprogramm auch im Produktivbetrieb einsetzen, ohne neu zu übersetzen und zu binden.
- Der volle Funktionsumfang von openUTM in Unix-/Windows-Systemen kann auch bei verteilten Anwendungen getestet werden.
- Wie bei einer „echten“ Produktionsanwendung werden Wartezeiten mit dem Timerprozess überwacht.
- Sie können unter Unix-Systemen auch Ausgabe-Aufträge an Drucker testen.
- Beim Testbetrieb werden alle Prozesse außer den utmwork-Prozessen genau wie bei der Produktionsanwendung gestartet. Deshalb ist beispielsweise der Zugriff vom und zum Netzwerk in vollem Umfang möglich.

Das folgende Bild zeigt schematisch den Aufbau beim Testen im Dialog.



Außerdem ist der Zugriff vom Netz aus möglich.

2.1.1 Testen einer UTM-Anwendung unter Unix-Systemen

In Unix-Systemen können Sie Testhilfen wie *dbx*, *gdb*, *sdb*, COBOL-Testhilfen (Animator) usw. einsetzen. Im Normalfall kann man sich mit dem Debugger an die einzelnen Prozesse der laufenden Anwendung anhängen, so dass die Anwendung ganz normal (mit TEST STARTUP) gestartet werden kann, siehe „[Debuggen einer UTM-Anwendung unter Unix-Systemen](#)“.

Wenn Sie Programme testen wollen, die beim Start der Anwendung automatisch ablaufen, z.B. Start-Exit, MSGTAC, dann müssen Sie anders vorgehen, siehe [Abschnitt „Starten einer UTM-Anwendung unter Unix-Systemen mit Debugger“](#) auf Seite 30.

Voraussetzungen für das Debuggen von Micro Focus COBOL-Programmen mit Animator

Für COBOL-Programme, die mit Micro Focus COBOL-Compiler übersetzt werden und die Sie mit Animator debuggen möchten, sind folgende Vorbereitungen notwendig:

1. Setzen Sie die Compiler-Option *-a*
2. Bereiten Sie die Umgebung für den Start unter Animator vor mit:
 - `export COBSW=+A`
Ermöglicht die dynamische Animation ohne expliziten Aufruf des Animators.
 - `export COBPATH=source-dateiverzeichnis`
Gibt den Pfad an, unter dem der Animator die *.cbl*-Dateien und *.int*-Dateien sucht.

2.1.1.1 Debuggen einer UTM-Anwendung unter Unix-Systemen

Wenn Sie ein Teilprogramm debuggen wollen, dann gehen Sie wie folgt vor:

1. Starten Sie die Anwendung normal im Debug-Modus.
2. Starten Sie einen Debugger und hängen sich an den/die Workprozess(e) an.

Beispiele

- Solaris:
`dbx pid` oder
`dbxtool pid`
- Linux:
`gdb pid` oder
`ddd pid`

pid ist die Prozess-Id eines Workprozesses.

2.1.1.2 Starten einer UTM-Anwendung unter Unix-Systemen mit Debugger

Wenn Sie ein Teilprogramm debuggen wollen, das beim Start der Anwendung automatisch abläuft, z.B. Teilprogramme für einen Start-Exit oder das MSGTAC-Programm für Start-Meldungen, müssen Sie die Anwendung speziell starten:

1. Sie starten den Mainprozess als Hintergrundprozess wie bei einer UTM-Produktiv-anwendung. Sie müssen jedoch zusätzlich den Parameter TEST angeben.

```
utmpfad/ex/utmmain filebase startparam-file TEST &
```

Die Startparameterdatei müssen Sie immer vollqualifiziert angeben, auch dann, wenn der Standardname verwendet wird.

Der Mainprozess *utmmain* erzeugt dann nur den Timerprozess und weitere Betriebsmittel wie z.B. die Pipe zur Kommunikation mit den Workprozessen. Es werden aber keine Workprozesse erzeugt.

2. Der Mainprozess fordert Sie mit der Meldung U244 auf, einen oder mehrere Workprozesse zu starten. Beachten Sie bitte, dass beim Testen im Dialog der zweite Workprozess ein UTM-System-Prozess ist und deshalb ggf. drei Workprozesse gestartet werden müssen, wenn z.B. Programme mit PGWT getestet werden sollen.

Den ersten Workprozess *utmwork* müssen Sie unter der Kontrolle der Testhilfe mit folgenden Argumenten starten:

```
utmwork version appliname filebase startpar pid id mode
```

Die gesamte Anweisung muss **ohne** Zeilenvorschub geschrieben werden, auch wenn sie mehr als eine Zeile beansprucht. Die Argumente haben folgende Bedeutung:

version	Verwendete openUTM-Version, z.B. V06.3A00
appliname	Name der Anwendung wie in MAX APPLINAME generiert.
filebase	Dateiverzeichnis der Anwendung (vollqualifiziert)
startpar	Datei, in der die Startparameter definiert sind (vollqualifiziert)
pid	Prozess-ID des Vorgängerprozesses bei PENDER bzw. Wert 0 bei zusätzlichen Prozessen
id	ID des utmwork-Prozesses, beim Starten des ersten und eines zusätzlichen Workprozess (Mode=Y oder N) geben Sie immer den Wert 0 an. Im Fall von Mode=E ist immer der Wert aus der korrespondierenden U244 Meldung anzugeben.
mode	Kennzeichen für den Startmodus Y: erster Workprozess der Anwendung N: zusätzlicher neuer Workprozess E: nach PENDER nachgestartet

Beispiel für dbx

```
run V06.3A00 sample /home/utmbp /home/utmbp/startparameter 0 0 Y
```

Die Parameter für das Starten der Workprozesse entnehmen Sie aus der dazugehörigen Meldung U244.

3. Workprozesse sollten immer dann nachgestartet werden, wenn der Mainprozess mit der Meldung U244 dazu auffordert.

Zusätzliche Prozesse benötigen als letztes Argument immer den Wert N.

```
utmwork version appliname filebase startpar pid id N
```

Beispiel für dbx

```
run V06.3A00 sample /home/utmbp /home/utmbp/startparameter pid id N
```

Sie können nicht mehr Workprozesse starten als in der Startparameterdatei bei TASKS= angegeben wurden. Die Maximalzahl der Workprozesse wird bei der Generierung in der MAX-Anweisung festgelegt. Versuchen Sie mehr Workprozesse zu starten, führt dies zu Startfehlern.

4. Nach dem Beenden der UTM-Anwendung ist der Mainprozess mit dem Kommando `kill` zu löschen, wenn nicht alle Workprozesse die vorgesehene Prozess-Endebehandlung vollständig durchlaufen haben.

Beispiel

```
kill -9 1234
```

Dabei ist 1234 die Prozess-ID von utmmain.

Eine ausführliche Beschreibung der Arbeitsschritte zum Starten finden Sie im openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“.

Verhalten bei Fehlern

Falls ein Workprozess einen PEND ER-Dump erzeugt, wird dieser Workprozess beendet. Der Mainprozess startet dann keinen neuen Workprozess. Sie müssen den Workprozess dann manuell nachstarten und als letzten Parameter E angeben. Die genauen Parameter können aus der Meldung U244 übernommen werden. Als *pid* geben Sie die Prozess-ID des Vorgänger-Workprozesses an, der sich mit PEND ER beendet hat.

```
utmwork version appliname filebase startpar pid id E
```

Wird ein Workprozess fehlerhaft beendet, sollten Sie vor dem nächsten Start der Anwendung das Tool KDCREM aufrufen, siehe entsprechender Abschnitt im openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“.

Beenden der Anwendung

Die Anwendung beenden Sie wie sonst auch durch die UTM-Administration oder mit dem Tool KDCSHUT. Falls sich die Anwendung nicht normal beendet, sollten Sie das Tool KDCREM aufrufen.

Details siehe entsprechende Abschnitte zur KDCSHUT und KDCREM im openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“.

2.1.2 Testen einer UTM-Anwendung unter Windows-Systemen

In Windows-Systemen verwenden Sie zum Testen den Debugger, der im Microsoft Visual Studio 2010 integriert ist. Im Normalfall kann man sich mit Visual Studio an die einzelnen Prozesse der laufenden Anwendung anhängen, so dass die Anwendung ganz normal (mit TEST STARTUP) gestartet werden kann, siehe „[Debuggen einer UTM-Anwendung unter Windows-Systemen](#)“.

Wenn Sie Programme testen wollen, die beim Start der Anwendung automatisch ablaufen, z.B. Start-Exit, MSGTAC, dann müssen Sie anders vorgehen, siehe „[Starten einer UTM-Anwendung unter Windows-Systemen mit Debugger](#)“.

Die folgende Beschreibung bezieht sich auf die englische Variante des Visual Studio 2010.

2.1.2.1 Debuggen einer UTM-Anwendung unter Windows-Systemen

Wenn Sie ein Teilprogramm debuggen wollen, gehen Sie wie folgt vor:

1. Starten Sie die Anwendung normal im Debug-Modus.
2. Öffnen Sie das Anwendungsprojekt mit dem Microsoft Visual Studio.
3. Klicken Sie in der Menü-Zeile bei *Debug* den Punkt *Attach to Process* an, wählen einen Prozess *utmwork.exe* aus und klicken *Attach*.

Wiederholen Sie dies für jeden work-Prozess Ihrer UTM-Anwendung.

2.1.2.2 Starten einer UTM-Anwendung unter Windows-Systemen mit Debugger

Wenn Sie ein Teilprogramm debuggen wollen, das beim Start der Anwendung automatisch abläuft, z.B. Teilprogramm für einen Start-Exit oder das MSGTAC-Programm für Start-Meldungen, dann müssen Sie die Anwendung speziell starten:

1. Sie starten den Mainprozess wie bei einer UTM-Produktivanwendung. Sie müssen jedoch zusätzlich den Parameter TEST angeben.

Öffnen Sie dazu ein Fenster *Eingabeaufforderung*, wechseln Sie in das Anwendungsverzeichnis und geben Sie Folgendes ein:

```
utmmain . startparam-file TEST 1>utmp-out.txt <nul
```

Dabei muss die PATH-Variable entsprechend gesetzt sein. Die Ausgabe auf *stdout* wird hier in Datei umgelenkt. *null* entspricht */dev/null* in Unix-Systemen.

Der Mainprozess *utmmain* erzeugt dann nur den Timerprozess und weitere Betriebsmittel wie z.B. die Pipe zur Kommunikation mit den Workprozessen. Es werden aber keine Workprozesse erzeugt.

Der Mainprozess fordert Sie mit der Meldung U244 auf, einen Workprozess zu starten. U244 kann z.B. so aussehen:

```
U244 utmmain: Bitte ./utmwork starten mit Argumenten:
V06.3A00 sample01 . ./startp.std 0 0 Y
```

Die zweite Zeile benötigen Sie später für den Debugger. Das Argument 0 bedeutet, dass dieser Prozess neu gestartet wird, d.h. er ersetzt keinen zuvor beendeten Workprozess. Y bedeutet zusätzlich, dass dies der erste Workprozess ist.

Sie dürfen das Fenster jetzt nicht schließen, sonst wird der Mainprozess beendet!

Beachten Sie bitte, dass beim Testen im Dialog der zweite Workprozess ein UTM-System-Prozess ist und deshalb ggf. drei Workprozesse gestartet werden müssen, wenn z.B. Programme mit PGWT getestet werden sollen.

2. Den ersten Workprozess utmwork müssen Sie unter der Kontrolle des Debuggers starten. Dabei gehen Sie wie folgt vor:
 - a) Öffnen Sie das Anwendungsprojekt mit dem Microsoft Visual Studio und wählen Sie den Menüpunkt *Project - utmwork properties* aus.
 - b) Klicken Sie im Navigationsbereich bei *Configuration Properties* den Punkt *Debugging* an, geben im Feld *Working Directory* den *filebase*-Namen an und tragen Sie in das Feld *Command Arguments* die Werte aus der Meldung U244 ein.
Diese Werte haben folgenden Aufbau:


```
utmversion appliname filebase startparameter-file 0 0 Y
```
 - c) Klicken Sie auf OK.
 - d) Starten Sie den ersten Workprozess mit dem Debugger, indem Sie im Menü *Debug* den Punkt *Start Debug* anklicken.
3. Danach können Sie die weiteren Workprozesse starten. Workprozesse sollten immer dann nachgestartet werden, wenn der Mainprozess mit der Meldung U244 dazu auffordert. Zusätzliche Prozesse benötigen als letztes Argument immer den Wert N.

Beachten Sie bitte, dass beim Testen im Dialog der zweite Workprozess ein UTM-System-Prozess ist und deshalb ggf. drei Workprozesse gestartet werden müssen, wenn z.B. Programme mit PGWT getestet werden sollen.

Die Parameter für das Starten der Workprozesse entnehmen Sie aus der dazugehörigen Meldung U244.

```
version appliname filebase startparam-file pid id N
```

Sie können nicht mehr Workprozesse starten als in der Startparameterdatei bei `TASKS=` angegeben wurden. Die Maximalzahl der Workprozesse wird bei der Generierung in der `MAX`-Anweisung festgelegt. Versuchen Sie mehr Workprozesse zu starten, führt dies zu Startfehlern.

4. Wenn nicht alle Workprozesse die vorgesehene Prozess-Endebehandlung vollständig durchlaufen haben, müssen Sie nach dem Beenden der Workprozesse den Mainprozess explizit beenden, indem Sie das zugehörige Fenster schließen oder den Prozess *utmmain.exe* mit dem Task-Manager beenden.

Nach dem Beenden der Anwendung müssen Sie den *utmain*-Prozess beenden, z.B. indem Sie das Fenster schließen.

Eine ausführliche Beschreibung der Arbeitsschritte zum Starten finden Sie im openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“.

Verhalten bei Fehlern

Falls ein Workprozess einen PEND ER-Dump erzeugt, wird dieser Workprozess beendet und Sie werden vom *utmain*-Prozess mit der Meldung U244 aufgefordert, einen neuen Workprozess zu starten.

Dazu wiederholen Sie Schritt 2, ersetzen jedoch folgende Werte:

- die Werte 0 für *pid* und *id* jeweils durch die in der Meldung U244 vorgegebenen Werte
- den Wert Y durch Wert E

2.1.3 Ausgabe von Meldungen beim Start eines Prozesses

Damit Sie Fehler, die beim Start einer UTM-Anwendung oder eines Folgeprozesses auftreten, leichter diagnostizieren können, werden alle K-Meldungen von openUTM, die in der Startphase auftreten, immer auf *stderr* und *stdout* ausgegeben.

2.1.4 Arbeiten am Terminal im Testbetrieb

Es können sich mehrere Terminals bei der Anwendung anmelden. Sie starten den Dialog-Terminalprozess wie im Produktivbetrieb, indem Sie das Programm `utmdtp` aufrufen.

`utmdtp` steht im Dateiverzeichnis `utmpfad/ex` bzw. `utmpfad\ex`. Das Programm wird wie folgt gestartet:

```
utmdtp [_-S[username]][_ -Anwendungsname][_ -D][_ -Ppterm-name]
```

Die Angaben in eckigen Klammern stellen Schalter dar, die Sie angeben können, aber nicht müssen. Zwischen einem Schalter und seinem nachfolgenden Argument darf kein Leerzeichen oder Zwischenraumzeichen stehen!

Die Schalter haben folgende Bedeutung:

`-S[username]`

Mit diesem Schalter steuern Sie, ob für die Berechtigungsprüfung im Dialog explizit eine UTM-Benutzerkennung übergeben wird, oder eine Unix- bzw. Windows-Kennung implizit als UTM-Benutzerkennung übergeben wird.

openUTM führt die Berechtigungsprüfung (Zugangskontrolle) nach dem erfolgreichen Verbindungsaufbau zur UTM-Anwendung durch.

Eine Berechtigungsprüfung erfolgt unabhängig davon, ob Schalter `-S` angegeben wurde, wenn die Anwendung mit Benutzerkennungen (USER-Anweisungen) generiert wurde, aber dem LTERM-Partner bei der Generierung keine Benutzerkennung zugeordnet worden ist (LTERM USER=...).

Starten Sie am Terminal den Dialog-Terminalprozess **mit** Schalter `-S`, so müssen Sie zur Berechtigungsprüfung eine UTM-Benutzerkennung an openUTM übergeben. Die UTM-Benutzerkennung können Sie direkt beim Start des Dialog-Terminalprozesses mit `-Susername` angeben (username = generierte UTM-Benutzerkennung).

Geben Sie nur `-S` an, fragt openUTM die UTM-Benutzerkennung nach dem Verbindungsaufbau im Dialog ab.

Geben Sie am Terminal eine UTM-Benutzerkennung an, für die ein Kennwort oder eine Ausweisüberprüfung generiert ist, dann fragt openUTM die entsprechenden Daten im Dialog ab.

Starten Sie am Terminal den Dialog-Terminalprozess **ohne** Schalter -S, so übergibt der Dialog-Terminalprozess die Unix- bzw. Windows-Kennung implizit als UTM-Benutzerkennung zur Berechtigungsprüfung. Ein Kennwort wird nicht an openUTM übergeben. In der UTM-Anwendung darf für die entsprechende Benutzerkennung ein Kennwort vergeben werden, zu dessen Eingabe - wie im Fall der expliziten Angabe der Benutzerkennung - der Benutzer am Terminal aufgefordert wird.

Verläuft eine Prüfung mit der implizit übergebenen Benutzerkennung negativ, dann folgt ein expliziter Berechtigungsdialog wie bei der Verwendung des Schalters -S.

-*Aanwendungsname*

Mit dem Schalter geben Sie am Terminal die Anwendung an, mit der Sie verbunden werden wollen. Für *anwendungsname* ist der Name der Anwendung anzugeben. Wird „-*Aanwendungsname*“ beim Start des Dialog-Terminalprozesses **nicht** angegeben, dann erfragt der Dialog-Terminalprozess in einem Dialog den Anwendungsnamen.

-D

Mit diesem Schalter bestimmen Sie die Reaktion des Dialog-Terminalprozesses auf die Taste DEL (Unix-Systeme) bzw. die Tastenkombination CTRL + C (Windows-Systemen).

Geben Sie den Schalter -D an, wird DEL bzw. CTRL + C vom Dialog-Terminalprozess ignoriert.

Geben Sie den Schalter -D **nicht** an, führt das Drücken der DEL-Taste bzw. der Tastenkombination CTRL + C zum Abmelden von der Anwendung und zur Beendigung des Dialog-Terminalprozesses.

-P*pterm-name*

Sie melden sich unter dem PTERM-Namen *pterm-name* an die UTM-Anwendung an.

Unix-Systeme: Standardmäßig verwendet openUTM als *pterm-name* den letzten Teil der Ausgabe des tty-Kommandos (den Term hinter dem letzten Schrägstrich).

Das entspricht der Ausgabe des Kommandos `basename `tty``. Bei der KDCDEF-Generierung sollte für alle lokalen Terminals und Pseudoterminals eine zugehörige PTERM-Anweisung (unter diesem *pterm-namen*) angegeben werden.

Es kann in Unix-Systemen vorkommen, dass die Standardzuordnung von *pterm-name* durch openUTM nicht eindeutig ist. Abhängig von der Art der Netze, an die das System angeschlossen ist, können zwei oder mehr Pseudoterminals existieren, die sich in dem letzten Bestandteil des *tty* (hinter dem letzten Schrägstrich) nicht unterscheiden. Nur eins dieser Terminals kann dann mit diesem Pterm-Namen die Verbindung zur Anwendung aufbauen. Die Verbindungsanforderung des zweiten Terminals wird von openUTM zurückgewiesen.

Windows-Systeme: Standardmäßig bildet openUTM *pterm-name* nach dem Muster `ttynnnnn`, wobei *nnnnn* die PID des `utmdtp`-Prozesses als aufbereitete Dezimalzahl ist.

Unter Unix-Systemen können Sie das Programm `utmdtp` auch direkt als Startprogramm in die eigene `.profile` oder in die Datei `/etc/passwd` eintragen. Dann wird der Dialog-Terminalprozess sofort nach erfolgreicher Anmeldung bei Unix-Systemen gestartet.

Unter Windows-Systemen können Sie das Programm `utmdtp` auch in der Autostart-Gruppe eintragen. Das Programm wird dann automatisch gestartet.

2.2 Fehlerdiagnose

In diesem Abschnitt erfahren Sie

- welche Returncodes die Programmschnittstelle liefert
- wie openUTM Fehler durch Meldungen signalisiert
- welche Unterlagen bei Fehlern erstellt werden müssen
- welche Traces Sie zur Diagnose verwenden können

Die Beschreibung und Auswertung eines UTM-Dumps finden Sie im [Kapitel „Der UTM-Dump“ auf Seite 59](#).

2.2.1 Returncodes an der Programmschnittstelle

Nach jedem KDCS-Aufruf (außer dem PEND) gibt openUTM im Rückgabebereich des Kommunikationsbereichs folgende Returncodes und Kennzeichen zurück:

- den KDCS-Returncode,
- den internen Returncode.

KDCS-Returncode im Feld KCRCCC

Beachten Sie bitte:

- Treten mehrere Fehler gleichzeitig auf, so werden die Fehler der jeweils höchsten Kategorie angezeigt. Innerhalb einer Kategorie wird nicht immer der niedrigste Rückkehrcode angegeben (abweichend von DIN 66 265).
- Die genaue Bedeutung der KDCS-Returncodes ist im openUTM-Handbuch „Anwendungen programmieren mit KDCS“ bei jedem KDCS-Aufruf beschrieben. Eine Übersicht aller KCRCCC-Returncodes finden Sie ab [Seite 419](#).

Interner Returncode von openUTM in KCRCDC

Der interne Returncode enthält im Fehlerfall eine genauere Spezifikation des Fehlers als der KDCS-Returncode in KCRCCC. Meist wird dieser Returncode zusammen mit den KDCS-Returncodes 40Z oder 70Z (System- bzw. Generierungsfehler) gesetzt. Die genaue Bedeutung finden Sie ab [Seite 419](#).

Der interne UTM-Returncode ist nicht Bestandteil der genormten Schnittstelle KDCS.

2.2.2 UTM-Meldungen bei Programmfehlern

Programmfehler sind Fehler bei der Programmierung der KDCS-Schnittstelle (siehe KDCS-Returncode). openUTM erzeugt dann bestimmte Meldungen, die standardmäßig nach STDOUT, STDERR und in der Regel auch nach SYSLOG ausgegeben, siehe [Abschnitt „Ziele der UTM-Meldungen“ auf Seite 398](#).

Abnormale Beendigung eines Vorgangs

Bei einer abnormalen Beendigung eines Dialog-Vorgangs sendet openUTM die Meldung K017, bei einer abnormalen Beendigung eines Asynchron-Vorgangs die Meldung K055.

Diese Meldungen enthalten Returncodes, die weiteren Aufschluss über die Fehlerursache geben:

- K017 Vorgang &TCVG durch UTM beendet (&RCCC/&RCDC &RCF2A) – Bitte Eingabe

Die Einträge haben folgende Bedeutung:

&TCVG : TAC, mit dem der Vorgang gestartet wurde
&RCCC : KDCS-Returncode im Feld KCRCCC
&RCDC : interner Returncode im Feld KCRCDC
&RCF2A : (immer 0)

- K055 Asynchron-Vorgang &ATAC1 durch UTM beendet; KCRCCC= &RCCC ; KCRCDC= &RCDC ; Benutzer= &USER ; LTERM= <RM

Die Einträge haben folgende Bedeutung

&ATAC1 : TAC, mit dem der Asynchron-Vorgang gestartet wurde
&RCCC : KDCS-Returncode im Feld KCRCCC
&RCDC : interner Returncode im Feld KCRCDC
&USER : Benutzerkennung, die den Asynchron-Vorgang erzeugt hat.
<RM : LTERM-Partner, von dem Asynchron-Vorgang erzeugt wurde.

Fehler im Input-Exit

- Bei Fehlern im INPUT-Exit schickt openUTM in der Regel die Meldung K098 an das Terminal. Die Erklärung der in K098 enthaltenen Fehlercodes finden Sie auf [Seite 252](#).

2.2.3 Diagnose-Dump bei festgelegten Meldungen/Ereignissen

Beim Eintreten eines bestimmten Ereignisses können Sie das Ziehen eines Diagnose-Dumps, genannt Message-Dump, veranlassen, dessen Kennzeichen abhängig vom Ereignistyp ist.

Ein Message-Dump wird nur von der Task erstellt, in der das Ereignis eintritt. Dabei wird die UTM-Anwendung nicht beendet. Für das Ziehen eines solchen Message-Dumps müssen Sie den Testmodus für die Anwendung einschalten und das Ereignis, bei dem der Message-Dump gezogen werden soll, setzen. Beides kann per Startparameter oder administrativ erfolgen.

Folgende Ereignisse können Sie angeben:

- die Ausgabe einer bestimmten K-Meldung
- das Auftreten eines bestimmten KDCS-Returncodes (CC oder DC) in einem Teilprogrammmlauf
- das Auftreten eines bestimmten SIGN-Status beim Anmelden eines Benutzers

Einschalten des Testmodus

- über Startparameter

```
.UTM TESTMODE = ON
```

- über Administrationskommando

```
KDCDIAG TESTMODE = ON
```

- über die Programmschnittstelle zur Administration (z.B. über WinAdmin oder WebAdmin)

Im Objekt-Typ `KC_DIAG_AND_ACCOUNT` geben Sie an:

```
testmode='Y' (Datenstruktur kc_diag_and_account_par_str)
```

Die Funktion Message-Dump einschalten und zurücksetzen

Sie können die Funktion Message-Dump per Startparameter oder administrativ einschalten und zurücksetzen. Standardmäßig ist die Funktion beim Start der Anwendung ausgeschaltet.

Einschalten der Funktion Message-Dump

- über Startparameter

```
.UTM START DUMP-MESSAGE = (event-type,event)
```

Damit wird ein Message-Dump gezogen, sobald das Ereignis eintritt.

Beim Einschalten über Startparameter können Sie nur ein Event angeben. Die Angabe von Inserts zu einer Meldung ist nicht möglich.

- über Administrationskommando

```
KDCDIAG DUMP-MESSAGE = (event-type,event)
```

Bis zu drei unterschiedliche Ereignisse können Sie in gleicher Weise über die Parameter DUMP-MESSAGE_x (mit x = 1, 2, 3) für das Ziehen des Message-Dumps angeben. Dabei ist DUMP-MESSAGE synonym zu DUMP-MESSAGE1.

Für das Ereignis „Ausgabe einer bestimmten K-Meldung“ können Sie als zusätzliche Bedingung bis zu drei Inserts angeben (Parameter INSERT_x mit x = 1,2,3). Näheres siehe KDCDIAG im openUTM-Handbuch „Anwendungen administrieren“.

- über die Programmschnittstelle zur Administration (z.B. über WinAdmin oder WebAdmin)

Im Objekt-Typ KC_DIAG_AND_ACCOUNT geben Sie in den Feldern *event* und *event_type* und das Ereignis und den Ereignis-Typ an (Datenstruktur *kc_diag_and_account_par_str* mit entsprechenden Unterstrukturen). Zusätzlich können Sie bis zu drei Inserts als Bedingung angeben. An der Programmschnittstelle können Sie pro Aufruf ein Ereignis angeben. Weitere Details siehe Beschreibung von KC_DIAG_AND_ACCOUNT im openUTM-Handbuch „Anwendungen administrieren“.

Mit *event-type* geben Sie den Ereignistyp und mit *event* dazu ein bestimmtes Ereignis an, bei dem der Message-Dump gezogen werden soll. Folgende Ereignisse können Sie angeben:

- Ausgabe einer bestimmten K-Meldung (*event-type* = MSG)
Als *event* geben Sie die UTM-Meldungsnummer *Knnn* an.
Bei jedem Auftreten der Meldungsnummer wird ein Dump erzeugt, solange, bis Sie die Meldungsnummer zurücksetzen. Bei den Meldungsnummern K043, K061, K062 wird nur einmal ein Dump erzeugt, und dann die Meldungsnummer automatisch zurückgesetzt.

- Auftreten eines bestimmten kompatiblen KDCS-Returncodes (*event-type* = RCCC)
Als *event* geben Sie die Nummer des kompatiblen KDCS-Returncodes (KCRCCC), z.B. 14Z an.
Tritt der Returncode bei einem KDCS-Aufruf auf, wird nur einmal ein Dump erzeugt und anschließend der Parameter DUMP-MESSAGE[x] auf *NONE zurückgesetzt.
- Auftreten eines bestimmten inkompatiblen KDCS-Returncodes (*event-type* = RCDC)
Als *event* geben Sie den inkompatiblen KDCS-Returncode (KCRCDC), z.B. KD10 an.
Tritt der Returncode bei einem KDCS-Aufruf auf, wird nur einmal ein Dump erzeugt und anschließend der Parameter DUMP-MESSAGE[x] auf *NONE zurückgesetzt.

Hinweis

Bei allen KDCS-Returncodes $\geq 70Z$ und den zugehörigen inkompatiblen KDCS-Returncodes, bei denen grundsätzlich kein PENDER-Dump geschrieben wird (z.B. 70Z/K316), wird auch kein Message-Dump erzeugt.

- Auftreten eines bestimmten Anmeldestatus (*event-type* = SIGN)
Als *event* geben Sie den SIGNON-Statuscode in der Form *xyy* (z.B. U05) an:
 - *x* entspricht dem Wert in KCRSIGN1, wobei die Werte U, I, A oder R möglich sind
 - *yy* entspricht dem Wert in KCRSIGN2Tritt der Statuscode bei einem SIGN-Aufruf auf, wird nur einmal ein Dump erzeugt und anschließend der Parameter DUMP-MESSAGE[x] auf *NONE zurückgesetzt. Dies geschieht unabhängig davon, ob in der Anwendung ein Anmelde-Vorgang generiert ist oder nicht.

Ausschalten der Funktion Message-Dump

- über Startparameter
.UTM START DUMP-MESSAGE = *NONE
- über Administrationskommando
KDCDIAG DUMP-MESSAGE = *NONE
Bei Ereignissen, die mit den Parametern DUMP-MESSAGE_x eingeschaltet wurden, ist der entsprechende Parameter auf *NONE zu setzen.
- über die Programmschnittstelle zur Administration (z.B. über WinAdmin oder WebAdmin)
Im Objekt-Typ KC_DIAG_AND_ACCOUNT setzen Sie alle Ereignisse zurück, indem Sie für jedes Ereignis *event_type=NONE* setzen.

2.2.4 Fehlerunterlagen erstellen

Zur Fehlerdiagnose sind folgende Angaben nötig:

- Genaue Beschreibung der Fehlersituation
- Angabe der Versionsstände der beteiligten Software
- Genaue Angabe des Rechnertyps

Die Fehlerunterlagen sollten möglichst vollständig vorhanden sein. Als Fehlerunterlagen können dienen:

- UTM-Dumps von allen Workprozessen sowie zugehörige „gcores“ unter Unix-Systemen bzw. „Minidumps“ unter Windows-Systemen. Die „Minidumps“ stehen standardmäßig im DUMP-Verzeichnis und haben die Endung DMP. Diese Dateien sollten als Binärdateien, d.h. in nicht aufbereiteter Form, vorliegen.
- Die SYSLOG-Datei(en) (siehe [Seite 153](#)).
- Die *stdout*- und *stderr*-Protokolle aller UTM-Prozesse.
- Die *stdout*-, *stdin*- und *stderr*-Protokolle der KDCDEF-Generierung und die Startprozedur einschließlich der Startparameter.
- Alle Binderlisten, Übersetzungslister und Übersetzungsprozeduren.
- Bei Fehlern, die in Zusammenhang mit der UTM-Netzanbindung stehen, können zusätzlich folgende Unterlagen erstellt werden:
 - Meldungen der UTM-Netzprozesse auf *stdout* und *stderr*
 - CMX-Traces
 - OSS-Traces
 - dynamischer openUTM-Trace
 - Mapped Hostname-Datei
 - CHECKTNS-Datei, die mit der KDCDEF-Anweisung OPTION erzeugt wurde. Diese Datei befindet sich im Verzeichnis *filebase* und heißt *def_tns*.

Wie Sie die genannten Traces erzeugen, ist auf [Seite 46](#) (CMX-Trace), [Seite 53](#) (OSS-Trace) und [Seite 46](#) (dynamischer openUTM-Trace) beschrieben.

- Bei Fehlern in UTM-Cluster-Anwendungen werden zusätzlich folgende Unterlagen benötigt:
 - Alle Cluster-globale Dateien, Protokolle (und DUMPs) aller Knoten-Anwendungen
 - Die Cluster-Konfigurationsdatei und bei administrativen Problemen auch alle Dateien des Administrations-Journals mit Suffix JKAA, JRN1, JRN2.
 - 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
- Bei Problemen zu Benutzern (z.B. Anmeldeprobleme) auch die Cluster-User-Datei (d.h. die Datei mit dem Suffix UTM-C.USER)
- (nur Unix-Systeme): Die core-Dateien mit den zugehörigen Phasen (`utmwork`) und Shared Objects. Die Shared Objects können Sie mit dem Kommando `ldd utmwork` ermitteln.

Sie sollten versuchen, die Fehler unter Verwendung statischer Bibliotheken zu reproduzieren.

Vorgehen bei Fehlern

- Bei Vorgangsabbruch/Anwendungsabbruch gehen Sie wie folgt vor:
 1. UTM-Dump auswerten mit dem Tool KDCDUMP, siehe [Seite 62](#).
 2. Fehler reproduzieren unter Verwendung geeigneter Debugger wie z.B. `dbx`, `sdb`, `adb`, `xdb`, `gdb`, `debug` unter Unix-Systemen bzw. dem in Microsoft Visual Studio integrierten Debugger unter Windows-Systemen.
 3. Aufrufhierarchie beim `core`-Schreiben mit Hilfe eines Debuggers ermitteln. (Wenn Sie mit der Beispielanwendung arbeiten, können Sie sich diese Aufrufhierarchie unter Unix-Systemen von dem Shell-Skript `p/stack` anzeigen lassen).
- Abbruch mit Signalen

Trat ein PENDING-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.

Der Startparameter `STXIT=OFF` bewirkt, dass das System nach einem fehlerhaften Befehl automatisch den Debugger startet (Windows-Systeme) bzw. sofort (ohne Verzögerung durch `openUTM`) einen `core`-Dump erzeugt und den Prozess ohne UTM-Dump beendet (Unix-Systeme).

Vor dem nächsten Neustart müssen Sie auf jeden Fall `KDCREM` aufrufen, da `openUTM` bei `STXIT=OFF` keine Prozessende-Behandlung durchführt.
- Nach Startfehlern wie z.B. Fehlernummer 32 oder 40 muss vor einem erneuten Start das Tool `KDCREM` aufgerufen werden.

2.2.5 Traces

Für openUTM können Sie außer den Traces im UTM-Dump noch folgende Traces und Tools zur Diagnose heranziehen:

- dynamischer openUTM-Trace (über Umgebungsvariable UTMTRAC)
- Protokollierung von COBOL und C/C++-Teilprogrammaufrufen
- BCAM-Trace in openUTM
- KTA-Trace in Datei
- OSS-Trace für OSI TP
- ADMI-Trace, d.h. Trace der Programmschnittstelle zur Administration (KDCADMI)
- Erzeugen eines `core` bei Anwendungsabbruch (nur Unix-Systeme)
- Tool KDCIPC
- Tool KDCKAA

2.2.5.1 Dynamischer openUTM -Trace über Umgebungsvariable

Durch das Setzen der Umgebungsvariablen UTMTRAC kann ein dynamischer Trace eingeschaltet werden.

Die Umgebungsvariable wird beim Start des Prozesses ausgewertet.

Voreinstellung: Die Protokollierung ist ausgeschaltet.

Syntax

Unix-Systeme:

```
UTMTRAC=prog1#trace1,trace2 [, ...][.file1][:prog2#trace1, ...][: ...]
export UTMTRAC
```

Windows-Systeme:

```
SET UTMTRAC=prog1#trace1,trace2 [, ...][.file1][:prog2#trace1, ...][: ...]
```

Bedeutung der Parameter:

- | | |
|-----------------------|---|
| prog <i>n</i> | entweder <code>all</code> , wenn alle UTM-Programme protokolliert werden sollen, oder UTM-Programm, das protokolliert werden soll. Erlaubt sind:
<code>utmmain, utmwork, utmtimer, utmnet, utmdtp, ...</code>
<code>kdcdef, kdcrem, ..., kdcupd</code> und Lese- und Schreibzugriffe von <code>kdcupd</code> ,
<code>kdcv...</code> , <code>kdcw...</code> |
| trace <i>n</i> | entweder <code>all</code> , wenn alle Trace-Units aktiviert werden sollen, oder Zahl aus der Menge 1, 2, ... UTM_MAX_TRACE_UNITS zur Aktivierung der angegebenen Trace-Unit. |

`file n` optionale Angabe der Ausgabedatei.
`file n` kann `%d` (Unix-Systeme) bzw. `%%d` (Windows-Systeme) enthalten: dieser Platzhalter wird durch die aktuelle `pid` ersetzt.
 Ist `file n` nicht angegeben, wird der Trace nach `stderr` ausgegeben.

Sie können in UTMTRAC eine Liste von Programmen mit unterschiedlichen Angaben für die einzelnen Programme, Trace-Units und Ausgabedateien angeben oder Sie können den Trace für alle Programme und Trace-Units anschalten. Die Definition der Trace-Units finden Sie in der mitausgelieferten Include-Datei `xidyntrc.h`, die sich im Verzeichnis `utmpfad/include` befindet.

Beispiel 1

```
UTMTRAC=all#all (Unix-Systeme)
export UTMTRAC

SET UTMTRAC=all#all (Windows-Systeme)
```

Für alle Prozesse sind alle Trace-Units eingeschaltet. Der Trace wird nach `stderr` ausgegeben.

Beispiel 2

```
UTMTRAC=utmwork#1.wrkp.%d (Unix-Systeme)
export UTMTRAC

SET UTMTRAC=utmwork#1.wrkp.%dd (Windows-Systeme)
```

Für `utmwork` wird die Trace-Unit 1 (KCXPIPE) aktiviert. Der Trace wird nach `wrkp.pid` ausgegeben (`pid`= aktuelle Prozess-ID des `utmwork`-Prozesses).

2.2.5.2 Protokollierung von Teilprogrammaufrufen

Durch das Setzen der Umgebungsvariablen `KDCS_C_DEBUG` wird die Protokollierung von COBOL und C/C++-Teilprogrammaufrufen eingeschaltet.

Die Umgebungsvariable wird bei jedem Start eines Workprozesses ausgewertet. Ist sie gesetzt, wird der Start jedes COBOL und C/C++-Teilprogramms protokolliert. Wenn die C/C++ Programme mit den KDCS-Makros aus dem Headerfile `kcmac.h` programmiert wurden, werden zusätzlich alle KDCS-Aufrufe aus diesen C/C++-Teilprogrammen protokolliert. Die Protokollierung erfolgt standardmäßig nach `stdout`.

Falls die Protokollierung in eine andere Datei erfolgen soll, müssen Sie die Konstante `KDCS_DEBUG_FP` vor dem Inkludieren von `kcmac.h` auf den Wert einer globalen FILE Variablen setzen. Diese Variable versorgen Sie im Startexit.

Voreinstellung: Die Aufrufe werden nicht protokolliert.

2.2.5.3 BCAM-Trace in openUTM

Mit der Funktion BCAM-Trace von openUTM können alle Verbindungs-bezogenen Aktivitäten innerhalb einer UTM-Anwendung protokolliert werden.

Inhalt des BCAM-Trace

Es werden folgende Typen von Trace-Records geschrieben:

- Parameterblock: Die BCAM-Parameterblöcke der Aufrufe REQCON, ACCON, REJCON, DISCON sowie die BCAM-Parameterblöcke der Aufrufe RECLET und SENDLET, die einen Returncode lieferten.
- Announcement: Alle Verbindungs-bezogenen Announcements
- Connection-Letter
- Kommunikation mit TS-Anwendung vom Typ Socket: Parameterblöcke der Funktionen `connection request`, `connection response` und `disconnect request` der Socket-Schnittstelle
- Nachricht: Alle Ein-/Ausgabe-Nachrichten
- CMX-Record: Die Parameter, die im Workprozess bei den CMX-Aufrufen `t_conrq`, `t_conrs`, `t_event`, `t_datain`, `t_datarq` und `t_disrq` verwendet werden.

Jeder Trace-Record enthält folgende Einträge in der angegebenen Reihenfolge:

1. Zeitstempel
2. BCAMAPPL/ACCESS-POINT-Name
3. PTERM/CON/TSEL-Name bei OSI-CON
4. PROCESSOR-Name
5. LTERM/LPAP-Name
6. USER-Name
7. Typ des Trace-Records (s.o.):
 - Announcement
 - Parameterblock
 - Connection-Letter
 - TCP/IP-Record
 - Nachricht (Ein-/Ausgabe-Nachricht)
 - CMX-Record
8. Bis zu 32767 Byte Daten (abhängig vom Recordtyp und vom Wert des Operanden *length* beim Startparameter BTRACE).

Ein-/Ausschalten BCAM-Trace

Der BCAM-Trace kann per Startparameter oder per Administrationskommando ein- und ausgeschaltet werden. Standardmäßig ist die Funktion beim Start der Anwendung ausgeschaltet.

- Ein-/ Ausschalten über Startparameter

$$.UTM \text{ START, BTRACE} = \left\{ \begin{array}{l} \text{ON/OFF} \\ \text{(ON / OFF, length)} \end{array} \right\}$$

Damit wird die Trace-Funktion beim Start der Anwendung eingeschaltet (ON) bzw. bleibt ausgeschaltet (OFF).

Außerdem können Sie die maximale Länge der Daten, die aufgezeichnet werden, angeben.

Minimum: 32

Maximum: 32767

Standardwert: 256

Diese Maximallänge kann nur über Startparameter festgelegt werden.

Wenn Sie den BCAM-Trace für die Funktion UPIC Capture einsetzen (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“), dann wird empfohlen, den Maximalwert zu verwenden.

- Ein-/ Ausschalten über Administrationskommando

```
KDCDIAG BTRACE=ON/OFF[, LTERM=lterm-name / LPAP=lpap-name / USER=user-name]
```

Damit wird die Trace-Funktion im Betrieb der Anwendung ein- bzw. ausgeschaltet.

Bei Angabe eines LTERM-oder eines LPAP-Namens werden nur die zu dieser Verbindung gehörenden Ereignisse aufgezeichnet.

Bei Angabe eines USER-Namens werden nur die zu dieser Benutzerkennung gehörenden Ereignisse aufgezeichnet.

- Ein-/ Ausschalten über Programmschnittstelle zur Administration (z.B. über WinAdmin oder WebAdmin)

Im Objekt-Typ KC_DIAG_AND_ACCOUNT geben Sie an:

```
bcam_trace='Y' bzw. 'N' (Datenstruktur kc_diag_and_account_par_str)
```

Jeder Workprozess erzeugt seine eigene Trace-Datei der Form *filebase/KDCBTRC.pid*.

Nach dem Einschalten des BCAM-Trace wird die Trace-Datei angelegt bzw. mit dem Modus „append“ geöffnet.

Nach dem Ausschalten des BCAM-Trace werden die Trace-Dateien geschlossen und können danach ausgewertet werden. Die Trace-Funktion beendet sich, wenn bei Dateizugriffen auf die Trace-Datei Fehler auftreten.

Auswerten des BCAM-Trace

Zum Aufbereiten einer Trace-Datei steht das Tool KDCBTRC zur Verfügung. Mit dem Tool KDCBTRC können nur Trace-Dateien der gleichen UTM-Version ausgewertet werden.

Vor dem Auswerten können die Trace-Dateien der verschiedenen Workprozesse mit dem Tool *kdcsort* in eine Datei zeitlich sortiert werden:

```
utmpfad/ex/kdcsort btrcace_out btrcace-1 btrcace-2 ... btrcace-n (Unix-Systeme)
```

```
utmpfad\ex\kdcsort btrcace_out btrcace-1 btrcace-2 ... btrcace-n (Windows-Systeme)
```

Details siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“.

Das Tool KDCBTRC wird wie folgt aufgerufen:

```
utmpfad/ex/kdcbtrc_ btrcace-datei_[argument-1 .... argument-n] (Unix-Systeme)
```

```
utmpfad\ex\kdcbtrc_ btrcace-datei_[argument-1 .... argument-n] (Windows-Systeme)
```

Mit den optionalen Argumenten *argument-1*, ..., *argument-n* wird die Aufbereitung gesteuert. Werden keine Argumente angegeben, dann wird die gesamte Trace-Datei aufbereitet. Das Ergebnis der Auswertung wird auf *stdout* geschrieben.

Die möglichen Argumente und ihre Bedeutung werden im nächsten Abschnitt erläutert.

Argumente des Aufbereitungsprogramms

- | | |
|--------|--|
| LT=ltn | Mit diesem Operanden kann ein LTERM-Name oder ein LPAP-Name angegeben werden. Damit werden nur die Trace-Records aufbereitet, die den LTERM/LPAP-Namen enthalten.
Voreinstellung: alle Trace-Records werden aufbereitet. |
| PT=ptn | Mit diesem Operanden kann ein PTERM-Name, CON-Name oder ein TSEL-Name von OSI-CON angegeben werden. Damit werden nur die Trace-Records aufbereitet, die den PTERM/CON/TSEL-Namen (OSI-CON) enthalten. Für den TSEL-Namen ist die Angabe CMX=Y notwendig (siehe unten).
Voreinstellung: alle Trace-Records werden aufbereitet. |
| BC=bcn | Mit diesem Operanden kann ein BCAMAPPL-Name oder ein ACCESS-POINT-Name angegeben werden. Damit werden nur die Trace-Records aufbereitet, die den BCAMAPPL/ACCESS-POINT-Namen enthalten. Für den ACCESS-POINT-Name ist die Angabe CMX=Y notwendig (siehe unten).
Voreinstellung: alle Trace-Records werden aufbereitet. |

PR=prn	Mit diesem Operanden kann ein Prozessname angegeben werden. Damit werden nur die Trace-Records aufbereitet, die den Prozessnamen enthalten. Voreinstellung: alle Trace-Records werden aufbereitet.
US=usr	Mit diesem Operanden kann eine Benutzerkennung (USER) angegeben werden. Damit werden nur die Trace-Records aufbereitet, die den Namen dieser Benutzerkennung enthalten. Voreinstellung: alle Trace-Records werden aufbereitet.
AN=Y/N	Bei Angabe von AN=Y werden die Trace-Records aufbereitet, die Announcements enthalten. Bei AN=N wird die Aufbereitung unterdrückt. Voreinstellung: AN=N
PB=Y/N	Bei Angabe von PB=Y werden die Trace-Records aufbereitet, die BCAM-Parameterblöcke enthalten. Bei PB=N wird die Aufbereitung unterdrückt. Voreinstellung: PB=N
CL=Y/N	Bei Angabe von CL=Y werden die Trace-Records aufbereitet, die Connection-Letters enthalten. Bei CL=N wird die Aufbereitung unterdrückt. Voreinstellung: CL=N
IN=Y/N	Bei Angabe von IN=Y werden die Trace-Records aufbereitet, die Eingabe-Nachrichten enthalten. Bei IN=N wird die Aufbereitung unterdrückt. Voreinstellung: IN=Y
OUT=Y/N	Bei Angabe von OUT=Y werden die Trace-Records aufbereitet, die Ausgabe-Nachrichten enthalten. Bei OUT=N wird die Aufbereitung unterdrückt. Voreinstellung: OUT=Y
CMX=Y/N	Bei Angabe von CMX=Y werden die Trace-Records aufbereitet, die Traces von CMX-Funktionen enthalten. Bei CMX=N wird die Aufbereitung unterdrückt. Voreinstellung: CMX=N
SOCKET=Y/N	Bei Angabe von SOCKET=Y werden die Trace-Records, die Traces für Funktionen zur Kommunikation mit dem Protokoll TCP/IP enthalten, aufbereitet. Bei SOCKET=N wird die Aufbereitung unterdrückt. Voreinstellung: SOCKET=Y

2.2.5.4 KTA-Trace in Datei

In einigen speziellen Fehlerfällen reichen die im UTM-Dump zu findenden Trace-Einträge nicht aus, um die Ursache eines Fehlers zu ermitteln. Für diese Fälle wird der KTA-Trace angeboten, der die KTA-Trace-Einträge bei jedem Überlauf der Trace-Area in eine Datei schreibt.

Der Name dieser Datei setzt sich zusammen aus dem Basisnamen *filebase* und der PID des jeweiligen Workprozesses, d.h. es wird pro Workprozess folgende Datei angelegt:

filebase.KTATRC.pid (pid max. 4-stellig)

KTA-Trace ein- und ausschalten

Mit dem Administrationskommando:

```
KDCDIAG TESTMODE = FILE
```

wird die UTM-Trace-Datei angelegt und geöffnet. Die Trace-Einträge werden jedoch erst geschrieben, wenn die KTA-Trace-Area voll ist oder dieser Workprozess normal beendet wird.

Ausgeschaltet wird diese Funktion mit dem Administrationskommando

```
KDCDIAG TESTMODE = OFF
```

Dabei wird der Inhalt der KTA-Trace-Area in die Datei geschrieben und diese dann geschlossen.

Beim Start der Anwendung ist die Funktion ausgeschaltet. Falls die Funktion bei Ende der Anwendung eingeschaltet war, wird noch der Rest der Traces in die Datei geschrieben.

KTA-Trace auswerten

Unix-Systeme: Mit Hilfe des Shell-Kommandos `hd` können Sie sich die Trace-Datei ansehen bzw. sie in eine ausdrückbare Form bringen.

Windows-Systeme: Mit einem für Binärdateien geeigneten Programm können Sie die Trace-Datei auswerten.

2.2.5.5 OSS-Trace

Der OSS-Trace besteht aus mehreren Typen von Trace-Records. Die Protokollierung kann entweder für jeden Typ einzeln oder für alle Typen zusammen eingeschaltet werden.

filebase.OSST.nr.pid

Dabei ist *filebase* der Basisname der KDCFILE in der MAX-Anweisung, *nr* die laufende Nummer des Trace (0 - 9) und *pid* die Nummer des Workprozesses. openUTM überwacht die Größe der OSS-Trace-Datei und schaltet bei Bedarf auf die nächste Datei. Die Nummern 0 bis 9 werden zyklisch vergeben und nach dem Schließen der Trace-Datei 9 wird die Trace-Datei 0 überschrieben.

Inhalt des OSS-Trace

Der OSS-Trace besteht aus mehreren Typen von Trace-Records. Jeder Typ kann entweder einzeln oder es können alle zusammen eingeschaltet werden.

Es gibt folgende Typen von Trace-Records:

SPI	Das XAP-TP System Programming Interface wird protokolliert.
INT	Der interne Ablauf im XAP-TP-Baustein wird protokolliert.
OSS	Die OSS-Aufrufe werden protokolliert.
SERV	Die OSS-internen Trace-Records vom Typ O_TR_SERV werden protokolliert.
PROT	Die OSS-internen Trace-Records vom Typ O_TR_PROT werden protokolliert.

OSS-Trace ein- und ausschalten

Dieser Trace kann entweder über Startparameter, d.h. beim Anwendungsstart, oder durch ein Administrationskommando ein- und ausgeschaltet werden.

- Ein-/Ausschalten über Startparameter:

```
[.UTM]  START  [ ,OTRACE=ON ]
          [ ,OTRACE=( typ1 [, typ2 ], ... )
          [ ,OTRACE=OFF ]
```

Bei OTRACE=ON ist der OSS-Trace nach dem Anwendungsstart eingeschaltet und es werden alle Typen von Trace-Records protokolliert.

Bei OTRACE=(typ1 [, typ2], ...) ist der OSS-Trace nach dem Anwendungsstart für die angegebenen Typen eingeschaltet (die Reihenfolge der Typangaben ist beliebig).

Bei OTRACE=OFF (Standard) ist der OSS-Trace nach dem Start ausgeschaltet.

- Ein-/Ausschalten per Administrationskommando:

```
KDCDIAG OT[RACE]=ON | ( typ1 [, typ2 ], ... ) | OFF
```

OTRACE = ON schaltet den OSS-Trace für alle Typen von Trace-Records ein,
OTRACE=(typ1 [, typ2], ...) schaltet den OSS-Trace für die angegebenen Typen ein,
OTRACE=OFF schaltet den OSS-Trace wieder aus.

An der OSS-Schnittstelle werden die Parameter beim Einschalten des Trace mit folgenden Werten belegt:

```
o_trmode      =  O_TR_NEW  
o_trsel       =  O_TR_USER + O_TR_PROT  
o_traopt      =  0  
o_mludata     =  32767  
o_mldt        =  0  
o_mltd        =  0
```

Diese Parameter sind ausführlich im Handbuch „OSS (UNIX)“ beschrieben.

OSS-Trace auswerten

Zum Auswerten der erzeugten OSS-Trace-Dateien wird mit openUTM das Programm STEP ausgeliefert. Das Programm steht unter *utmpfad/oss* zur Verfügung. Die Handhabung dieses Auswerteprogramms ist ausführlich im Handbuch „OSS (UNIX)“ beschrieben.

2.2.5.6 ADMI-Trace

Der ADMI-Trace protokolliert alle Aufrufe der Programmschnittstelle KDCADMI.

Vor dem Aufruf werden folgende Daten geschrieben:

- Inhalt des Datenbereichs

Nach dem Aufruf werden folgende Daten geschrieben:

- Adressen von Parameterbereich, Identifikationsbereich, Selektionsbereich und Datenbereich
- Inhalt der Felder des Parameterbereichs, z.B. Operationscode (opcode), Objekttyp, Länge des Datenbereichs, Returncode
- Inhalt des Datenbereichs

Der ADMI-Trace kann über den Startparameter ADMI-TRACE eingeschaltet und über WinAdmin, WebAdmin oder die Programmschnittstelle zur Administration (KDCADMI) ein- oder ausgeschaltet werden.

Der Trace wird standardmäßig in folgende Datei im Dateiverzeichnis der Anwendung (*filebase*) geschrieben:

```
KDC.TR.C.ADMI.appliname.hostname.pid
```

Dabei ist *appliname* der Name der UTM-Anwendung (MAX APPLINAME), *hostname* der Name des Rechners, auf dem die Anwendung läuft, und *pid* die Nummer des Workprozesses.

2.2.5.7 Erzeugen eines core bei Anwendungsabbruch

Durch das Setzen der Umgebungsvariablen UTM_ABORT_WITH_EXCEPTION wird unter Unix-Systemen beim Anwendungsabbruch ein `core` erzeugt.

Unter Windows-Systemen wird der Debugger aktiviert.

Es wird empfohlen, diese Umgebungsvariable nur dann zu setzen, wenn der Startparameter STXIT den Wert OFF hat.

Voreinstellung: Die Umgebungsvariable ist nicht gesetzt.



VORSICHT!

Wenn Sie STXIT=OFF angeben, werden im Fehlerfall zunächst keine Transaktionen zurückgesetzt. Sperren bleiben bis zum nächsten Warmstart der Anwendung bestehen, dann wird die Transaktion zurückgesetzt.

2.2.5.8 Unterdrückung von gcore-Speicherabzügen

Durch das Setzen der Umgebungsvariablen UTM_CORE_DUMP können Sie das Erzeugen von Core-Dumps (Unix-Systeme) bzw. Mini-Dumps (Windows-Systeme) in Prozessen einer UTM-Anwendung unterdrücken.

Enthält die gesetzte Umgebungsvariable den Wert „NO“, wird im Workprozess zu einem UTM-Dump bzw. in den externen Prozessen bei abnormaler Prozessbeendigung kein Core-Dump/Mini-Dump erzeugt. Falls Sie die Umgebungsvariable nicht setzen, oder diese nicht den Wert „NO“ enthält, wird in den oben beschriebenen Situationen ein Core-Dump/Mini-Dump erzeugt.

2.2.5.9 Tool KDCIPC

Bei Problemen mit der openUTM-internen Interprozesskommunikation kann während des Anwendungslaufs ein Auszug des IPC Shared Memories aufbereitet und auf *stdout* ausgegeben werden. Dazu ist kein zusätzliches Aufbereitungsprogramm notwendig.

Zusätzlich besteht die Möglichkeit, während des Anwendungslaufs den IPC-Trace ein- oder auszuschalten, und zwar unabhängig vom Wert des Startparameters TESTMODE=.

Das Tool schreibt alle Informationen auf *stdout*.

KDCIPC aufrufen

Das Tool wird wie folgt aufgerufen:

utmpfad/ex/kdcipc_filebase_[T] [D] [,tron/troff] (Unix-Systeme)

utmpfad\ex\kdcipc_filebase_[T] [D] [,tron/troff] (Windows-Systeme)

filebase Basisname der KDCFILE in der MAX-Anweisung.

T Mit diesem Operanden wird erreicht, dass der Puffer der IPC-Trace Area in zeitlicher Reihenfolge auf *stdout* ausgegeben wird. D.h. der letzte Record in der Liste ist auch der zuletzt erzeugte.

Voreinstellung: Keine Ausgabe der IPC-Trace Area auf *stdout*.

D Mit diesem Operanden wird erreicht, dass das gesamte IPC Shared Memory aufbereitet auf *stdout* ausgegeben wird.

Voreinstellung: Ohne Ausgabe des gesamten IPC Shared Memory.

tron/troff Mit *tron* wird der IPC-Trace-Modus eingeschaltet, mit *troff* wird er ausgeschaltet.

Voreinstellung: Der IPC-Trace-Modus ist abhängig vom Startparameter TESTMODE ein- bzw. ausgeschaltet.

2.2.5.10 Tool KDCKAA

Zur Diagnosehilfe können während des Anwendungslaufs Informationen aus dem KAA aufbereitet auf *stdout* ausgegeben werden. Dazu ist kein zusätzliches Aufbereitungsprogramm notwendig.

KDCKAA aufrufen

Das Tool wird wie folgt aufgerufen:

utmpfad/ex/kdckaa_filebase (Unix-Systeme)

utmpfad\ex\kdckaa_filebase (Windows-Systeme)

Dabei ist *filebase* der Basisname der KDCFILE in der MAX-Anweisung.

3 Der UTM-Dump

Bei schwerwiegenden Fehlern erzeugt openUTM einen Speicherabzug (UTM-Dump) von allen relevanten Daten. Ein UTM-Dump kann folgende Ursachen haben:

- ein Teilprogramm hat einen PEND ER-Aufruf abgesetzt
- im Anwendungsprogramm trat ein KDCS-Returncode KCRCCC $\geq 70Z$ auf wegen eines schweren Fehlers bei einem KDCS-Aufruf oder bei einem Fehler in Zusammenarbeit mit einer Datenbank.
- ein Diagnosedump wurde angefordert (z.B. KDCDIAG)
- ein Fehler im UTM-Systemcode oder in anderen Softwarekomponenten verursachte einen Abbruch des Anwendungslaufs.

Im letzten Fall enthält der UTM-Dump Daten von allen Workprozessen der Anwendung; in den anderen Fällen nur die Daten des betroffenen Workprozesses. Grundsätzlich werden alle UTM-Dumps komprimiert geschrieben.

Auf Ihrem Rechner können Sie sich unter Unix-Systemen mit dem Kommando `what <datei> | grep UNAME` die Betriebssystemversion anzeigen lassen.

Die Tabellendarstellung des UTM-Dumps

Die Länge der einzelnen Zeilen eines UTM-Dumps (80 oder 132 Zeichen) ist abhängig vom EDIT-Operanden des Dienstprogramms KDCDUMP (siehe auch [Seite 62](#)). Daher müssen Sie beim Ausdrucken im lpr-Kommando eventuell den Schalter `-pb3` angeben. Mit Ausnahme der PROGRAM-TABLE hat jede Zeile des aufbereiteten Dumps die Form (für 32Bit-Plattformen):

I	AAAAAAAA	XXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXX	CCCCCCC	CCCCCCC
---	----------	--------------------	--------------------	---------	---------

- I Index oder Laufnummer, 4 Zeichen. Nicht bei jeder Zeile.
- A Adresse, 12 Zeichen.
- X Sedezimaldarstellung der eingetragenen Daten, 64 Zeichen (entspricht 32 Bytes); entfällt in der PROGRAM-TABLE.
- C Klartext der eingetragenen Daten, sofern sie aus abdruckbaren Zeichen bestehen; 32 Zeichen (Ausnahme: PROGRAM-TABLE).

3.1 Die Dateien des UTM-Dumps

Die Dump-Dateien stehen in dem Dateiverzeichnis *filebase/DUMP* bzw. *filebase\dump*. Dabei ist *filebase* der Basisname der KDCFILE in der MAX-Anweisung.

Wird der Dump durch einen Fehler im UTM-Systemcode verursacht (System-Dump), wird eine Dateigenerationsgruppe (FGG) eingerichtet (siehe unten).

Ansonsten hat der Name der Dump-Datei die Form

RRRRRR.PPPP.NN

Dabei bedeuten

RRRRRR Kennzeichen für die Ursache des Speicherabzugs (6 Zeichen).

PPPP die letzten vier Zahlen der Prozessnummer (PID), unter der der Dump erzeugt wurde.

NN Dumpnummer

Bei einem System-Dump wird eine Dateigenerationsgruppe (FGG) im Dateiverzeichnis *RRRRRR.PPPP.NN* eingerichtet. *PPPP* sind die letzten vier Ziffern der Prozessnummer des Prozesses, der den Systemfehler entdeckt hat. Laufen in der Anwendung mehrere Workprozesse, dann wird je Workprozess eine Dump-Datei eingerichtet. Die FGG hat die Form:

/INFO (Verwaltungsdatei der FGG)

/0001 (erste Dump-Datei)

/0002 (zweite Dump-Datei usw., vergleiche USLOG-FGG)

Der UTM-Dump wird unterdrückt, wenn die UTM-Anwendung mit *TESTMODE=OFF* gestartet wurde und ein *PEND ER* mit einem der nachfolgend aufgeführten *KCRCDC*-Codes auftritt.

FH01, K301, K302, K345, K601, K602, K603, K608,
 KM01, KM02, KM03, KM04, KM05, KM07, KM08,
 KR01, KR02,
 KT01, KT02, KT04,
 KU14

In diesen Fällen unterbleibt auch das Beenden und Neustarten des Workprozesses, d.h. der Workprozess arbeitet weiter für die UTM-Anwendung. Ausnahme: Für den aktuellen TAC sind PGWT-Aufrufe erlaubt und der Aufruf, bei dem der Fehler auftrat, war kein *PEND*-Aufruf.

Beim KCRCDC-Code K316 wird unabhängig vom Test-Modus kein UTM-Dump geschrieben. Wenn für den aktuellen TAC PGWT-Aufrufe erlaubt sind und wenn der Aufruf, bei dem der Fehler auftrat, war kein PEND-Aufruf war, dann wird das Programm nachgeladen.

Reduzierung der Dump-Information über Startparameter DUMP-CONTENT

Über den Startparameter DUMP-CONTENT können Sie einstellen, ob openUTM die Dump-Information reduzieren soll oder nicht. Reduzieren heißt, dass Prozess-übergreifende Speicherbereiche (Shared Memories) nur im Dump des Workprozesses enthalten sind, der den Anwendungsabbruch verursacht hat. Durch die Reduzierung der Dump-Information benötigen die Diagnoseunterlagen bei einem Anwendungsabbruch erheblich weniger Platz. Die Kosten für Datenträger und File Transfer beim Weiterleiten der Diagnoseunterlagen sind geringer. Die Reduzierung der Dump-Information ist deshalb als Standard eingestellt. Sie kann durch den Startparameter DUMP-CONTENT bei Bedarf ausgeschaltet und wieder eingeschaltet werden.

Syntax des Startparameters:

```
.UTM START DUMP-CONTENT={ STANDARD | EXTENDED }
```

- STANDARD** Wenn UTM eine Dump-Dateigeneration erzeugt, dann sind Prozess-übergreifende Speicherbereiche nur im Dump des ersten Prozesses (Verursacher) enthalten. Für die Diagnose ist das normalerweise ausreichend. STANDARD ist voreingestellt.
- EXTENDED** Die Prozess-übergreifenden Speicherbereiche sind in allen Dumps einer Dump-Dateigeneration enthalten. Diesen Wert sollten Sie nur auf besondere Anforderung des zuständigen Services einstellen.

3.2 Das Tool KDCDUMP

Sie müssen zur Dump-Aufbereitung das Programm KDCDUMP aufrufen. KDCDUMP bietet dazu zwei Möglichkeiten:

- Auswertung im Dialog:
KDCDUMP lädt einen Dump oder einen Teil davon in den Speicher, so dass Sie ihn am Terminal im Dialog bearbeiten können. Der Dump wird nach *stderr* ausgegeben. Im Dialog können Sie z.B. in einem umfangreichen Dump gezielt bestimmte Tabellen aufbereiten und durchsuchen.
Mit der Umgebungsvariablen EDITOR können Sie festlegen, welchen Editor Sie beim Editieren verwenden wollen (KDCDUMP-Kommando EDT).
Voreinstellung unter Unix-Systemen: vi
Voreinstellung unter Windows-Systemen: WORDPAD.
- Aufbereitung ganzer Dump-Dateien in Listenform:
KDCDUMP erzeugt komplette aufbereitete Dump-Dateien oder Dateigenerationsgruppen (FGGs), die Sie z.B. mit einem Editor anschauen können.

Die Aufbereitung des Dumps steuern Sie über Anweisungen, die Sie nach dem Start von KDCDUMP eingeben.

KDCDUMP wird mit der Steueranweisung END beendet.

Wenn KDCDUMP während der Auswertung auf Fehler läuft, sollten Sie die Aufbereitung ohne Summary-Information versuchen (INFO=DUMP).

Wenn KDCDUMP und die vorliegende Dump-Datei nicht zur selben Version von openUTM gehören, wird der Dump nicht aufbereitet. KDCDUMP gibt dabei die Meldung K719 aus, aus der man erkennen kann, zu welcher Version KDCDUMP und der UTM-Dump jeweils gehören.

Beachten Sie bitte: Wenn Sie Fehlerunterlagen zu Diagnosezwecken zusammenstellen, benötigen Sie die Speicherauszüge aller Workprozesse einer Anwendung.

3.2.1 KDCDUMP starten

KDCDUMP wird aufgerufen durch:

utmpfad/ex/kcdump bzw. *utmpfad*\ex\kcdump

KDCDUMP liest die unten aufgelisteten Anweisungen von *stdin* und gibt Meldungen und Ausgaben nach *stderr* aus. Eine Liste der KDCDUMP-Meldungen finden Sie in [Abschnitt „Meldungen des UTM-Tools KDCDUMP“ auf Seite 352](#).

Das Programm schreibt die aufbereiteten Dumpdateien in das aktuelle Dateiverzeichnis und speichert sie ab unter dem Namen, den Sie mit dem Operanden OUTFILE im Kommando FGG bzw. FILE festgelegt haben. Haben Sie OUTFILE nicht angegeben, speichert KDCDUMP die Ausgabedatei relativ zum aktuellen Dateiverzeichnis unter dem Namen

`dmp1st/anwendungsname/RRRRRR/PPPP.NN` (Bedeutung der Buchstaben siehe [Seite 60](#)).

3.2.2 KDCDUMP-Anweisungen

Die folgende Tabelle zeigt eine Übersicht über alle Anweisungen von KDCDUMP sowie deren Bedeutung:

Anweisung	Bedeutung
!	KDCDUMP unterbrechen und Systemkommando ausführen
!!	Zuletzt ausgeführtes Systemkommando nochmals ausführen
AFIND	Adresse im UTM-Dump suchen
Blätter-Anweisungen	Cursor im Arbeitsbereich positionieren
DUMP	Gesamten UTM-Dump oder Teil davon in den Speicher einlesen
EDT	Editor aufrufen
END	KDCDUMP beenden
FGG	Alle Dateien einer FGG (Dateigenerationsgruppe) aufbereiten
FILE	Einzelne Dump-Datei aufbereiten
FIND	Tabelleneintrag suchen und anzeigen
HELP	Hilfe zu KDCDUMP-Anweisungen ausgeben
LIST	Tabellenausschnitt aufbereiten
SFIND	Suchen nach einem Muster im Dump
SH SYS	Startet aus dem KDCDUMP heraus eine Bourne-Shell unter Unix-Systemen bzw. öffnet ein Eingabeaufforderungs-Fenster unter Windows-Systemen (<code>cmd.exe</code>).
SYSLST	Protokollierung einschalten oder ausschalten
TABLE	Tabelle anzeigen

Eingabe von KDCDUMP-Anweisungen

KDCDUMP-Anweisungen werden von *stdin* gelesen. Eine Anweisung darf maximal aus 256 Zeichen bestehen, längere Eingaben führen zu der Fehlermeldung K759.

Sofern Anweisungsnamen abgekürzt werden können, ist die Abkürzung durch Fettdruck gekennzeichnet. Beispielsweise bedeutet **AFIND**, dass Sie den Anweisungsnamen durch AF abkürzen können.

Bei einigen Operanden muss eines der drei Eingabeformate „C-String“, „X-String“ oder „Dezimal“ eingehalten werden:

Bezeichnung	Format der Eingabe / Beispiel
C-String	[C]'Dies ist ein C-String'
X-String (sedezimal)	X'AAAF' oder X'aaaf' oder X'AaAf'
Dezimal	12345

Werden Indizes oder Displacements/Offsets angegeben, dann werden diese immer positiv interpretiert.

Hinweis für openUTM unter Unix- und Windows-Systemen auf Intel-Prozessoren

Sedezimale Eingaben werden immer arithmetisch interpretiert, d.h. Indizes oder Distanzen, die direkt aus der sedezimalen Aufbereitung abgelesen werden, müssen für „little endian“-Maschinen (z.B. Intel) Byte für Byte in umgekehrter Reihenfolge, d.h. arithmetisch eingegeben werden. Generell ist es nicht erforderlich, führende Nullen einzugeben.

Beispiele für little endian

1. Das 4-Byte Displacement X'00010203' wird im sedezimalen Teil als X'03020100' aufbereitet und muss als X'00010203' eingegeben werden.
2. Der 2-Byte Index X'FEAF' wird im sedezimalen Teil als X'AFFE' aufbereitet und muss als X'FEAF' eingegeben werden.

Im Folgenden steht „Ausgabe auf Terminal“ für die Ausgabe auf *stderr*.

Die folgenden Abschnitte beschreiben die KDCDUMP-Anweisungen in alphabetischer Reihenfolge.

! Systemkommando eingeben

Mit dieser Anweisung kann während einer KDCDUMP-Sitzung ein Unix- bzw. Windows-Kommando ausgeführt werden.

```
!_ command
```

command Unix-Kommando bzw. Windows-Kommando.

!! Zuletzt ausgeführtes Systemkommando wiederholen

Mit dieser Anweisung kann während einer KDCDUMP-Sitzung das zuletzt (mit ! oder !!) ausgeführte Unix- bzw. Windows- Kommando nochmals ausgeführt werden.

!!

AFIND Adresse im Dump suchen

Mit dieser Anweisung kann eine Speicheradresse im UTM-Dump gesucht werden, die während des Speicherabzuges eine gültige Adresse für einen UTM-Tabellenbereich war.

Als Ausgabe auf dem Terminal wird der zugehörige Tabelleneintrag entweder ab Anfang des Eintrages oder ab der gesuchten Stelle angezeigt. Sollte die angegebene Adresse keine Speicheradresse darstellen oder aber sich nicht genau einem Tabelleneintrag zuordnen lassen, so wird eine entsprechende Meldung ausgegeben (K712 respektive K713).

AFIND_ address [,FORMAT]=BE|LE]

address	Gesuchte Adresse im Speicher. Die Adresse muss dezimal angegeben werden. Adressen in der dynamischen Tabelle XAP-DYNN-AREA und in der Tabellengruppe XAP-LOC-BUFF werden nicht gefunden.
FORMAT=	Gibt bei "little endian" Maschinen an, in welchem Format KDCDUMP die unter <i>address</i> angegebene Adresse erwartet ("big endian" oder "little endian"). Bei "big endian" Maschinen wird dieser Parameter ignoriert.
BE	KDCDUMP erwartet die Adresse im "big endian" Format , Standardwert.
LE	KDCDUMP erwartet die Adresse im "little endian" Format.

Blätteranweisungen für Dialog-Aufbereitung

Falls die gewünschte Information nicht auf einem Bildschirm darstellbar ist, kann mit folgenden Anweisungen geblättert werden:

Anweisung	Bedeutung
+ _	Vorwärtsblättern um eine Bildschirmseite (23 Zeilen)
nur Return (leere Eingabe)	

Anweisung	Bedeutung
+n	<ul style="list-style-type: none"> – Bei Tabellenelementen, die nicht komplett auf einer Bildschirmseite darstellbar sind, wird n zur Anfangsadresse der momentan dargestellten Information addiert. Das Ergebnis dieser Addition ist die Adresse, ab der die nächste Information angezeigt wird. – Bei Tabellen, von denen ein Element vollständig auf einer Bildschirmseite darstellbar ist, wird n zum Index des ersten momentan angezeigten Tabellenelementes addiert. Das Ergebnis dieser Addition ist der Index des Elementes, mit dem die Fortsetzung der Tabellenanzeige beginnt. Bei Überschreiten des Tabellenendes wird das letzte Tabellenelement angezeigt, bei Trace-Tabellen wird automatisch umbrochen. n kann dezimal oder sedezimal angegeben werden. <p>n wird für die Anweisung HELP TABLE-NAMES ignoriert. Es wird normal weitergeblättert.</p>
++	Es wird das Ende der Tabelle oder des Tabelleneintrages angezeigt. Ausnahmen bilden die Trace-Tabellen, bei denen zum zeitlichen Ende geblättert wird. Bei der Anweisung HELP TABLE-NAMES wird eine normale „+“-Anweisung durchgeführt.
- -n --	Für das Rückwärtsblättern gilt sinngemäß - mit umgekehrtem Vorzeichen - das gleiche wie beim Vorwärtsblättern. Es wird höchstens bis zum Tabellenanfang zurückgeblättert.
<	Nur bei der Tabelle TRCA: Mit „<“ kann bis zum Anfang der Bearbeitung des aktuellen (bzw. vorherigen) ANNOUNCEMENTS zurückgeblättert werden. Bei SFIND-Kommando wird mit „<“ zum vorherigen Treffer positioniert, auch wenn sich der aktuelle Treffer in einer TRCA-Tabelle befindet.
<<	Bei SFIND-Kommando kann mit „<<“ zum Anfang der Trefferliste geblättert werden.
>	Nur bei der Tabelle TRCA: Mit „>“ kann bis zum Anfang der Bearbeitung des nächsten ANNOUNCEMENTS vorwärts geblättert werden. Bei SFIND-Kommando wird mit „>“ zum nächsten Treffer positioniert, auch wenn sich der aktuelle Treffer in einer TRCA-Tabelle befindet.
>>	Bei SFIND-Kommando kann mit „>>“ zum Ende der Trefferliste geblättert werden.
X	Die Tabellenanzeige wird abgebrochen. Alternativ dazu kann z.B. auch das Kommando END eingegeben werden.

Zur Unterstützung wird auf dem Bildschirm angezeigt, welche Anweisungen jeweils erlaubt sind, z.B.:

„+/-/X“ für alle Tabellen außer TRCA
 „+</>/-/X“ für die Tabelle TRCA

DUMP UTM-Dump in den Speicher einlesen

Mit dieser Anweisung können Sie eine UTM-Dump-Datei (keine Dump-Dateien eines Dateiverzeichnisses) wahlweise vollständig oder nur einen Teil davon in den Speicher einlesen. Diese Anweisung müssen Sie absetzen, bevor Sie einen UTM-Dump interaktiv am Terminal bearbeiten können (z.B. mit den Anweisungen TABLE, LIST).

Besonders bei großen UTM-Dumps, bei denen der zur Verfügung stehende Arbeitsspeicher nicht ausreicht, um den gesamten dekomprimierten Dump aufzunehmen, ist es sinnvoll, nacheinander (mit mehreren DUMP-Kommandos) einzelne Dump-Bereiche in den Arbeitsspeicher einzulesen und zu bearbeiten.

Wurde der UTM-Dump nicht vollständig geschrieben, so kann jeweils ein vollständig abgezogener Dump-Bereich eingelesen und ausgewertet werden.

Mit dem Operanden DOMAIN geben Sie an, welchen Bereich des UTM-Dumps Sie bearbeiten wollen. Um weitere Bereiche des UTM-Dumps bearbeiten zu können, müssen Sie dann weitere DUMP-Anweisungen absetzen. Geben Sie hierbei statt des Dateinamens FILE=*SAME an, liest openUTM den neuen Bereich aus der bereits dekomprimierten UTM-Dump-Datei (Suffix .T). Das Einlesen geht dann viel schneller.

Bei jedem DUMP-Kommando wird der zuvor in den Arbeitsspeicher eingelesene Bereich entfernt und der neue Bereich eingelesen..

```
DUMP_      { FILE = { filename | *SAME }
           [ ,DOMAIN = { ALL | KAA | KTA | CACHE | MPGP | SLOT |
                       XAPTP | ROOT | STACK | USERFILE |
                       JOURNALFILE-1 | JOURNALFILE-2 |
                       BUFSEGMENTS | GSSBFILE | LOCKFILE | CFGFILE |
                       ULSFILE } ]
```

FILE=

filename Name der UTM-Dump-Datei

*SAME Geben Sie statt des Datei- bzw. Linknamens FILE=*SAME an, wenn Sie einen neuen Bereich des zuvor mit DUMP dekomprimierten UTM-Dumps in den Arbeitsspeicher einlesen wollen (siehe Operand DOMAIN).

DOMAIN= Dieser Operanden legt fest, welcher Bereich des UTM-Dumps in den Arbeitsspeicher eingelesen werden soll.

ALL Der gesamten UTM-Dump wird in den Speicher eingelesen.
Standardbelegung

KA A **KDC Application Area** (anwendungsglobale Systempeicher) in den Speicher einlesen.

KTA	KDC Task Area (Task-spezifische Systemspeicher) in den Speicher einlesen.
CACHE	UTM-Cache in den Speicher einlesen.
MPGP	Memory Pagepool in den Speicher einlesen.
SLOT	Slots (dynamisch angelegte Tabellen) in den Speicher einlesen.
ROOT	KDCROOT -Bereiche mit den DIAGAREAS in den Speicher einlesen.
XAPTP	Bereiche des XAPTP-Bausteins in den Speicher einlesen.
STACK	STACK-Bereich in den Speicher einlesen.
USERFILE	Cluster-User-Datei in den Speicher einlesen.
JOURNALFILE-1, JOURNALFILE-2	Datei(en) des Cluster-Administrations-Journals in den Speicher einlesen.
BUFSEGMENTS	Segmente der Knoten-lokalen Pufferverwaltung in den Speicher einlesen.
GSSBFILE	Cluster-GSSB-Datei in den Speicher einlesen.
LOCKFILE	Cluster-Lock-Datei in den Speicher einlesen.
CFGFILE	Cluster-Konfigurations-Datei in den Speicher einlesen.
ULSFILE	Cluster-ULS-Datei in den Speicher einlesen.

EDT Editor aufrufen

Mit dieser Anweisung kann ein Editor aufgerufen werden. Falls die Umgebungsvariable EDITOR mit dem Namen eines Editor versorgt ist, versucht KDCDUMP, diesen Editor aufzurufen. Ist EDITOR nicht gesetzt, wird unter Unix-Systemen der „vi“ aufgerufen, unter Windows-Systemen WORDPAD.

```
EDT_ [ filename ]
```

filename Name der Datei, die eingelesen werden soll.

END KDCDUMP beenden

Mit dieser Anweisung wird KDCDUMP ordnungsgemäß beendet.

END

FGG Alle Dateien einer FGG aufbereiten

Mit der Anweisung FGG können alle Dateien einer FGG gemeinsam in einer Ausgabedatei aufbereitet werden. Der Name der Ausgabedatei wird mit dem Operanden OUTFILE festgelegt.

Im Kommandobetrieb können beliebig viele FGG-Kommandos angegeben werden.

```
FGG_          fgg-name
              [,EDIT = { PRINTER | TERMINAL } ]
              [,INFO = { LONG | DUMP | SHORT } ]
              [,OUTFILE = filename ]
```

fgg-name bereitet alle Dump-Dateien des Dateiverzeichnisses *fgg-name* auf. Für jeden Workprozess der Anwendung erzeugt openUTM eine eigene FGG-Datei.

EDIT Dieser Operand steuert die Aufbereitung

Bei EDIT=PRINTER (Standard) wird eine Ausgabedatei erzeugt, die zum Ausdruck auf dem Drucker bestimmt ist. Die Ausgabe enthält Vorschubsteuerzeichen, Seitenköpfe und maximal 132 Zeichen pro Zeile. Das Inhaltsverzeichnis am Ende der Ausgabe bezieht sich auf Druckseiten (Standard).

Bei EDIT=TERMINAL wird die Ausgabedatei so aufbereitet, dass man sie mit einem Editor am Bildschirm auswerten kann. Die Zeilen enthalten maximal 80 Zeichen. Das Inhaltsverzeichnis am Ende der Ausgabe bezieht sich auf Zeilennummern.

INFO= Dieser Operand steuert die Ausgabe der Summary-Information. Sie ist ein Extrakt der gesamten Dump-Information und enthält die Daten, die bei der Diagnose häufig benötigt werden.

INFO=LONG: Aufbereitung mit Summary-Information (Standard).

INFO=DUMP: Aufbereitung ohne Summary-Information.

INFO=SHORT: KDCDUMP gibt nur Summary-Information aus.

OUTFILE= Mit diesem Operanden können Sie festlegen, dass die Ausgabe in eine Datei mit dem Namen *filename* geschrieben werden soll.

Ist OUTFILE nicht angegeben, vergibt KDCDUMP den Standardnamen `dmp1st/anwendungsname/RRRRRR/PPPP.NN` (Erläuterung siehe [Seite 60](#)) und schreibt die Ausgabedatei in das aktuelle Dateiverzeichnis.

Hinweise

- Bei der Eingabe der Steueranweisung FGG wird eine UTM-Dump-Datei, die mit dem Kommando DUMP eingelesen wurde, vollständig aus dem Speicher entfernt. Damit kann der vor dem FGG-Kommando untersuchte UTM-Dump nicht weiter diagnostiziert werden.
- Die einzelnen Dateigenerationen der FGG werden nacheinander gelesen und nach dem Bearbeiten wieder aus dem Speicher entfernt, so dass nach Ausführung des Kommandos keine Datei mehr im Speicher ist.

FILE Einzelne Dump-Datei aufbereiten

Mit dieser Steueranweisung wird eine einzelne Dump-Datei aufbereitet. Das Ergebnis der Aufbereitung wird in eine Ausgabedatei geschrieben. Der Name der Ausgabedatei wird mit dem Operanden OUTFILE festgelegt. Fehlt die Angabe, vergibt KDCDUMP einen Standardnamen.

```
FILE_      [ dumpfile ]
           [,EDIT={ PRINTER | TERMINAL } ]
           [,INFO= { LONG | DUMP | SHORT } ]
           [,OUTFILE = filename ]
```

dumpfile Name der UTM-Dump-Datei. Diese Datei kann auch einer FGG angehören.

Fehlt der Stellungsoperand, so wird vorausgesetzt, dass eine UTM-Dump-Datei mit der DUMP-Anweisung schon eingelesen worden ist. Auf diese wird dann die FILE-Anweisung angewandt. Die UTM-Dump-Datei wird dabei nicht aus dem Speicher entfernt.

EDIT Dieser Operand steuert die Aufbereitung.

Bei EDIT=PRINTER (Standard) wird eine Ausgabedatei erzeugt, die zum Ausdruck auf dem Drucker bestimmt ist. Die Ausgabe enthält Vorschubsteuerzeichen, Seitenköpfe und maximal 132 Zeichen pro Zeile. Das Inhaltsverzeichnis am Ende der Ausgabe bezieht sich auf Druckseiten (Standard).

Bei EDIT=TERMINAL wird die Ausgabedatei so aufbereitet, dass man sie mit einem Editor am Bildschirm auswerten kann. Die Zeilen enthalten maximal 80 Zeichen. Das Inhaltsverzeichnis am Ende der Ausgabe bezieht sich auf Zeilennummern.

INFO= Dieser Operand steuert die Ausgabe der Summary-Information. Sie ist ein Extrakt der gesamten Dump-Information und enthält die Daten, die bei der Diagnose häufig benötigt werden.

INFO=LONG: Aufbereitung mit Summary-Information (Standard).

INFO=DUMP: Aufbereitung ohne Summary-Information.

INFO=SHORT: KDCDUMP gibt nur Summary-Information aus.

OUTFILE= Mit diesem Operanden können Sie festlegen, dass die Ausgabe in eine mit dem Namen *filename* geschrieben werden soll.

Ist OUTFILE nicht angegeben, vergibt KDCDUMP den Standardnamen `dmp1st/anwendungsname/RRRRRR/PPPP.NN` (Erläuterung siehe [Seite 60](#)) und schreibt die Ausgabedatei in das aktuelle Dateiverzeichnis.

Hinweis

- Es ist zu beachten, dass eine UTM-Dump-Datei, die mit dem Kommando DUMP eingelesen worden ist, vollständig aus dem Speicher gelöscht wird, wenn die FILE-Steuerung mit dem Stellungsoperanden genutzt wird. Nach Angabe einer FILE-Anweisung mit Dateinamen wird die Datei vollständig aus dem Speicher gelöscht.
- In den Listen werden Bindestriche in den Tabellen-Namen (z.B. CPTRT-NSR) durch Unterstriche (z.B. CPTRT_NSR) ersetzt.

FIND Tabelleneintrag suchen und anzeigen

Mit FIND wird der Tabelleneintrag eines UTM-Betriebsmittels angezeigt. Die Art der Anzeige ist die gleiche wie beim Kommando TABLE.

FIND kann nur ausgeführt werden, wenn der aktuelle Dump einen KAA-Teil enthält. Dumps mit REASON PENDER oder FMterr enthalten z.B. keinen KAA-Teil.

```
FIND_      { *ANY | type }
           [ {, *ANY | kdcdef-name } ]
```

***ANY** Alle unter *type* angegebenen UTM-Typen aus der KDCDEF-Anweisung werden durchsucht.

type Typ des UTM-Betriebsmittels. Folgende Typen können angegeben werden:

type	Bedeutung	type	Bedeutung
AC	ACCESS-POINT	PO	(LTERM-) POOL
BC	BCAMAPPL	PR	PROGRAM
ED	EDIT	PT	PTERM, CON, OSI-CON
GB	GSSB	SB	KEYSET
LC	LTAC	TC	TAC
LM	LMOD	TL	TLS
LS	LSES	UL	ULS
LT	LTERM, LPAP, OSI-LPAP	US	USER

Ist das zu suchende Betriebsmittel ein Sekundärspeicher-Bereich, wird der entsprechende Eintrag der Tabellen DICN-NSR bzw. DICN-SR angezeigt. Sonst wird entweder der Eintrag in der jeweiligen NSR-Tabelle oder einer SR-Tabelle ausgegeben.

***ANY** Es werden alle KDCDEF-Namen ausgegeben, die zu dem angegebenen *type* gefunden werden.

kdcdef-name Name aus der KDCDEF-Generierung. Dieser Name muss als C-String eingegeben werden.

Falls KDCDUMP das Betriebsmittel nicht eindeutig identifizieren kann, weil z.B. *kdcdef-name* nicht angegeben wurde, dann bietet KDCDUMP Vorschläge einzeln zur Auswahl. Diese müssen mit „+“ oder „>“ quittiert werden:

+ Anzeigen des zugehörigen Eintrages. Nach der Anzeige kann mit „>“ weitergeblättert werden.

> weitersuchen.

Statt „+“ oder „>“ kann auch ein neues Kommando eingegeben werden.

HELP Hilfe zu KDCDUMP

Die Kommando HELP liefert Informationen zur Bedienung des Programms KDCDUMP. Diese Informationen werden auf *stderr* ausgegeben.

```
HELP_ [ { ALL | command-name | TABLE-NAMES } ]
```

HELP ohne Operanden gibt nur eine Liste aller KDCDUMP-Anweisungen aus. Die Operanden haben folgende Bedeutung:

ALL Gibt eine Übersicht über die KDCDUMP-Anweisungen aus. Standardwert.

command-name
Name einer KDCDUMP-Anweisung, zu der eine Kurzinformation ausgegeben wird.

TABLE-NAMES
gibt alle gültigen Tabellennamen aus. Die Buchstaben hinter dem Tabellennamen haben nur eine interne Bedeutung.

Die dynamische Tabelle XAP-DYNN-AREA und die Tabellengruppe XAP-LOC-BUFF können mit dem TABLE-Kommando nicht angezeigt werden.

Tabellengruppen werden mit einem dem Namen vorangestellten Stern (*) gekennzeichnet. Die einzelnen Tabellen der Tabellengruppe werden nach dem Namen der Tabellengruppe aufgelistet und sind mit einem vorangestellten Plus (+) gekennzeichnet.

Befindet sich ein Dump im Speicher, dann wird die Anzahl der Tabelleneinträge ausgegeben (in sedezimaler Form). Ausnahmen sind die SLOT-Tabellen: Hier wird der höchste Tabelleneintragsindex ausgegeben - die tatsächliche Anzahl der Tabellenelemente ist meist kleiner.

Auf Grund von negativen Ergebnissen bei der Adressvalidierung beim Erzeugen eines Dumps kann es vorkommen, dass UTM-Bereiche oder Tabellen nicht im Dump enthalten sind. Da in solchen Fällen keine Tabelleneinträge vorhanden sind, werden die Tabellen wie folgt gekennzeichnet:

UA_ERROR: Der UTM-Bereich, in dem die Tabelle liegt, konnte nicht abgezogen werden.

TA_ERROR: Die Tabelle konnte nicht abgezogen werden.

LIST Tabellenausschnitt aufbereiten

Mit dieser Steueranweisung kann man einen Tabellenausschnitt auf eine Datei schreiben lassen.

```
LIST_      table-name,listfile
           [, START-INDEX = { FIRST | start } ]
           [, END-INDEX = { LAST | end } ]
           [, EDIT = { TERMINAL | PRINTER } ]
```

table-name Name der Tabelle, von der ein Teil ausgegeben werden soll. Die gültigen Namen lassen sich mit `HELP TABLE-NAMES` erfragen. Als Tabellennamen können Sie auch einen Tabellengruppennamen angeben (bei der Ausgabe von `HELP TABLE-NAMES` mit einem Stern gekennzeichnet). In diesem Fall werden alle zu der Gruppe gehörenden Tabellen ausgegeben (gekennzeichnet durch ein vorangestelltes Plus (+)).

listfile Der Tabellenausschnitt wird auf die Datei „listfile“ ausgegeben. Eine bestehende Datei wird ohne Warnung überschrieben.

START-INDEX=

Eintrag der Tabelle, mit dem die Ausgabe beginnen soll:

FIRST Erster Eintrag der Tabelle, Standardwert.

start Tabelleneintragsindex, bei dem die Ausgabe beginnen soll. Dieser Wert kann sowohl dezimal als auch sedezimal eingegeben werden.

END-INDEX= Letzter Tabelleneintrag, der ausgegeben werden soll:

LAST Es wird bis Tabellenende ausgegeben, Standardwert.

end Tabelleneintragsindex, mit dem die Ausgabe beendet werden soll. Dieser Wert kann sowohl dezimal als auch sedezimal eingegeben werden.

EDIT= Aufbereitung für Drucker oder Terminal, siehe Beschreibung der Anweisung FGG auf [Seite 69](#).

Hinweise

- Das Kommando LIST unterstützt bei „Bittabellen“ den Tabellenlayouttyp „MEMORY LAYOUT“ nicht.
- Wenn man von einer existierenden SLOT-Tabelle oder der RSBF-Tabelle, die im UTM-Dump enthalten ist, einen Ausschnitt ausgeben lassen möchte, der nicht vorhanden ist, so wird keine Meldung ausgegeben.

Beispiel

Sie geben die folgende Anweisung ein:

```
LIST S-S-S,V.LIST,S-I=2,E-I=3
```

Falls zwar die Tabelle SLOT-SCB-STD, aber weder der Eintrag 2 noch 3 existiert, so wird die Datei V.LIST geschrieben, die nur den HEADER enthält.

SFIND String suchen

Mit dieser Anweisung können Sie im UTM-Dump einen String suchen und diesen auf dem Terminal ausgeben lassen. Sie können aber nur Strings suchen, die zu den Nutzinformationen im Dump zählen. Informationen, die vom Dump-Erzeuger KCSDDUMP zur Aufbereitung des Dumps angelegt worden sind, werden nicht mit dem Suchstring verglichen.

Für die Stacks gilt, dass nur der AUTOMATIC-Bereich durchsucht wird.

Weiterhin werden die Strings nur innerhalb eines Tabelleneintrages gesucht. D.h. beginnt der gesuchte String z.B. in der Tabelle USRT-NSR im Eintrag 1 und endet im Eintrag 2, so wird dies nicht als Treffer gezählt.

Bei einigen Tabellen (MPV, NPROGRAM-TABL, ...) werden die gefundenen Informationen aufbereitet ausgegeben, d.h. wenn der gesuchte String enthalten ist, wird die entsprechende Tabelle speziell aufbereitet angezeigt. Daher ist der gefundene String nicht immer unmittelbar zu sehen.

```
SFIND_ search
      [, ALIGN = {1 | 2 | 4 | 8}]
      [, HITS = {ALL | nr_max_hits}]
      [, DOMAIN = {ALL | KAA | KTA | AUTO | SLOT | ROOT |
                  CACHE | MGP | XATP-LOCAL | XATP-GLOBAL |
                  USERFILE | JOURNALFILE-1 | JOURNALFILE-2 } ]
                  BUFSEGMENTS | GSSBFILE | LOCKFILE |
                  CFGFILE | ULSFILE } ]
```

search Der Stellungsoperand steht für die gesuchte Zeichenkette, die maximal 190 Bytes umfassen darf. Die Angabe kann als C- oder X-String erfolgen.

ALIGN =	Hiermit kann die Ausrichtungsgrenze des Suchstrings angegeben werden.
1	Byte-Ausrichtung. Standardbelegung
2	2-Byte-Ausrichtung
4	4-Byte-Ausrichtung
8	8-Byte-Ausrichtung
HITS =	Mit HITS kann angegeben werden, nach wieviel Treffern bei der Suche die Funktion beendet wird.
ALL	Der Dump wird vollständig durchsucht. Standardbelegung
nr_max_hits	Die Suche wird nach <i>nr_max_hits</i> Treffern beendet. <i>nr_max_hits</i> kann einen Wert zwischen 0 und 32767 annehmen. 0 wird wie ALL behandelt.
DOMAIN	Damit kann angegeben werden, welche Teile des UTM-Dumps durchsucht werden sollen.
ALL	Gesamten Dump durchsuchen Standardbelegung
KAA	KDC Application Area durchsuchen (Anwendungs-globale Systemspeicher)
KTA	KDC Task Area durchsuchen (Prozess-spezifische Systemspeicher)
AUTO	AUTOMATIC STORAGES durchsuchen (Arbeitsspeicherbereiche der unterbrochenen UTM-Module)
SLOT	Slots durchsuchen (dynamisch angelegte Tabellen)
ROOT	KDCROOT -Bereiche mit den DIAGAREAS durchsuchen
CACHE	Speicherbereiche des CACHE durchsuchen
MPGP	Memory Pagepool durchsuchen
XAPTP-LOCAL	Prozesslokale Bereiche des XAPTP-Bausteins durchsuchen
XAPTP-GLOBAL	Globale Bereiche des XAPTP-Bausteins durchsuchen
USERFILE	Cluster-User-Datei durchsuchen

JOURNALFILE-1,JOURNALFILE-2
Datei(en) des Cluster-Administrations-Journals durchsuchen

BUFSEGMENTS
Segmente der Knoten-lokalen Pufferverwaltung durchsuchen.

GSSBFILE
Cluster-GSSB-Datei durchsuchen.

LOCKFILE
Cluster-Lock-Datei durchsuchen.

CFGFILE
Cluster-Konfigurations-Datei durchsuchen.

ULSFILE
Cluster-ULS-Datei durchsuchen.

SH und SYS KDCDUMP unterbrechen

Die Anweisung SH bzw. SYS startet aus dem Programm KDCDUMP heraus eine Bourne-Shell unter Unix-Systemen bzw. öffnet ein Eingabeaufforderungs-Fenster unter Windows-Systemen, so dass Sie Kommandos eingeben können.

```
{ SH | SYS }
```

Mit dem Kommando `exit` können Sie zum Programm KDCDUMP zurückkehren. Die gestartete Shell wird beendet bzw. das Eingabeaufforderungs-Fenster geschlossen.

SYSLST Protokollierung ein/auschalten

Diese Anweisung dient dazu, die Ergebnisse der Anweisungen AFIND, FIND, HELP TABLE-NAMES und TABLE auf *stdout* auszugeben.

Die Länge einer Ausgabezeile beträgt maximal 80 Zeichen. Es werden keine Meldungen auf *stdout* geschrieben.

```
SYSLST_ { ON | OFF }
```

ON Es wird auf *stdout* ausgegeben.

OFF Keine Ausgabe auf *stdout*.
Beim Start von KDCDUMP gilt „SYSLST OFF“.

TABLE Tabelle anzeigen

Mit der Anweisung TABLE kann eine Tabelle des gerade zu bearbeitenden UTM-Dump teilweise oder ganz ausgegeben werden. Die Ausgabe kann im Dump-Format oder symbolisch erfolgen.

```
TABLE_      table-name
            [, start-index
              [, { END-INDEX = { SAME | LAST } |
                DISPL = displacement
              }
            ] ]
```

table-name Maximal 13 Zeichen langer Name der Tabelle, die ausgegeben werden soll. Mit dem Kommando `HELP TABLE-NAMES` kann man sich eine Liste aller Tabellennamen anzeigen lassen.

Die dynamische Tabelle XAP-DYNN-AREA und die Tabellengruppe XAP-LOC-BUFF können mit TABLE nicht angezeigt werden, sondern nur mit der Anweisung LIST.

Die Tabellennamen dürfen abgekürzt werden, solange die Eindeutigkeit gewahrt bleibt. Dabei gilt folgende Regel:

- Am Anfang und nach einem Bindestrich muss mindestens ein Zeichen vorhanden sein. Dieses Zeichen muss das erste Zeichen im ausgeschriebenen Namen sein.
- Die einzelnen Namensteile, die mit einem Bindestrich beginnen und durch den nächsten Bindestrich oder das Ende begrenzt werden, können entfallen, wenn der nachfolgende Namensteil - so weit einer vorhanden war - auch entfernt wird.

Beispiel

Der Tabellename SLOT-VGT-DYN kann abgekürzt werden als SL-V, S-VGT oder S-V-D etc.

start-index Dieser Stellungsoperand gibt die Nummer des Eintrages an, ab dem die Ausgabe erfolgen soll.

Standardwert ist 0.

Erhält der Operand den Wert 0, so wird die ganze Tabelle ausgegeben, alle weiteren Operanden sind wirkungslos. Ausnahmen bilden hier die SLOT-Tabellen und der STACK-Bereich:

- Für SLOT-Tabellen ist der zweite Stellungsoperand *nicht* optional, sondern muss sich einem gültigen SLOT zuordnen lassen.
- Für den STACK-Bereich werden beim Operandenwert 0 (Standardwert) alle Interruptadressen (absolut und relativ - Modulname + modulinterne Adresse) angezeigt. Ist der Operandenwert n größer als 0, dann wird der zur Interruptadresse n gehörende Stack-Eintrag vollständig angezeigt.

Für beide Ausnahmen besitzen die folgenden Operanden keine Gültigkeit und sind bei Angabe wirkungslos.

Die Eingabe kann sedezimal oder dezimal erfolgen.

Hinweis

- Bei der Angabe Null wird bei den Trace-Tabellen der aktuelle Eintrag als Letztes auf der Bildschirmseite angezeigt.
- Wenn bei den Tabellen PTRM-NSR, LTRM-NSR, USRT-NSR, TACT-NSR, POOL-NSR, EDIT-NSR und LTAC-NSR ein bestimmter Eintrag angezeigt werden soll, so wird der Generierungsname mit ausgegeben. Für die ersten drei Tabellen werden im Falle einer bestehenden Verbindung beim Dump-Zeitpunkt Querverweise auf die anderen Tabellen gegeben.
- Für Bittabellen und Tabellen, deren Einträge immer zwei Bytes lang sind, kann i.a. nicht genau auf den gewünschten Eintrag positioniert werden. Der Eintrag befindet sich dann aber in der ersten ausgegebenen Zeile mit Tabelleninformationen.

END-INDEX= Mit diesem Operanden kann angegeben werden, ob außer dem bei „start-index“ angegebenen Eintrag noch weitere Tabelleneinträge angezeigt werden sollen.

SAME Es wird nur der Eintrag angezeigt, der mit „start-index“ angegeben wurde.

LAST Es wird die ganze Tabelle ab dem Eintrag „start-index“ angezeigt.

DISPL=displacement

Diese Möglichkeit kann man nutzen, um den im zweiten Stellungsoperanden angegebenen Tabelleneintrag mit einem Displacement ab Tabelleneintragsanfang auszugeben. Diese Möglichkeit wird nur für Tabellen im normalen Dump-Format, d.h. bei sedezimaler Darstellung, unterstützt.

Das Eingabeformat ist dezimal oder sedezimal (siehe Beispiel).

Für die optionalen Operanden END-INDEX und DISPL gilt der Standardwert END-INDEX=SAME.

Hinweise

- Sollte die ausgesuchte Tabelleninformation nicht auf eine Bildschirmseite passen, so können mit Hilfe der Blätter-Anweisungen (siehe dort) weitere Nutzdaten sichtbar gemacht werden (gilt nicht für die symbolische Aufbereitung).
- Das Kommando TABLE unterstützt bei „Bittabellen“ den Tabellenlayouttyp „MEMORY LAYOUT“ nicht.

Beispiel

1. T KAA,1,D=100

Gibt die KAA-Struktur sedezimal und abdruckbar aufbereitet ab dem Offset 100 aus.

2. T ROOT,1,S=*YES

Die Struktur ROOTDATA wird vollständig symbolisch aufbereitet ausgegeben.

3.2.3 Meldungen von KDCDUMP

Das Dienstprogramm KDCDUMP gibt Meldungen K7nn aus, die Meldungstexte und Zusatzinformationen zu den Meldungen findet man in weiteren Kapiteln dieses Dokuments.

3.3 Inhalt des UTM-Dumps

Im Folgenden wird der Aufbau eines UTM-Dumps beschrieben. Ein mit KDCDUMP aufbereiteter UTM-Dump kann folgende Informationen enthalten:

KAA den Anwendungs-globalen Systemspeicher.

KTA die Prozess-spezifischen Systemspeicher.

AUTOMATIC STORAGES

Stack-Bereiche der Aufrufhierarchie des utmwork-Prozesses.

KDCROOT mit den KDCROOT-Tabellen und den DIAGAREAs.

User File, Journal Files, ...

Bereiche für UTM-Cluster-Anwendungen.

SUMMARY Eine Zusammenfassung

Inhaltsverzeichnis.



Im Folgenden werden die Begriffe Task und Prozess synonym verwendet.

Wird der Dump auf Grund eines PENDING erzeugt (entweder vom Anwender programmiert oder intern erzeugt nach $KCRCC \geq 70Z$), enthält der Dump nur den KDCROOT-Bereich.

Zum leichteren Auffinden der Tabellen enthält der aufbereitete Dump am Ende eine Liste aller Tabellen mit den Seitennummern. Jede dieser aufbereiteten Tabellen hat eine Überschrift, welche die aufbereitete Tabelle identifiziert. Die Überschriften sind auch abhängig davon, ob der Dump für das Terminal oder für das Drucken aufbereitet wurde.

Die Überschrift beginnt jeweils mit dem abgezogenen Speicherbereich (KAA TABLE / KTA TABLE oder KDCROOT) sowie dem Tabellennamen.

Im Folgenden sind nur die aufbereiteten Tabellennamen aufgeführt.

Beispiel: (KAA TABLE) TRAN_SYN_NSR / Transfer Syntax NSR.

Sofern in dem Namensteil hinter dem Schrägstrich nur die Bindestriche durch Unterstriche oder Leerzeichen ersetzt sind, wurde dieser Namensteil im Folgenden weggelassen.

Hinweis

Speicherbereiche, die sedezimal aufbereitet werden, müssen Sie abhängig vom Maschinentyp auswerten. Die in diesem Kapitel verwendeten Beispiele sind mit INTEL-Maschinen erstellt.

3.3.1 Anwendungsglobaler Systemspeicher (KAA)

Der Speicherbereich KAA (KDCS application control area) ist Anwendungs-spezifisch, d.h. openUTM legt für jede laufende UTM-Anwendung einen eigenen Systemspeicher KAA an. Dieser Speicherbereich ist standardmäßig nur im Dump der ersten Prozess (Task) der Anwendung enthalten (Verursacher), siehe Startparameter DUMP-CONTENT.

Er besteht aus folgenden Bereichen und Tabellen (das Suffix NSR steht im Folgenden für „nicht sicherungsrelevant“, SR für „sicherungsrelevant, DSR für „dynamisch sicherungsrelevant“):

KAA	Bereich mit fester Struktur; enthält im Wesentlichen die Parameter der Generierungsanweisung MAX und andere, nicht sicherungsrelevante Parameter, die für die Laufzeit der Anwendung benötigt werden.
MPV	Memory Page Vektor, der die KAA-relativen Adressen für alle UTM-Seiten der KAA enthält
PGPT-NSR	Pagepool Bitliste
PGBT-NSR	Pagepool Blocktabelle (Bitliste)
PGCT-NSR	Pagepool Blockseitenzählertabelle
CATRT-NSR	Cache Translation Table
CPTRT-NSR	Cache Translation Table für Cluster-Pagepool (NSR).
HASH-NSR	Hash-Tabelle für DICN_NSR
MPGP-HL-CHNT	MEMORY-Pagepool High-Level Chain-Table
MPGP-LL-CHNT	MEMORY-Pagepool Low-Level Chain-Table
DICN-NSR	Dictionary of Names
TSKT-NSR	Tasktabelle
PTRM-NSR	physikalische Terminaltabelle
PTRM-DYN-NSR	poolrelative Adressen für dynamische phys. Terminaltabellen
DCB-DYN-NSR	poolrelative Adressen für dynamische UTM-D Kontrollblöcke
LTRM-NSR	logische Terminaltabelle
TACT-NSR	Transaktionscode Tabelle
USRT-NSR	Benutzertabelle
QUEU-NSR	Tabelle für temporäre Queues
VGT-DYN-NSR	poolrelative Adressen von dynamischen Vorgangstabellen

POOL-NSR	Terminalpool-Tabelle
FPMM-NSR	Tabelle für die Verwaltung von Asynchron-Aufträgen
TASEQ-NSR	Tabelle für die Reihenfolge von Transaktionen
LOG-CB-NSR	Logging-Exit Kontroll-Block
CARD-NSR	interne Verwaltungs-Tabelle (NSR)
CDCONT-NSR	poolrelative Adressen von eingelesener Ausweisinformation
BCAT-NSR	BCAM Application Table
ACPNT-NSR	OSI TP Zugriffspunkt-Tabelle
CON-NSR-EXT	OSI TP Erweiterungen für physikalische Terminaltabellen
PRGT-NSR	Programm-Tabelle
LMOD-NSR	Lademodul-Tabelle
LTAC-NSR	Local Tac Table für UTM-D
KSET-NSR	Tabelle mit Keysets
LPAP-NSR-EXT	Logical Partner Application Table Extension
CACT-NSR	Cache Control Table
CONS-ENTRIES	Consistency Entries
CL-CONS-ENTR	Consistency Entries der UTM-Cluster-Dateien (NSR)
SLOTMP-NSR	Slot Memorypool Control Table
UTMX-NSR	openUTM (Unix-/Windows-Systeme) spezifische Daten
MMOD-NSR	Tabelle der Meldungsmodule
LIBRARY-TABLE	Tabelle der Bibliotheken (Libraries) für KDCROOT (NSR)
AREA-TABLE	Area-Tabelle für KDCROOT (NSR)
DB-NSR	Tabelle mit der Generierungsinformation zu Datenbanksystemen
EXIT-TABLE	Tabelle der Exits für KDCROOT (NSR)
NODE-NSR	Node-Tabelle für OSI TP
INST-NSR	Tabelle der Instanzen für OSI TP
TRAN-SYN-NSR	Transfer Syntax Tabelle für OSI TP
AB-SYN-NSR	Abstract Syntax Tabelle für OSI TP
APL-CNTXT-NSR	Application Context Tabelle für OSI TP

PRCP-NSR	interne Verwaltungstabelle für Principal-Einträge (NSR)
CLND-NSR	Adressinformation für die Knoten eines Clusters
SLOG-NSR	SYSLOG-Pufferbereich (NSR)
CLFB-NSR	Daten der Cluster-Konfigurationsdatei vom Zeitpunkt des letzten Lesezugriffs
KAA-SR	Bereich für globale Ablaufparameter
KAA-SR-MAP	Bitliste für gültige KAA-Seiten auf der KDCFILE
HASH-SR	Hash-Tabelle für DICN_SR
HASHQU-SR	Hash-Tabelle für QUEU_SR
DICN-SR	Dictionary of Names
LTRM-SR	logische Terminaltabelle
TACT-SR	Tabelle der Transaktionscodes
USRT-SR	Benutzertabelle
QUEU-SR	Tabelle für temporäre Queues
VGT-SR	Vorgangstabelle
POOL-SR	Terminalpool-Tabelle
FPMM-SR	Tabelle für die Verwaltung von Asynchron-Aufträgen
RVGT-SR	Remote Vorgang Table
CHNT-SR	Chaintable zur Verwaltung des Pagepools auf der KDCFILE
NODE-SR	Node-Tabelle für OSI TP
KAA-DSR	Tabelle für globale Ablaufparameter
KAA-DSR-MAP	Bitliste für gültige KAA-Seiten auf der KDCFILE
TSKT-DSR	Task-Tabelle
HASHPT-DSR	Hash-Tabelle für PTRM_DSR
HASHLT-DSR	Hash-Tabelle für LTRM_DSR
HASHTC-DSR	Hash-Tabelle für TACT_DSR
HASHUS-DSR	Hash-Tabelle für USRT_DSR
HASHPR-DSR	Hash-Tabelle für PRGT_DSR
HASHKS-DSR	Hash-Tabelle für KSET_DSR
HASHLC-DSR	Hash-Tabelle für LTAC_DSR

PTRM-DSR	PTERM-Tabelle
LTRM-DSR	LTERM-Tabelle
TACT-DSR	TAC-Tabelle
USRT-DSR	USER-Tabelle
PRGT-DSR	Programm-Tabelle
LMOD-DSR	Lademodul-Tabelle
KSET-DSR	Keyset-Tabelle (= Schlüsselbund-Tabelle)
LTAC-DSR	Local TAC Tabelle für UTM-D
RSAKEY-DSR	Tabelle für RSA-Keys (DSR)
CPCH-DSR	Chain Table für Cluster-Pagepool (DSR)

3.3.1.1 Die Tabelle CONS_ENTRIES

Die Tabelle CONS_ENTRIES bietet Platz für 10 Einträge und wird zyklisch beschrieben, d.h. der 11. Eintrag überschreibt den 1. Eintrag.

Einträge in die Tabelle werden bei folgenden Ereignissen geschrieben:

- bei der KDCDEF-Generierung
- bei einem Aufruf von KDCUPD
- bei jedem Start der UTM-Anwendung

Ein Tabelleneintrag ist folgendermaßen aufgebaut:

typ	r1	r2	f1	i1	i2	i3	f2	yymmdd	hhmmss	tttt	Byte
4	1	1	2	1	1	1	1	6	6	4	

typ 'DEF' für KDCDEF-Lauf
 'UPD' für KDCUPD-Lauf
 'STRT' für Anwendungsstart.

r1 und r2 bezeichnen den Korrekturzustand von KDCDEF, KDCUPD und UTM-Systemcode (z.B. '10' für V6.3A10).

r1,r2 'nm' zwei Ziffern als Source-Korrekturzeichen
 '00' Leerzeichen für die Erstfreigabe einer Version.

f1, f2 Platzhalter (filler).

Für *typ* 'STRT' und 'DEF' enthalten i1,i2 und i3 zusätzliche Informationen.

Für *typ* = STRT

i1 'C' für UTM-Kaltstart
'W' für UTM-Warmstart

i2 'P' für Produktion

i3 'B' für Batch-Task
'D' für Dialog-Task

Für *typ* = DEF

i2 '-' wenn die KDCFILE erfolgreich erzeugt wurde
'W' wenn die KDCFILE fehlerhaft erzeugt wurde.

yymmdd
Datum (Jahr, Monat, Tag).

hhmmss
Uhrzeit (Stunde, Minute, Sekunden).

tttt TSN des UTM-Tasks bei *typ* = STRT.

3.3.1.2 CACHE-Buffer

CACHE-BUFFER Cache-Speicher

3.3.1.3 UTM SLOT POOLS

Dynamisch angelegte Tabellen (Slots):

SLOT-CARD	Inhalte eingelesener Ausweiskarten
SLOT-PTRM-DYN	dynamische phys. Terminaltabellen
SLOT-DCB-DYN	dynamische UTM-D Kontrolltabellen
SLOT-VGT-DYN	dynamische Vorgangstabellen
SLOT-LOG-CB	dyn. Kontroll-Tabelle für Logging-Exit

3.3.2 Anwendungsglobaler Systemspeicher von XAP-TP

Der Speicherbereich von XAP-TP enthält Anwendungs-globale Tabellen und Bereiche, die XAP-TP zur Steuerung des Anwendungslaufs anlegt. Der Speicherbereich existiert nur, wenn die UTM-Anwendung mit OSI TP-Kommunikation generiert ist. Der Bereich enthält nur nicht-sicherungsrelevante Informationen.

Der Speicherbereich ist standardmäßig nur im Dump des ersten Prozesses (=Task) der Anwendung enthalten (Verursacher), siehe Abschnitt „[Reduzierung der Dump-Information über Startparameter DUMP-CONTENT](#)“ auf Seite 61.

Der Speicherbereich besteht aus folgenden Bereichen und Tabellen:

XAP-SHAREMEM	Fixer Teil des globalen Speicherbereichs des XAP-TP-Bausteins; enthält u. A. Generierungswerte für die OSI TP-Kommunikation.
XAP-INST	Instanz-Tabelle für XAP-TP
XAP-TRAN-SYN	Transfer Syntax Tabelle für XAP-TP
XAP-AB-SYN	Abstract Syntax Tabelle für XAP-TP
XAP-ACCPT	OSI TP-Zugriffspunkt (Access Point)-Tabelle für XAP-TP
XAP-PARTNER	Partner-Tabelle für XAP-TP
XAP-ASSOC	Association Tabelle für XAP-TP
XAP-CAF	Channel Auxiliary Facility Tabelle für XAP-TP
XAP-DIAL	Dialog-Tabelle für XAP-TP
XAP-LOG-DAM	Log-Damage Record Tabelle für XAP-TP
XAP-RCH-GROUP	Recovery Context Handle Group Tabelle für XAP-TP
XAP-DYNNM-AREA	Dynamischer globaler Speicher für XAP-TP, dessen Größe der Parameter OSI-SCRATCH-AREA der MAX-Anweisung bestimmt.

3.3.3 Der Prozess-spezifische Systemspeicher (KTA)

Für jeden Prozess (Task) einer Anwendung wird der Systemspeicher KTA (KDCS task specific area) angezeigt. Dieser besteht aus folgenden Bereichen und Tabellen:

KTA	Bereich mit Task-spezifischen Ablaufparametern und Arbeitsbereichen
TACB	Transaction Control Block
PIBF	Processing Item Buffer
FILT	File Table
BCAT	BCAM Application IDs
ADMP	Addresses of Shared Memories
ILBF	Input Letter Buffer
OLBF	Output Letter Buffer
OSBF	OSI TP MSG-Buffer
ILBF-SNA	Input Letter Buffer SNA
OLBF-SNA	Output Letter Buffer SNA
PPLT	PAM Parameterlist Table
SPB	Station Level Parameter Block
VTCB	VTSU Control Block
FCBP	File Control Blocks PAM
FCBS	File Control Blocks SAM
TPGA	Task Program Table
MSGB	Meldungspuffer
MUXI	Eingabepuffer für MUX-Protokoll-Header
MUXO	Ausgabepuffer für MUX-Protokoll-Header
LTERM-BUNDLE	Master LTERM Bundle Tabelle
LPAP-BUNDLE	Master LPAP Bundle Tabelle
FMGT	FMGT File Management Table
LCBT	Lock Control Block für CLUSTER (globale Locks)
ETPNDS-TPR	Liste der im Subsystem UTM eingebundenen Komponenten mit ihren ETPNDs

TRCA	Trace Area
USBF	In UTM-Cluster-Anwendungen: User-File Buffer
JFBF	In UTM-Cluster-Anwendungen: Journal-File Buffer
CRBF	Confirmatory Record Buffer
REST	Restart-Kontrollbereich (nur bei Dumps im Warmstart)
RSBF	Restart-Pufferbereiche (nur bei Dumps im Warmstart)

3.3.4 Prozess-spezifischer Systemspeicher von XAP-TP

Der Speicherbereich existiert nur, wenn die UTM-Anwendung mit OSI TP-Kommunikation generiert ist.

Er besteht aus folgenden Bereichen und Tabellen:

XAP-LOCALMEM	Bereich mit Prozess-spezifischen Ablaufparametern und Arbeitsbereichen
XAP-TRACE	Trace-Bereich von XAP-TP
XAP-LOC-ACCPNT	Prozess-spezifische Tabelle für OSI TP-Zugriffspunkte (Access Points)
XAP-LOC-BUFF	Dynamischer lokaler Speicher, der eine Gruppe von Tabellen enthält (XAPL-CONO, XAPL-CTRE, XAPL-CTAC, XAPL-SCRA, XAPL-OSSB, XAPL-UDAT, ...). Diese Bereiche können nicht interaktiv mit dem TABLE-Kommando aufbereitet werden.

3.3.5 Der KDCROOT-Bereich

Es werden Tabellen und Bereiche aus KDCROOT ausgedruckt, diese können in vielen Fällen zur Diagnose von Anwenderfehlern herangezogen werden. Bei PEND ER-Dumps werden nur diese Daten zur Verfügung gestellt.

Im Einzelnen werden folgende Bereiche ausgegeben:

CONTEXT-AREA / Context Area

Dieser Bereich wird nur ausgegeben, falls der Dump durch ein Signal verursacht wurde. Er enthält die Adresse der Unterbrechung. Diese Adresse wird noch symbolisch als Funktion + Displacement ausgegeben.

PROGRAM-TABL / Program Table

Programmtabelle. Ein Eintrag enthält u.A. den Programmnamen und die Startadresse eines Teilprogramms.

LOAD-MODULE / Load Module Table

Bereich mit Informationen zu den Shared Objects.

AREA / Area Table

Bereich mit Informationen zu den generierten Areas

EXIT / Exit Table

Bereich mit Informationen zu den Exits (enthält Indizes in PRGT)

LIB / Library Table

Bibliothekstabelle

MEMORY-POOL / Memory Pool Table

Bereich mit Informationen zu den Memory Pools. Im Falle einer Generierung ohne Lademodule steht hier nur die Information aus der Generierung, weitere Information (z.B. die Adresse des Memory Pools) ist im Bereich User Root zu finden.

MSG-MODULE / Message Mod Table

Bereich mit Informationen zu den Meldungsmodulen

UTM-DIAGAREA / UTM Diagarea

Bereich mit Diagnose-Informationen zu allen KDCS-Aufrufen

KB / Kommunikationsbereich KB

Kommunikationsbereich, bestehend aus KB-Kopf, KB-Rückgabeinformation und KBPROG in der generierten Länge

SPAB / Standardprimärer Arbeitsbereich SPAB

Standard-Primärer Arbeitsbereich

MPUT-BUFFER / MPUT Buffer

Zwischenspeicher für MPUT-Nachrichten

- FORMUSER-BUF / FORMUSER Buffer
 - Pufferbereich mit logischer Ein/Ausgabe-Nachricht
- RESTART-BUFF / RESTART Buffer
 - Wiederanlaufbereich für die Bildschirmformatierung (in Unix- und Windows-Systemen nicht relevant)
- IO-BUFFER / IO Buffer
 - Pufferbereich mit physikalischer Ein-/Ausgabe-Nachricht
- ROOTDATA / ROOTDATA
 - Verständigungsbereiche zwischen KDCROOT und den UTM-Systemmodulen
- ROOT-TRACE / Root Trace
 - Bereich mit Trace-Records zum Root-Ablauf
- FORM-USER-AR
 - Schnittstellen-Parameter zum Formatierungssystem (in Unix- und Windows-Systemen nicht relevant)
- HLL-USER-ARE / HLL User Area
 - Parameterliste der IUTMHLL
- IPC-HEADER
 - Verwaltungsbereich für das IPC-Shared Memory
- IPC-FREE-QUEU
 - Anker für die Listen von freien Objekten von IPC-ELEMENT, IPC-LETTER und IPC-ANNO
- IPC-TIMER-ID
 - Identifikation für utmtimer-Aufträge
- IPC-SEMA
 - Semaphor-Tabelle
- IPC-APPL-GLOB
 - Globale Anwendungsinformation
- IPC-APPL
 - Tabelle der Anwendungsnamen
- IPC-EXTP
 - Tabelle der externen Partner
- IPC-BRSE
 - Börsentabelle
- IPC-ELEMENTS
 - Bereich für IPC-ELEMENT-Objekte (Nachrichtentyp, -länge usw.)

- IPC-ANNOS
Bereich für IPC-ANNO-Objekte
- IPC-LETTER
Bereich für IPC-LETTER-Objekte (Nachrichteninhalt)
- IPC-VIRT_HOST
Tabelle mit den virtuellen Hostnamen aus der Mapped Hostname Datei
- IPC-REAL_HOST
Tabelle mit den realen Hostnamen aus der Mapped Hostname Datei
- IPC-SM2-DATA
Bereich mit den Messwerten für openSM2
- IPC-TRACE
IPC-Trace-Bereich
- VGM-AREA / VGM Area
Bereich für das Vorgangsmemory einer angeschlossenen Datenbank
- USER-ROOT / Root gen by user
Bereich enthält Daten aus dem benutzereigenen Root-Modul
- OSS-AREA / OSS Area
OSS Shared Memory
- NLS-AREA / Environment Area
Environment des Workprozesses
- XA-AREA / XA Area
Datenbankbereich
- TIMER-AREA / Timer Area
Time-Verwaltungsbereich
- TABDESC-AREA / Table Descriptors
Bereich mit Daten zu den aufbereiteten Root-Tabellen (enthält Name, Adresse, Anzahl der Entries und Länge)
- SHMPROT-AREA / SHMPROT Area
Bereich für Shared Memory Protection
- ADMI-DIAGAREA / Administration DIAGAREA
Bereich mit Trace-Records für alle Aufrufe der Programmschnittstelle der Administration aus den Teilprogrammen
- ADMI-USERAREA / Administration USERAREA
Bereich mit einem Trace-Record für die an der Programmschnittstelle der Administration vom Teilprogramm übergebenen Daten

LOGEXITBUFFER/LOG-EXIT-MESSAGE-BUFFER

Bereich für Logging-Exit

STRT-PAR

Bereich mit den zum Start der Anwendung angegebenen Start-Parametern

ACCOUNTING-A / Accounting Area

Bereich für Abrechnungsdaten

TAM / TAM

Transaktionsspeicher für eine angeschlossene Datenbank

TSKM / TSKM

Task-spezifischer Speicher für die Datenbankkommunikation

DB-DIAGAREA / DB Diagarea

Bereich mit Diagnose-Informationen zu allen Datenbank-Aufrufen (nur wenn Datenbank generiert ist)

DB-USER-AREA / DB User Area

Parameterliste der IUTMDB (nur falls DB generiert)

DB-INF-PROG / Db Info Program

Bereich enthält Daten zum aktuellen Teilprogramm

DB-INF-APPL / Db Info Application

Bereich enthält Daten zur Anwendung

DB-SUMMARY

Bereich mit allgemeinen Informationen zu den generierten Datenbanken (z.B. Anzahl etc.)

DB-TABLE

Tabelle der generierten Datenbanken

3.3.5.1 PROGRAM-Tabelle

Die Einträge haben folgenden Aufbau und folgende Bedeutung:

Stelle	Bedeutung
1	Index des Eintrages in der ROOT-Programm-Tabelle
2	Name des Programms
3	Sprachtyp des Programms wie in der Anweisung PROGRAM... COMP= angegeben. Mögliche Werte: C = X'06' COB2 = X'0A' (COB2 ist Aliasname für alle Micro Focus Compiler) CPP= X'0C' NETCOBOL=X'0D'
4	Lademodus des Programms wie in der Anweisung PROGRAM .. LOAD= angegeben. Falls die Anwendung mit Shared Objekts generiert ist, entspricht diese Ausgabe dem in der SHARED-OBJECTS-Anweisung (Parameter LOAD-MODE) angegebenen Wert. Bedeutung der Werte: STATIC = X'00': Programm statisch zur Anwendung gebunden STARTUP = X'01': Programm wird beim Start der Anwendung nachgeladen POOL = X'03': Programm wird in einen Memory Pool geladen.
5	Austauschmodus des Programms, immer mit NOTCH = X'00' belegt (nicht austauschbar).
6	Nicht relevant, immer mit X'00' belegt.
7	
8	
9	Ladestatus des Programms, immer mit LOADED = X'01' belegt (Programm geladen).
10	Adresse des Programms

3.3.5.2 LOAD-MODULE-Tabelle

Die Einträge haben folgende Bedeutung:

Byte sedezimal: 32-Bit (64-Bit)	Bedeutung
-	Index des Eintrags in der ROOT-Shared Object-Tabelle (LMOD)
24-43 (48-67)	Name des Shared Objects
44 (68)	Lademodus des Shared Objects, mögliche Werte: X'00' = Shared Object statisch zur Anwendung gebunden X'01' = Shared Object wird beim Start der Anwendung nachgeladen X'02' = Shared Object wird beim ersten Ansprung geladen X'03' = Shared Object wird in einen Memory Pool geladen.
45 (69)	nicht genutzt
46 (6A)	Austauschmodus des Shared Objects, mögliche Werte: X'00' = Shared Object nicht austauschbar X'01' = Shared Object einzeln austauschbar X'02' = (derzeit nicht verwendet) X'03' = Shared Object nur mit gesamter Anwendung austauschbar, da in lokalen Pool geladen.
47-5F (6B-83)	Versionsnummer des zu ladenden Shared Objects (bei Austausch)
60-61 (84-85)	Index des Contexts, zu dem dieses Shared Object gehört
62-63 (86-87)	Index des Memorypools (MPOOL), in den dieses Shared Object geladen wird
64-65 (88-89)	Index der Bibliothek (LIB), aus der dieses Shared Object geladen wurde
66-67 (8A-8F)	nicht genutzt
68-6B (90-97)	UTM-interne Info
6C-6D (98-99)	Index des ersten Programms in diesem Shared Object
6E-85 (9A-B1)	Versionsnummer des aktuellen (=zuletzt geladenen) Shared Objects
86-87 (B2-B3)	UTM-interne Info

Byte sedezimal: 32-Bit (64-Bit)	Bedeutung
88-89 (B4-B5)	Index des nächsten Shared Objects im gleichen Context
8A-8B (B6-B7)	Index der ersten AREA in diesem Shared Object
8C (B8)	Ladestatus des Shared Objects: X'00' = Shared Object nicht geladen X'01' = Shared Object ist geladen X'02' = Shared Object wird bei Ansprung eines Programms geladen, das in diesen Shared Object gebunden ist. X'03' =(derzeit nicht verwendet)

Ist Versionsnummer neu = Versionsnummer alt, so bedeutet dies, dass seit dem letzten KDCDEF-Lauf kein Austausch dieses Shared Objects stattgefunden hat.

3.3.5.3 UTM-DIAGAREA

Die UTM-DIAGAREA ist ein Prozess-spezifischer Trace-Bereich, in den alle Ereignisse protokolliert werden. Dieser Bereich enthält somit auch alle Ereignisse, die unmittelbar vor dem Abbruch eines Vorgangs oder einer Anwendung aufgetreten sind.

Die UTM-DIAGAREA wird zyklisch beschrieben. Zwei Zyklen werden durch eine Trennlinie bestehend aus '='-Zeichen und Leerzeichen getrennt. Oberhalb der Trennlinie steht der jüngste Eintrag und unterhalb davon der älteste Eintrag. Jeder Eintrag ist 136 bzw. 256 (64-Bit) Bytes lang. Wieviele Einträge insgesamt in die UTM-DIAGAREA passen, ist abhängig vom Generierungsparameter `MAX TRACEREC`.

In die UTM-DIAGAREA werden folgende Arten von Einträgen geschrieben:

- UTM-Records (Typ `KDCS`)

UTM-Records werden bei folgenden Ereignissen geschrieben:

- bei einem `KDCS`-Aufruf in einem Teilprogramm oder
- bei einem internen Aufruf an den UTM-Systemcode oder
- bei einem internen `PEND ER`-Aufruf durch `openUTM` (System-`PEND ER`) auf Grund eines schwer wiegenden Fehlers

Im Falle eines System `PEND ER` enthält der Eintrag in den Bytes 22 - 57 einen Fehlertext.

Für Aufrufe der Administrationsschnittstelle (`KDCS-Opcode=ADMI`) werden zusätzliche Trace-Informationen geschrieben ([Abschnitt „ADMI-DIAGAREA“ auf Seite 112](#)).

- UTM-Records zur Vorgangsidifizierung (Typ `VGID`)

Ein `VGID`-Record wird bei jedem Start eines Teilprogramms und bei Rückkehr eines `PGWT`-Aufrufes in die UTM-DIAGAREA geschrieben.

Header der Records in der UTM-DIAGAREA

Jeder Record beginnt mit einem Header, der folgende Informationen enthält:

Byte	Bedeutung
0 - 1	Zähler der aktuellen Einträge in den DIAGAREAs (UTM und DB)
2 - 5	Typ-Identifizierung (<code>KDCS</code> , <code>VGID</code> , <code>FHCL</code> , <code>ITRC</code>)
6 - 7	Derzeit nicht genutzt (mit '=' vorbelegt)
8 - 15	Zeitstempel

Aufbau des Headers

Der Inhalt der Records ab Byte 16 ist abhängig vom Typ des Records.

Aufbau der UTM-DIAGAREA bei KDCS-Aufruf aus einem Teilprogramm

Byte: 32-Bit (64-Bit)	Feldname und Bedeutung	
16-19 (16-19)	KCOP ¹ :	USER-Operationscode: INIT, MGET, MPUT, usw.
		interner Operationscode: siehe Tabelle Seite 110
20-21 (20-21)	KCOM: Operationsmodifikation	
22-23 (22-23)	KCLA: Bereichslänge oder Queue Level (bei QCRE) oder KCLKBPRG: Länge KB-Programmbereich beim INIT	
24-25 (24-25)	KCLM: Nachrichtenlänge oder KCLPAB: Länge des Standard Primären Arbeitsbereichs beim INIT. KCWTIME: Wartezeit in Sekunden beim DGET	
26-33 (26-33)	KCRN: Bezugsname	
		MCOM-Aufruf
34-41 (34-41)	KCMF: Formatname oder Editprofilname KCLT: LTERM-Name des LTERM-Partners oder KCUS: Benutzerkennung oder KCPA: Name der Partner-Anwendung (beim APRO-Aufruf)	KCPOS: Ziel einer positiven Folgenachricht

Aufbau eines Eintrags der UTM-Diagarea bei einem KDCS-Aufruf

Byte: 32-Bit (64-Bit)	Feldname und Bedeutung					
42-43 (42-43)	KCDF: Bildschirmfunktion, siehe Tabelle auf Seite 101					KCNEG: Ziel einer negativen Folgenachricht
	DPUT/DADM	DGET	QCRE	PADM	APRO	
44 (44)	KCMOD: Modus ("A"/"R"/" _")	KCQTYP: Zieltyp ("U"/"Q"/"T")	KCQMODE: Modus ("S"/ "W"/binär null)	KCACT: Aktion (ON/ OFF/CON/ DIS)	KCPI: Vorgangs- identifi- kation	KCCOMID: Komplex- identifikation
45-47 (45-47)	KCTAG: Tag	leer	leer	KCADRLT: Neuer LTERM- Name des Druckers		
48-49 (48-49)	KCSTD: Stunde					
50-51 (50-51)	KCMIN: Minute	leer	leer	leer	KCOF: OSI TP Funktionen	
52-53 (52-53)	KCSEC: Sekunde					
54 (54)	KCQTYP: Zieltyp ("U"/"Q"/binär null)	leer	leer	leer	KCOF: OSI TP Funktionen	
55-57 (55-57)	leer					
58-89 (58-89)	KCRFELD in KCKBC: KB-Rückgabebereich (siehe Tabelle auf Seite 101)					
92-95 (96-103)	Rücksprungadresse zum Teilprogramm (Die Adresse zeigt hinter KDCS-Aufruf im Teilprogramm)					
96-99 (104-111)	Adresse des Benutzerdatenbereichs (2. Parameter bei KDCS-Aufruf).					
100-103 (112-119)	Vorgangs-Index					
104-111 (120-127)	KCLOGTER in KCKBC: LTERM - Name					
112-119 (128-135)	KCBENID in KCKBC: Name der aktuellen Benutzererkennung					

Aufbau eines Eintrags der UTM-Diagarea bei einem KDCS-Aufruf

¹ Bei KCOP=INFO (Byte 16-19) wird ohne Erhöhung des Zählers der Nachrichtenbereich (NB) des protokollierten Aufrufs in der Länge KCPAC in den nächsten Eintrag der UTM-DIAGAREA geschrieben. Bei Operationsmodifikation KCOM=CK (Byte 20-21) ist diese Information für die Diagnose interessant, da sie den zu prüfenden Aufruf protokolliert.

Gilt KCOP=INFO und KCOM=CK, wird ohne Erhöhung des Zählers der Nachrichtenbereich (NB) des protokollierten Aufrufs in der Länge KCPAC in den nächsten Eintrag der DIAGAREA geschrieben.

Bei internen Aufrufen von openUTM werden in das Feld KCOP folgende Codes eingetragen:

Inhalt KCOP	Situation, in der dieser Eintrag geschrieben wird	Einträge in nachfolgende Felder
STRT	Start des UTM-Anwendungsprogramms, Beginn Start-Behandlung im UTM-Systemcode	kein Eintrag
WAIT	UTM-Task hängt sich an die UTM-Work-Börse	kein Eintrag
CONT	Fortsetzung im UTM-System-Code nach einer DB-Aktion über KDCROOT oder nach Aufruf des INPUT-Exits.	nur KCRCCC, KCRCKZ und KCRCDC, beim INPUT-Exit die Parameter, siehe Seite 101
NOOP	Pufferbereich der MESSAREA muss geleert werden (nur möglich bei eingeschaltetem Mess-Monitor)	- - -
ADMI	UTM-Administrationsaktion	UTM-interne Schnittstelle

Aufbau von KCRFELD

58-59	KCRDF: Rückgabe Bildschirmfunktion KCRWVG: Anzahl wartender Vorgänge beim DGET		
60-61	KCRMLM: tatsächliche Länge der Nachricht		
	INFO CK-Aufruf	MGET-Aufruf	SIGN ON-Aufruf
62	KCRINFCC: KDCS-Returncode des geprüften KDCS-Aufrufs	KCVGST: Vorgangstatus	KCRSIGN1: Primärkode
63		KCTAST: Transaktionsstatus	KCRSIGN2: Sekundärkode
64		leer	
65	leer	KCRMGT: Nachrichtentyp	
66-68	KCRCCC: KCDS-Returncode		
69	KCRCKZ: Kennzeichen: P (Produktivanwendung)		
70-73	KCRCDC: interner Returncode		
74-81	KCRMF: Rückgabe Formatkennzeichen oder Editprofil		
82- 89	KCRPI: Rückgabe Vorgangs-Identifikation KCRUS: Benutzererkennung beim SIGN ST oder Erzeuger der Nachricht beim DGET KCRQN: Von openUTM vergebener Name der Temporären Queue bei QCRE NN		

DIAGAREA nach einem Aufruf des INPUT-Exits

Nach einem Aufruf des INPUT-Exit trägt openUTM in den KDCS-Record die wichtigsten Parameter ein, die der Exit bekommt bzw. setzt (KCPAC):

Byte	Feldname und Bedeutung
16-19	KCOP: Hier wird "CONT" eingetragen
20-25	leer
26-33	KCIFCH: Die ersten 8 Zeichen der Eingabe
34-35	KCICVST: Vorgangstatus: "ES"/"ET"/"RS"/"EC"
36-37	KCIFKEY: Wert der F-Taste: 1,...,24
38-39	KCIKKEY: Wert der K-Taste: 1,...,14
40-41	KCICFINF: Information über Steuerfelder: "UN"/"NO"/"ON"/"MO"
42-49	KCINTAC/KCINCMD: Nächster zu startender TAC bzw. Benutzer-Kommando
50-51	KCICCD: Code für die Wirkung der Eingabe: "ER"/"CC"/"SC"/"ST"/"CD"

Aufbau eines Eintrags der UTM-DIAGAREA nach einem Aufruf des INPUT-Exits

52	KCICUT: TAC abschneiden: "Y"/"N"
53	- - -
54-57	KCIERRCD: Fehlerinfo für Datensichtstation
58-...	Folgendes ist für die Diagnose nicht relevant

Aufbau eines Eintrags der UTM-DIAGAREA nach einem Aufruf des INPUT-Exits

Fehlertexte in der DIAGAREA bei SYSTEM-PEND ER

Zur schnelleren Diagnose bei einem SYSTEM-PEND-ER ist in der entsprechenden Zeile der UTM DIAGAREA in Byte 22-57 ein abdruckbarer Fehlertext abgelegt.

In der folgenden Tabelle sind alle möglichen Fehlertexte aufgeführt, dazu jeweils die Fehlerursache sowie mögliche Maßnahmen und die Fehlerart (Systemfehler oder Anwenderfehler).

Fehlertext	Ursache, Modul	Fehlerart
APPL. PROGRAM DOES NOT EXIST	Beim Start eines Teilprogramms war kein Indikator für ein Teilprogramm vorhanden (70Z mit KR01)	Systemfehler
APPL. PROGRAM WITHOUT PEND	Das Anwendungsprogramm wurde nicht mit PEND beendet. (84Z)	Anwenderfehler
ASYNC. PROGRAM NOT FOUND	Asynchron-Programm ist nicht mehr verfügbar (evtl. Programmaustausch). PEND ER-Dump mit 70Z und KR02 folgt.	Anwenderfehler
ERROR IN "START-TP" OF LGCON	Beim Start eines Teilprogramms lieferte der Language-Connection-Modul einen fehlerhaften Returncode.	Systemfehler
KB END LABEL OVERWRITTEN	Der KB im Anwendungsprogramm ist größer als in der Generierung angegeben. (70Z mit KR04) Maßnahme: Operand KB= in der MAX-Anweisung ändern.	Anwenderfehler
SPAB END LABEL OVERWRITTEN	Der SPAB im Anwendungsprogramm ist größer als in der Generierung angegeben (70Z mit KR05). Maßnahme: Operand SPAB= in der MAX-Anweisung ändern.	Anwenderfehler
KCRCCC > = 70Z AFTER UTM SVC	Nach SVC UTM wurde der KCRCCC im KB-Kopf auf ≥70Z gesetzt.	Anwender-/ Systemfehler

Fehlertexte in der UTM-DIAGAREA bei SYSTEM--PEND ER

Fehlertext	Ursache, Modul	Fehlerart
ROOTDATA CODE INVALID	Vom SVC UTM wurde der ROOTDATA-CODE auf ungültigen Wert gesetzt	Systemfehler
ERROR ROUTINE XT... ENTERED	Es ist ein SIGNAL mit der Nummer ... (in KDCRTDI) aufgetreten.	Anwender-/ Systemfehler
ERROR ROUTINE EXIT ENTERED	Es wurde ein unerlaubter exit()-Aufruf (COBOL: STOP RUN-Anweisung) während eines Teilprogramm-Ablaufs erkannt, siehe auch bei Returncode KRCDC= EXIT.	Anwenderfehler
DB-ERRORCODE = TA_CHAIN_RSET	Der Datenbank-Verbindungsmodul lieferte den Returncode TA_CHAIN_RSET. Wurde dieser Returncode durch eine K210, K211, K216 mit XAER_DUPID ausgelöst, ist es möglich, dass ein abnormales Ende des vorherigen Anwendungslaufs diese Transaktion in der Datenbank hinterließ und in diesem Zustand eine neue KDCFILE generiert wurde. Mögliche Maßnahme: Herunterfahren und Neustarten der Datenbank	Systemfehler
KDCS-CALL IN VORGANG EXIT	Im VORGANG-Exit-Programm wurde ein unerlaubter KDCS-Aufruf gemacht.	Anwenderfehler
VORGANGEXIT-PROGRAM NOT LOADED	Das Programm für den VORGANG-Exit ist nicht geladen.	Anwenderfehler
DATABASE DOWN AT USER DB CALL	Zum Zeitpunkt des CALL zur Datenbank aus dem Anwender-Teilprogramm ist die Datenbank nicht mehr konnektiert.	Anwender-/ Systemfehler
ILLEGAL RTCODE FROM DBCON	Der Datenbank-Connection-Modul setzt unerlaubten Returncode.	Systemfehler
NO DB CALL ALLOWED IN SIGN-ON	Der Anmelde-Vorgang hat einen unerlaubten DB-USER-CALL gegeben.	Anwenderfehler
PROGRAM INDEX = 0 INVALID	Vom SVC UTM wurde ein ungültiger Programm-Index gesetzt.	Systemfehler

Fehlertexte in der UTM-DIAGAREA bei SYSTEM--PEND ER

Eintrag zur Vorgangs-Identifikation (Typ VGID)

Byte: 32-Bit (64-Bit)	Bedeutung
16 (16)	Vorgangskennzeichen
17 (17)	Sessionzähler
18-19 (18-19)	Transaktionszähler im Vorgang
20-23 (24-31)	Vorgangszähler
24-27 (32-39)	Summe von USED und ERROR für den aktuellen TAC
28-31 (40-47)	Länge des Global Transaction Identifier (GTRID) der XID
32-35 (48-55)	Länge des Branch-Qualifier (BQUAL) der XID
36-67 (56-87)	die ersten 32 Bytes des Global Transaction Identifier (GTRID)
68-115 (88-137)	die ersten 48 Bytes des Branch-Qualifier (BQUAL)
116-117 (138-139)	Index zur Programm-Tabelle
118-119 (140-141)	Index des Vorgangsexit in Programm-Tabelle
120-127 (142-149)	Name des Transaktionscodes, der den Vorgang gestartet hat
128-135 (150-157)	Name des aktuellen Transaktionscodes

Aufbau des Eintrags zur Vorgangs-Identifikation

Beispiel

In den ersten beiden Beispielen wurde der sedezimale Teil des Dump-Eintrages weglassen, das dritte Beispiel zeigt, wie der sedezimale Teil bei der Auswertung von C-Namen und Längenfeldern (KCLM, KCLA, ...) benutzt wird.

1. Fehlerhafter Operationscode bei einem KDCS-Aufruf.

```

KDCROOT : UTM Diagarea
.
0012 08227F2C 0000    ...    ..INIT.....
      08227F4C 0020    ...    ..          .. 000P0000
0013 08227F90 0000    ...    ..MPUTNT.... *f0114 ..
      08227FB0 0020    ...    ..          .. 000P0000
0014 08227FF4 0000    ...    ..SIGNOB.....
      08228014 0020    ...    ..          .. 000P0000
0015 08228058 0000    ...    ..XXXXFI....
      08228078 0020    ...    ..          .. 79ZP0000
0016 082280BC 0000    ...    ..PENDER: KCRCC >= 70Z AFTER
      082280DC 0020    ...    UTM SVC !!
    
```

2. PEND ER durch Signal 11 wegen eines Adressfehlers

```

KDCROOT : UTM Diagarea
.
0004 082279B4 0000    ...    ..INIT..t.*.....
      082279D4 0020    ...    ..          .. 000P0000
      082279F4 0040    ...    ..          .. "J" ..
0005 08227A18 0000    ...    ..PENDER: ERROR ROUTINE XT11 ENT
      08227A38 0020    ...    ERED !!
      08227A58 0040    ...    .....POOLT001....
    
```

3. Auswertung der Längenfelder beim Dump

Bei den Angaben in den Längenfeldern müssen Sie die maschinenabhängige Art der Darstellung (little endian oder big endian) berücksichtigen! Bei little endian-Maschinen ist das Oktett mit der niedrigsten Adresse auch das niedrigstwertige Oktett, bei big endian-Maschinen ist das Oktett mit der niedrigsten Adresse das höchstwertige Oktett.

Den unterschiedlichen Aufbau verdeutlicht folgende Tabelle.

dezimale Darstellung	sedezimale Darstellung INTEL-Prozessor (little endian)	sedezimale Darstellung RISC-Prozessor (big endian)
228	E400	00E4

Bei Dezimalfeldern (unterstrichen) werden die Bytes in umgekehrter Reihenfolge dargestellt.

INTEL-Prozessor (z.B. Windows-Systemen)

```

0008  B66EC3CC 0000  07004B44 43533D3D FF86AC45 53A20D00 ..KDCS==...ES...
      B66EC3DC 0010  494E4954 00000000 00000000 00000000 INIT.....
      B66EC3EC 0020  00000000 00000000 00000000 00000000 .....
      B66EC3FC 0030  00000000 00000000 00002020 00002020 .....
      B66EC40C 0040  20203030 30503030 30302020 20202020 ..
      B66EC41C 0050  20202020 20202020 20203D3D 00000000 000P0000
      B66EC42C 0060  0A3C49B7 02000000 4C545030 30303031 ==....
0009  B66EC454 0000  08004B44 43533D3D FF86AC45 67A20D00 .<I.....LTP00001
      B66EC464 0010  4D474554 00006D01 00000000 00000000 ..KDCS==...Eg... 1
      B66EC474 0020  00002020 20202020 20200000 00000000 MGET..m.....
      B66EC484 0030  00000000 00000000 00000000 08004F43 ..
      B66EC494 0040  204D3030 30503030 30302020 20202020 .....OC
      B66EC4A4 0050  20202020 20202020 20203D3D 00000000 M000P0000
      B66EC4B4 0060  710756B7 02000000 4C545030 30303031 ==....
                                           q.V.....LTP00001

```

¹ Beim MGET-Aufruf wurde in KCLA = 365 (sedezimal: 016D) angegeben.

RISC-Prozessor (z.B. Solaris, HP-UX, AIX)

```

0008  0004B2EC 0000  00074B44 43533D3D 45AC8284 000B2204 ..KDCS==E....."
      0004B2FC 0010  494E4954 00000000 00000000 00000000 INIT.....
      0004B30C 0020  00000000 00000000 00000000 00000000 .....
      0004B31C 0030  00000000 00000000 00002020 00002020 .....
      0004B32C 0040  20203030 30503030 30302020 20202020 ..
      0004B33C 0050  20202020 20202020 20203D3D 00000000 000P0000
      0004B34C 0060  FFF25214 00000002 4C545030 30303031 ==....
0009  0004B374 0000  00084B44 43533D3D 45AC8284 000B2230 .R.....LTP00001
      0004B384 0010  4D474554 0000016D 00000000 00000000 ..KDCS==E....."0 1
      0004B394 0020  00002020 20202020 20200000 00000000 MGET...m.....
      0004B3A4 0030  00000000 00000000 00000000 00084F43 ..
      0004B3B4 0040  204D3030 30503030 30302020 20202020 .....OC
      0004B3C4 0050  20202020 20202020 20203D3D 00000000 M000P0000
      0004B3D4 0060  FF1E1315 00000002 4C545030 30303031 ==....
                                           .....LTP00001

```

¹ Beim MGET-Aufruf wurde in KCLA = 365 (sedezimal: 016D) angegeben.

3.3.5.4 DB-DIAGAREA

Bei jedem USER-CALL an das DB-System wird ein DB-Record in die DB-DIAGAREA geschrieben. Der Bereich wird, ebenso wie die UTM-DIAGAREA, zyklisch mit Trace-Records beschrieben.

Zwei Zyklen werden durch eine Trennlinie bestehend aus '='-Zeichen getrennt. Der Record oberhalb der Trennlinie ist der jüngste, der Record unterhalb der älteste.

Byte: 32-Bit (64-Bit)	Bedeutung
0-1	Zähler der aktuellen Einträge in den DIAGAREAs (UTM und DB)
2-5	"DBCL"= Kennzeichen für DB-Record
6-7	Derzeit nicht genutzt (mit '=' vorbelegt)
8-15	Zeitstempel
16-19 (16-19)	Status der DB-Transaktion ¹ vor dem DB-Aufruf, siehe Tabelle auf Seite 109
20-23 (20-23)	Status der DB-Transaktion ¹ nach dem DB-Aufruf, siehe Tabelle auf Seite 109
24 (24)	DB Operation Code, siehe Tabelle „DB Operation Codes“ auf Seite 110
25 (25)	DB Secondary Opcode
26 (26)	DB Error Code, siehe Tabelle „DB Error Codes“ auf Seite 111
27 (27)	Kennzeichen für DB-System (sedezimal): 02=XA-Interface
28-31 (28-31)	DB Trace Information des letzten XA-Aufrufs: 28: RM-Nummer, 29-30: XA-Funktionscode, 31: XA-Returncode
32-63 (32-63)	DB Trace Informationen der XA-Aufrufe dieser DB-Operation
64-67 (64-67)	kombinierte Statusinfo TAM
68-71 (68-71)	nicht benutzt
72-73 (72-73)	Zähler der Transaktionen innerhalb des Vorgangs
74 (74)	Nummer des UTM-Anwendungslaufs (beginnt nach der KDCDEF-Generierung mit 1)

DB-Record in der DB-DIAGAREA

Byte: 32-Bit (64-Bit)	Bedeutung
75 (75)	Kennzeichen für Tabellenindex, = ' T '
76-77 (76-77)	UTM-Tabellen-Index, der zum DB-Aufruf führte
78-79 (78-79)	Aktionsindex innerhalb dieser Tabelle
80-83 (80-87)	Vorgangszähler (eindeutig innerhalb des UTM-Anwendungslaufs)
84-87 (88-95)	intern benutzte Adresse
88-91 (96-103)	Rücksprungadresse im Teilprogramm. Die Adresse zeigt hinter den Aufruf CALL DB-System.
92 (104)	* als „eye catch“ zum Erkennen des Endes des Diagnose-Satzes

DB-Record in der DB-DIAGAREA

¹ Bei Operation Codes, die sich auf DB-Vorgänge beziehen, wird hier der Status des VGM angezeigt.

Ist das DB-System über die XA-Schnittstelle an openUTM gekoppelt und ein sogenannter „dynamischer xa-switch“ eingebunden (z.B. bei Oracle das Modul „oraswd“), dann wird beim An - bzw. Abmelden einer gemeinsamen Transaktion der folgende Diagnose-Satz geschrieben (32-Bit und 64-Bit stimmen überein):

Byte	Bedeutung
16-23	abdruckbar „ax_reg“ beim Anmelden der Transaktion abdruckbar „ax_unreg“ beim Amelden der Transaktion
24-27	Nummer der RM-Instanz
28-31	unbenutzt
32-47	abdruckbar der Returncode des Aufrufs
48	interner Status der Transaktion
49	interner Status der An - , bzw. Abmeldung
50 -51	unbenutzt
52-54	befindet sich der Aufruf innerhalb einer openUTM-Transaktion (abdruckbar „yes“ oder „no“)
55	unbenutzt
56	* als „eye catch“ zum Erkennen des Endes des Diagnose-Satzes

Diagnose-Satz

Status der DB-Transaktion (Inhalt von Byte 0 bis 3 bzw. 4 bis 7 des DB-Records:

Inhalt (sedez.)	Bedeutung des DB-Status
04	Die DB-Transaktion wurde in das vorläufige Transaktionsende (PTC) versetzt.
08	Innerhalb einer DB-Transaktion wurden Updates vorgenommen.
10	Die DB-Transaktion ist zurückgesetzt worden.
20	Die DB-Transaktion wurde geschlossen.
40	Der Teilprogrammlauf hat einen Aufruf CLOSE DB-Transaktion gegeben, die DB-Transaktion ist aus Sicht des Teilprogrammlaufs beendet.
80	Die DB-Transaktion ist offen.

Status der DB-Transaktion

Auch Kombinationen dieser Werte treten auf, z.B. „88“.

DB-Operationscodes (Byte 8 des DB-Records)

Inhalt (sedez.)	DB-Op.-Codes	Bedeutung
00	STPA	DB Connection Modul soll DB-spezifische Startparameter prüfen. Aufruf erfolgt beim Start eines UTM-Anwendungsprogramms.
04	CONC	Aufbau der Verbindung zwischen UTM-Task und Verbindungsmodul des DB-Systems. Aufruf erfolgt beim Start eines UTM- Anwendungsprogramms.
08	DCON	Abbau der Verbindung zwischen UTM-Task und dem DB-System. Aufruf erfolgt in der Endbehandlung eines UTM- Anwendungsprogramms.
10	USRC	DB-Aufruf des Teilprogramms ausführen (User Call).
14	FITA	DB-Transaktion normal beenden. Der Aufruf wird von openUTM am Ende einer gemeinsamen DB/DC Transaktion gegeben.
18	CATA	DB-Transaktion abnormal beenden. D.h. die DB-Transaktion wird auf den letzten Sicherungspunkt zurückgesetzt.
1C	BKTA	Der Aufruf wird bei PEND KP oder PEND PA/PR mit TASK-Wechsel (wegen TAC-Klassen) verwendet, die UTM-Task löst sich von der Bearbeitung einer DB/DC Transaktion.
20	COTA	Der Aufruf erfolgt dann, wenn eine Mehrschritt-Transaktion fortgesetzt wird, also nach vorherigem PEND KP oder PEND PA/PR mit TASK-Wechsel (wegen TAC-Klassen).
24	STAT	Status einer DB-Transaktion anzeigen oder Auftrag zum Löschen aller DB-Stati. Das DB-System führt Statusinformationen über DB-Transaktionen als Mittel zum koordinierten Wiederanlauf in openUTM und dem DB-System.
28	PETA	Vorläufiges Ende einer DB-Transaktion anfordern
2C	EDVG	Ende eines UTM-Vorgangs bei offenem DB-Vorgang.
30	BKVG	Unterbrechung eines offenen DB-Vorgangs.
34	COVG	Fortsetzung eines unterbrochenen DB-Vorgangs nach BKVG.
38	RSVG	openUTM zeigt dem DB-System einen UTM-Vorgangswiederanlauf an, falls auch ein DB-Vorgang offen ist.
3C	CNFPTC	Das DB-Verbindungsmodul bestätigt dem DB-System über die XA-Schnittstelle den Zustand der precommitted Transaktionen und löscht diese aus der internen Liste.
40	STRT	Das DB-Verbindungsmodul übermittelt dem DB-System über die XA-Schnittstelle den Beginn einer DB-Transaktion.
44	PEND	Das DB-Verbindungsmodul übermittelt dem DB-System über die XA-Schnittstelle das vorläufige Transaktionsende und leitet die erste Commitment-Phase ein.

DB-Operationscodes

DB-Error-Codes

Inhalt (sedez.)	Bedeutung des DB Error Codes
00	Auftrag ausgeführt.
04	Die DB-Transaktion musste zurückgesetzt werden. openUTM setzt dann auch die UTM-Transaktion zurück. Bei einem DB-Aufruf (User Call) bekommt das Anwendungs- teilprogramm wieder die Kontrolle.
08	Bewirkt bei openUTM Abbruch des Vorgangs mit PEND ER.
0C	Das DB-System ist nicht verfügbar, nicht hochgefahren.
10	Das DB-System ist aus einem anderen Grund nicht verfügbar.
14	Auftrag wurde nicht ausgeführt, später wieder versuchen!
18	Ein evtl. behebbarer Systemfehler liegt vor. openUTM erzeugt einen Task-Dump und versucht die offene DB Transaktion zu beenden.
1C	Ein nicht behebbarer Fehler liegt vor. Die Zusammenarbeit mit dem DB-System in dieser DB-Session erscheint nicht mehr sinnvoll.
20	Vom DB-System wurde ein Benutzerfehler entdeckt, z.B. bei der Prüfung der DB-Start- parameter.
24	Vom DB-System wurde ein Schnittstellenfehler entdeckt.

DB Error Codes

3.3.5.5 ADMI-DIAGAREA

Die ADMI-DIAGAREA ist ein Task-spezifischer Trace-Bereich in KDCROOT. Dieser wird, ebenso wie die UTM-DIAGAREA, zyklisch mit Trace-Records beschrieben. Bei jedem Aufruf der Programmschnittstelle der Administration wird ein Record in diesen Bereich geschrieben.

In den Bereich passen 71 (64-Bit: 62) Records, ein Record ist 112 (64-Bit: 128) Bytes lang. Der Bereich wird zyklisch beschrieben. Zwei Zyklen werden durch eine Trennlinie bestehend aus '='-Zeichen getrennt. Der Record oberhalb der Trennlinie ist der jüngste, der Record unterhalb der älteste.

Ein Record hat folgende Struktur:

Byte: 32-Bit (64-Bit)	Bedeutung
0-1 (0-1)	Zähler des aktuellen Eintrages in der ADMI-DIAGAREA
2-3 (2-3)	Zähler des zugehörigen Eintrages (Opcode „ADMI“) in der UTM-DIAGAREA
4-5 (4-5)	abdruckbares Kürzel für den Opcode
6-7 (6-7)	abdruckbares Kürzel für den Objekttyp bzw. den Subopcode1, je nach angegebenem Opcode
8-11 (8-15)	Adresse parameter area
12-15 (16-23)	Adresse identification area
16-19 (24-31)	Adresse selection area
20-23 (32-39)	Adresse data area
24-79 (40-95)	parameter area
80-111 (96-127)	Name des administrierten Objektes in der Objekttyp-spezifischen Länge (2-32) aus der identification area bzw. aus der data area

Aufbau der ADMI-DIAGAREA

Dabei werden der Eintragszähler und die Adressen der Bereiche vor Aufruf des UTM-Systemcodes mitprotokolliert. Die restlichen Daten werden nach der Rückkehr aus dem UTM-Systemcode, vor dem Rücksprung in das Teilprogramm mitprotokolliert. Daher sind

in der *parameter area* auch die Rückgabewerte - unter anderem der Returncode - enthalten. Der Inhalt der *identification area* wird nur dann mitprotokolliert, wenn der Bereich bei dem Administrationsaufruf verwendet wurde. Der Name aus der *data area* wird nur bei KC_CREATE_OBJECT mitprotokolliert.

Wird der Administrationsaufruf mit PEND ER beendet, weil die Adresse der *parameter area* nicht zugreifbar oder nicht auf Wortgrenze ausgerichtet ist, dann kann man diesem Protokoll entnehmen, welche Adresse angegeben wurde.

Es werden folgende Kürzel verwandt

Kürzel Opcode	Opcode	Kürzel Subopcode1/ Objektyp	Subopcode1 bzw. Objektyp
CA	KC_CHANGE_APPLICATION	N S O	KC_NEW KC_SAME KC_OLD
CD	KC_CREATE_DUMP		
CS	KC_CREATE_STATEMENTS		
EN	KC_ENCRYPT	V C D A N	KC_ACTIVATE_KEY KC_CREATE_KEY KC_DELETE_KEY KC_READ_ACTIV_PUBLIC_KEY KC_READ_NEW_PUBLIC_KEY
OI	KC_ONLINE_IMPORT	A	KC_ALL
PE	KC_PTC_TA	R	KC_ROLLBACK
LO	KC_LOCK_MGMT	UF US UA AB AA AP	KC_UNLOCK_USF KC_SIGNOFF_SINGLE KC_SIGNOFF_ALL KC_ABORT_BOUND_SERVICE KC_ABORT_ALL_BOUND_SERVICE KC_ABORT_PTC_SERVICE
UP	KC_UPDATE_IPADDR	A P	KC_ALL KC_PARTNER

Kürzel für Opcode und Subcode1 bzw. Objektyp

Kürzel Opcode	Opcode	Kürzel Subopcode1/ Objektyp	Subopcode1 bzw. Objektyp
GR	KC_CREATE_OBJECT		
DL	KC_DELETE_OBJECT		
GT	KC_GET_OBJECT		
MD	KC_MODIFY_OBJECT		
		AB	KC_ABSTRACT_SYNTAX
		AC	KC_ACCESS_POINT
		AP	KC_APPLICATION_CONTEXT
		BC	KC_BCAMAPPL
		CL	KC_TACCLASS
		CN	KC_CLUSTER_NODE
		CC	KC_CLUSTER_PAR
		CO	KC_CON
		CP	KC_CURR_PAR
		DA	KC_DIAG_AND_ACCOUNT_PAR
		DB	KC_DB_INFO
		DP	KC_DYN_PAR
		ED	KC_EDIT
		GB	KC_GSSB
		KS	KC_KSET
		LC	KC_LTAC
		LM	KC_LOAD_MODULE
		LP	KC_LPAP
		LS	KC_LSES
		LT	KC_LTERM
		MC	KC_MSG_DEST_PAR
		MM	KC_MESSAGE_MODULE
		MP	KC_MAX_PAR
		MX	KC_MUX
		OA	KC_OSI_ASSOCIATION
		OC	KC_OSI_CON
		OL	KC_OSI_LPAP
		PE	KC_PTC
		PO	KC_TPOOL

Kürzel für Opcode und Subcode1 bzw. Objektyp

Kürzel Opcode	Opcode	Kürzel Subopcode1/ Objektyp	Subopcode1 bzw. Objektyp
CR DL GT MD (Forts.)	KC_CREATE_OBJECT KC_DELETE_OBJECT KC_GET_OBJECT KC_MODIFY_OBJECT (Forts.)	PR PT QP QU SI SF SP TA TC TI TR UP US UF U1 U2	KC_PROGRAM KC_PTERM KC_QUEUE_PAR KC_QUEUE KC_SIGNON KC_SFUNC KC_SYSTEM_PAR KC_TASKS_PAR KC_TAC KC_TIMER_PAR KC_TRANSFER-SYNTAX KC_UTMD_PAR KC_USER KC_USER_FIX KC_USER_DYN1 KC_USER_DYN2
SH	KC_SHUTDOWN	K N W G	KC_KILL KC_NORMAL KC_WARN KC_GRACEFUL
SL	KC_SYSLOG	I CS SC SW WB	KC_INFO KC_CHANGE_SIZE KC_SWITCH_AND_CHANGE KC_SWITCH KC_WRITE_BUFFER
SM	KC_SEND_MESSAGE		
SP	KC_SPOOLOUT		
UL	KC_USLOG	SW	KC_SWITCH

Kürzel für Opcode und Subcode1 bzw. Objektyp

3.3.5.6 ADMI-USERAREA

Die ADMI-USERAREA ist ein Task-spezifischer Trace-Bereich in KDCROOT. Dieser Bereich dient dazu, die an der Programmschnittstelle vom Teilprogramm an openUTM übergebenen Daten mitzuprotokollieren.

Da die übergebenen Daten sehr umfangreich sein können, werden nur die Daten eines Aufrufs in dem Bereich abgelegt. Der Bereich besteht also nur aus einem Record und fasst 4140 Bytes. Der Bereich wird nur für die Aufrufe beschrieben, bei denen Daten an übergeben werden. Es wird der Inhalt der *data area* oder der Inhalt der *selection area* protokolliert, je nachdem, welcher Bereich bei dem Aufruf verwendet wurde. Wird der Bereich für die Diagnose benötigt, so muss darauf geachtet werden, dass der entsprechende Aufruf der letzte Aufruf ist, für den Daten mitprotokolliert werden, damit der Bereich nicht durch einen nachfolgenden Aufruf überschrieben wird.

Wird in der *data area* für einen Benutzer ein Passwort übergeben, dann wird dieses nicht mitprotokolliert, sondern mit binär Null überschrieben.

Ein Record hat folgende Struktur:

Byte 32-Bit (64-Bit)	Bedeutung
0-1 (0-1)	Zähler des zugehörigen Eintrages in der ADMI-DIAGAREA
2-3 (2-3)	irrelevant
4-59 (4-59)	parameter area
60-4139 (60-4139)	Inhalt data area bzw. selection area in der übergebenen Länge

Aufbau der ADMI-USERAREA

Der Inhalt der *parameter area* und die übergebenen Daten werden im UTM-Systemcode mitprotokolliert. Die *parameter area* wird so protokolliert, wie sie vom Teilprogramm übergeben wird, also ohne besetzte Rückgabewerte. Der Subreturncode ist auf Null gesetzt.

Der Eintragszähler wird nach der Rückkehr aus dem UTM-Systemcode vor dem Rücksprung in das Teilprogramm mitprotokolliert.

3.3.5.7 Der Kommunikationsbereich KB

Der Kommunikationsbereich besteht aus dem KB-Kopf, dem KB-Rückgabebereich und dem KB-Programmbereich in der generierten Länge.

Byte 32-Bit (64-Bit)	Feldnamen (COBOL und C/C++) und Inhalt	
0-7 (0-7)	KCBENID kcuserid	UTM-Benutzerkennung
8-15 (8-15)	KCTACVG kccv_tac	TAC, der verwendet wurde, um diesen Vorgang zu starten.
16-17 (16-17)	KCTAGVG: kccv_day	Tag
17-19 (17-19)	KCMONVG kccv_mon	Monat
20-21 (20-21)	KCJHRVG kccv_year	Jahr
22-24 (22-24)	KCTJHVG kccv_doy	Industrietag
25-26 (25-26)	KCSTDVG kccv_hour	Stunde
27-28 (27-28)	KCMINVG kccv_minute	Minute
29-30 (29-30)	KCSEKVG kccv_second	Sekunde
31 (31)	KCKNZVG kccv_status	Vorgangskennzeichen (mögliche Einträge siehe Anhang)
32-39 (32-39)	KCTACAL kopr_tac	TAC, mit dem das Programm adressiert wurde
40-41 (40-41)	KCTACAL kopr_hour	Stunde
42-43 (42-43)	KCMINAL kopr_minute	Minute
44-45 (44-45)	KCSEKAL kopr_second	Sekunde
46 (46)	KCAUSWEIS kccard	Ausweis-Kennzeichen: A (Ausweis steckt) oder Leerzeichen.
47 (47)	KCTAIND kctaind	Transaktionskennzeichen: F (erste) oder N (Folge-Transakt.)

KDCS-Kommunikationsbereich

Byte 32-Bit (64-Bit)	Feldnamen (COBOL und C/C++) und Inhalt	
48-55 (48-55)	KCLOGTER kclogter	Name des LTERM-Partners (Sender)
56-57 (56-57)	KCTERMN kctermn	Gerätetyp des Terminals oder Druckers, siehe Tabelle bei PTERM-Anweisung
58-59 (58-59)	KCLKBPB kclpa	Maximallänge des KB-Programmbereichs gemäß Generierung
60-61 (60-61)	KCHSTA: kchsta	Anzahl der gekellerten Vorgänge aus der Sicht des aktuellen Vorgangs.
62 (62)	KCDSTA kcdsta	Veränderung der Anzahl der gekellerten Vorgänge
63 (63)		leer
64 (64)	KCPRIND kcprind	Programmkennzeichen. "A" = Teilprogrammmlauf in einem Asynchron-Vorgang "D" = Teilprogrammmlauf in einem Dialog-Vorgang
65 (65)	KCOF1 kcof1	erlaubte OSI TP-Funktionen "B" = Basisfunktionen "H" = Basisfunktionen + Handshake-Funktionen "C" = Basis- und Commit-Funktionen mit Chained Transactions "O" = Other combination Leerzeichen, wenn Vorgang nicht über OSI TP gestartet wurde
66 (66)	KCCP kccp	UTM Client Protocol "0" = Asynchron-Vorgang "1" = LU6.1 "2" = OSI TP "3" = UPIC "4" = DTP "5" = APPLI "6" = SOCKET
67 (67)	KCTARB kctarb	TA ist mit Rollback gekennzeichnet.
68-71 (68-71)	KCYEARVG kccv_year	4-stellige Jahresangabe des Vorgangbeginns
73-83 (73-83)		leer (FILLER)
84-115 (84-115)	KCRFELD	KB-Rückgabebereich

KDCS-Kommunikationsbereich

Byte 32-Bit (64-Bit)	Feldnamen (COBOL und C/C++) und Inhalt		
84-85 (84-85)	KCRDF kcrdf KCRWVG kcrwvg	Rückgabe Bildschirmfunktion oder Anzahl wartender Vorgänge beim DGET	
86-87 (86-87)	KCRLM kcrIm	Rückgabe tatsächliche Länge der Nachricht	
	INFO CK-Aufruf	MGET-Aufruf	SIGN ON-Aufruf
88 (88)	KCRINFC/ kcrinfcc: KDCS-Returncode des geprüften KDCS-Aufrufs	KCVGST/ kcpcv_state: Vorgangstatus	KCRSIGN1/kcrsign1: Primärkode
89 (89)		KCTAST/ kcpta_state: Transaktionsstatus	KCRSIGN2: Sekundärkode
90-91 (90-91)		leer	
92-94 (92-94)	KCRCCC kcrccc	KDCS-Returncode	
95 (95)	KCRCKZ kcrckid	Kennzeichen: P (Produktivanwendung), T (Testanwendung)	
96-99 (96-99)	KCRCDC krcdc	interner Returncode	
100-107 (100-107)	KCRMF kcrfn	Rückgabe abstrakte Syntax	
108-115 (108-115)	KCRPI kcrpi KCRUS kcrus KCRQN kcrqn	Rückgabe Vorgangs-Identifikation Benutzerkennung beim SIGN ST oder Erzeuger der Nachricht beim DGET Von openUTM vergebener Name der Temporären Queue bei QCRE NN	
ab 116 (ab 116)	KB-Programmbereich		

KDCS-Kommunikationsbereich

3.3.6 Speicherbereiche in UTM-Cluster-Anwendungen

Die Cluster-globalen Speicherbereiche bestehen aus folgenden Tabellen:

UF-HDR	Header der Cluster-User-Datei
UF-ENT	Einträge der Cluster-User-Datei
JF-1-HDR	Header der ersten Datei des Cluster-Administrations-Journals
JF-1-ENT	Einträge der ersten Datei des Cluster-Administrations-Journals
JF-2-HDR	Header der zweiten Datei des Cluster-Administrations-Journals
JF-2-ENT	Einträge der zweiten Datei des Cluster-Administrations-Journals
BUF-SGMT	Segmente der Knoten-lokalen Pufferverwaltung
GF-HDR	Header der Cluster-GSSB-Datei
GF-ENT	Einträge der Cluster-GSSB-Datei
LF-HDR	Header der Cluster-Lock-Datei
LF-DLK	Bereich für Deadlock-Erkennung der Cluster-Lock-Datei
LF-ENT	Einträge der Cluster-Lock-Datei
CF-HDR	Header der Cluster-Konfigurations-Datei
CF-ENT	Einträge der Cluster-Konfigurations-Datei
UL-HDR	Header der Cluster-ULS-Datei
UL-ENT	Einträge der Cluster-ULS-Datei

3.3.7 Summary

Am Schluss des Dumps finden Sie eine Zusammenfassung, auch Summary-Information genannt, sowie ein Inhaltsverzeichnis und einen Meldungsabschnitt, der die Meldungen enthält, die während der Aufbereitung des UTM-Dumps ausgegeben wurden.

Die Summary-Information ist ein Extrakt der gesamten Dump-Information. Dabei sind solche Daten ausgewählt, die bei der Diagnose häufig benötigt werden. Das erspart Sucharbeit bei der Bearbeitung eines UTM-Dumps, vor allem bei der Vordiagnose und Duplikat-Erkennung.

Die Summary-Information enthält wichtige Generierungsparameter der Anwendung, Versionsnummern des Betriebssystems und von openUTM, Startparameter der Anwendung, die wichtigsten aktuellen Tabelleneinträge und die letzten Sätze in der DIAGAREA und TRACE-Area.

Wurde der Dump von KDCUPD erzeugt, wird nur die erste Seite der Summary geschrieben.

Es gibt UTM-Dumps, die nicht alle Tabellen enthalten, z.B. der PEND ER-Dump. Bei der Auswertung solcher Dumps kann die Summary deshalb einige Daten nicht zeigen.

Die Ausgabe der Summary-Information und des Dumps kann man über den Operanden INFO steuern.

4 UTM-Meldungswesen

openUTM erzeugt beim Ablauf einer UTM-Anwendung Meldungen, die über bestimmte Ereignisse informieren.

Eine **UTM-Meldung** besteht aus einer Meldungsnummer, einem festen **Meldungstext** und variablen Parametern, den so genannten Inserts. Diese **Inserts** werden dynamisch bei Ausgabe der jeweiligen Meldung mit den aktuellen Werten versorgt. Bei den Inserts handelt es sich z.B. um den Namen der Anwendung oder des Kommunikationspartners, um Zähler, Fehlercodes oder Ähnliches.

Jede Meldung wird über ihre **Meldungsnummer** identifiziert. UTM-Meldungsnummern beginnen immer mit dem Buchstaben K, P oder U gefolgt von einer 3-stelligen Nummer, z.B. K008.

Die UTM-Meldungen haben unterschiedliche Aufgaben und können an verschiedene Empfänger (**Meldungsziele**) gerichtet werden, wobei Sie selbst in einem gewissen Rahmen die Meldungsziele für K- und P-Meldungen Anwendungs-spezifisch festlegen können. U-Meldungen gehen dagegen immer nach *stderr* und *stdout*. Andere Meldungsziele sind in diesem Fall nicht möglich.

Nachfolgend einige Erläuterungen und Beispiele für UTM-Meldungen.

- Ein Benutzer kann am Terminal durch eine Meldung über ein bestimmtes Ereignis informiert und ggf. zu einer Eingabe aufgefordert werden.

Beispiel

Bei der Berechtigungsprüfung war das eingegebene Kennwort ungültig. openUTM fordert den Benutzer am Terminal auf, die KDCSIGN Eingabe zu wiederholen.

- Ein Ereignis innerhalb der UTM-Anwendung wird durch eine Meldung in der UTM-Protokolldatei SYSLOG protokolliert. So werden Daten zur Überwachung des Anwendungslaufes und für Diagnosezwecke gesammelt.

Beispiel

Eine Meldung informiert darüber, dass die Pagepool-Belegung in der KDCFILE einen bestimmten Pegel überschritten hat.

- Werden bestimmte UTM-Meldungen erzeugt, und Sie haben für diese Meldungen das Meldungsziel MSGTAC festgelegt, dann ruft openUTM ein ereignisgesteuertes Teilprogramm der Anwendung auf (siehe Abschnitt „Event-Service MSGTAC“ im openUTM-

Handbuch „Anwendungen programmieren mit KDCS“). Dieser Event-Service kann u.a. per FPUT asynchrone Aufrufe an die Administration absetzen. Man kann somit auf Ereignisse, bei denen openUTM-Meldungen erzeugt werden, mit programmierter Administration reagieren.

Beispiel

Beim Ereignis „Vorgangsabbruch“ wird die Meldung K017 bzw. K055 erzeugt. Der MSGTAC-Service kann darauf reagieren, indem er z.B. den TAC sperrt und eine Nachricht an den Administrator schickt.

Zur Ausgabe einer Meldung greift openUTM auf einen UTM-eigenen Meldungsmodul zu. Dieser enthält die Eigenschaften und Texte aller UTM-Meldungen. Bei Verwendung von NLS werden die Meldungstexte aus speziellen Meldungskatalogen entnommen, siehe [Abschnitt „NLS-Meldungskataloge“ auf Seite 126](#).

Sie können die Ausgabe von UTM-Meldungen in gewissem Rahmen Anwendungs-spezifisch gestalten. Sie können z.B. die Meldungsziele (Empfänger) von K- und P-Meldungen ändern und die Meldungstexte in eine andere Sprache übersetzen (nicht auf Windows-Systemen). Näheres ist dem [Abschnitt „Gestaltung der Meldungs Ausgabe durch den Anwender“ auf Seite 134](#) zu entnehmen.

Sie können auch NLS-Meldungskataloge zur Gestaltung der Meldungs Ausgabe verwenden. Die Meldungskataloge enthalten die Meldungstexte für eine bestimmte Sprache und einen Zeichensatz (Codeset). Daraus ergeben sich folgende Vorteile:

- Bei Verwendung von Meldungskatalogen erscheinen UTM-Meldungen beim User in der von ihm gewählten Sprache (sofern entsprechende NLS-Meldungskataloge existieren), d.h. eine Anwendung kann entsprechend den Erfordernissen mehrsprachig betrieben werden.
- Die Sprache eines Users muss nicht generiert werden, da die Meldungstexte zur Laufzeit automatisch in der richtigen Sprache ausgewählt werden. Hierzu wird die Umgebungsvariable %LANG% des Users vom zugehörigen Dialog-Terminalprozess ausgewertet.

Mit openUTM werden standardmäßig Meldungskataloge in Deutsch und Englisch ausgeliefert.

Sie können Inserts der Meldungen in den Meldungstext aufnehmen oder aus diesem entfernen.

4.1 Meldungsmodul, Meldungsdefinitionsdatei

Mit openUTM wird die Meldungsdefinitionsdatei *utmpfad/msgdescription* ausgeliefert sowie die beiden Standardmeldungsmodulare

kcsmsgs.o (K- und P-Meldungen)

kcxmsgs.o (U-Meldungen)

Diese Module sind unter Unix-Systemen in der Bibliothek *utmpfad/sys/libwork.a* bzw. *utmpfad/sys/libwork.so* enthalten.

Unter Windows-Systemen in der Bibliothek *utmpfad\sys\libwork.lib*.

Die Standardmeldungsmodulare enthalten die englischen Meldungstexte und die Standardeinstellungen für die Meldungsziele (z.B. Terminal, SYSLOG-Datei). Die Meldungsdefinitionsdatei dient als Eingabedatei zum Ändern der UTM-Meldungen durch den Anwender. Sie enthält die Meldungstexte in deutscher und englischer Sprache sowie die Rahmendefinitionen für die Meldungen (Strukturen der Meldungen).

Im folgenden Text wird für beide Standardmeldungsmodulare gemeinsam der Ausdruck „Standardmeldungsmodul“ verwendet.

Die Meldungsdefinitionsdatei kann mit Meldungstexten in weiteren Sprachen angereichert werden (nicht auf Windows-Systemen). Sie können die Meldungstexte übersetzen und sie mit dem Tool KDCMTXT (siehe [Seite 135ff](#)) in die Meldungsdefinitionsdatei eintragen. Aus der Meldungsdefinitionsdatei können Sie mit dem Tool KDCMMOD (siehe [Seite 141ff](#)) ein eigenes Meldungsmodul erstellen.



VORSICHT!

Sie dürfen die Meldungsdefinitionsdatei *msgdescription* nur mit den openUTM-Tools KDCMTXT und KDCMMOD bearbeiten! Andere Schreibzugriffe - z.B. mit einem Editor - **zerstören** diese Datei!

Bei der Ausgabe einer Meldung greift openUTM auf das UTM-Meldungsmodul zu. Dieses enthält für jede Meldung u.a.

- die Meldungsnummer Knnn, Pnnn bzw. Unnn
- den Meldungstext
- die Meldungsziele
- die Inserts

Im Standardmeldungsmodul ist eine bestimmte Art der Meldungsangabe vorgegeben. Wenn Sie diese Vorgaben ändern wollen, müssen Sie ein eigenes Meldungsmodul erzeugen. Sind keine NLS-Meldungskataloge für openUTM und kein benutzereigenes Meldungsmodul vorhanden, dann erzeugt openUTM die Meldungen aus dem Standardmeldungsmodul. Das Standardmeldungsmodul muss in **jedes** UTM-Anwendungsprogramm eingebunden werden; d.h. auch dann, wenn Sie NLS-Meldungskataloge oder ein eigenes Meldungsmodul verwenden.

4.2 NLS-Meldungskataloge

Mit NLS-Meldungskatalogen können Sie die Meldungstexte unabhängig vom eingebundenen Meldungsmodul in verschiedenen Sprachen ausgeben. Die Meldungstexte werden dann vom Programm zur Laufzeit Sprach-spezifisch aus dem entsprechenden Meldungskatalog entnommen. Die Sprache wird in Unix-/Windows-Systemen durch Auswertung der Umgebungsvariable `%LANG%` des Users festgelegt.

NLS-Meldungskataloge enthalten lediglich die Meldungstexte und keine Informationen über die Meldungsziele und Meldungsattribute. Zur Auswertung der Inserts, Meldungsziele und Meldungsattribute sowie im Fehlerfall verwendet openUTM den Standardmeldungsmodul bzw. einen benutzereigenen Meldungsmodul, sofern dieser existiert.

Eine UTM-Anwendung kann auch ohne NLS-Meldungskataloge betrieben werden. Aus diesen Gründen muss der UTM-Meldungsmodul (ausgeliefert in der Bibliothek `utmpfad/sys/libwork.*` bzw. unter Windows-Systemen in der Bibliothek `utmpfad\sys\libwork.lib`) in die Anwendung eingebunden werden. Zusätzlich kann wie bisher ein eigener, modifizierter Meldungsmodul eingebunden werden.

Die NLS-Meldungskataloge müssen weder bei der Generierung der Anwendung noch beim Binden des Anwendungsprogramms berücksichtigt werden.

Mit openUTM werden NLS-Standard-Meldungskataloge in deutscher und englischer Sprache ausgeliefert. Sie werden unter Unix-Systemen in den Dateiverzeichnissen `utmpfad/nls/msg/lang` abgelegt, wobei `lang` das Kennzeichen der jeweiligen Sprache ist. Unter Windows-Systemen werden sie in den Datenverzeichnissen `utmpfad\Nls\msg\lang` abgelegt.

Die Meldungskataloge unter `.../lang` haben folgende Namen:

Unix-Systeme	Windows-Systeme	
<code>utmsys.cat</code>	<code>utmsys.dll</code>	K- und P-Meldungen des Systems
<code>utmutil.cat</code>	<code>utmutil.dll</code>	K-Meldungen der Dienstprogramme
<code>utmxprog.cat</code>	<code>utmxprog.dll</code>	U-Meldungen

Anwendungs-spezifische Meldungskataloge, d. h. Meldungskataloge unter `filebase`, werden derzeit nur für Meldungen der Funktionseinheit SYS (Meldungskatalogname `utmsys.cat` bzw. `utmsys.dll`) ausgewertet.

4.2.1 Meldungskatalog-Sourcedatei für NLS

Anwendungs-spezifische Meldungskataloge können mit Hilfe der Tools KDCMTEXT bzw. KDCMMOD erzeugt werden (siehe [Seite 135ff](#) bzw. [Seite 141ff](#)). KDCMTEXT legt die Meldungstexte für jede Sprache und Funktionseinheit in einer Sourcedatei *FU_LN.msg* ab, wobei *FU* die ausgewählte Funktionseinheit und *LN* das in der FU-Anweisung verwendete Sprachkennzeichen ist.

Die spezifischen, d. h. vom Betreiber einer Anwendung geänderten Meldungskataloge müssen für die Funktionseinheit SYS unter

utmpfad/nls/msg/lang

mit dem Namen *utmsys.cat* abgelegt werden, wobei *lang* wie für die Standard-Meldungskataloge eine Sprachbezeichnung ist.

Die Meldungstexte der NLS-Sourcedateien enthalten für die Position der Inserts „Pseudo-printf-Kontrollstrings“ *%nn\$*, wie z. B. *%05\$*. Dabei ist die zweistellige Zahl *nn* die Nummer des Inserts der zugehörigen Meldung in der Meldungsdefinition (Rahmendefinition), hier also das fünfte Insert der Meldung.

4.3 Meldungsziele

openUTM erzeugt während einer laufenden Anwendung Meldungen. K- und P-Meldungen gehen an eines oder mehrere der folgenden Meldungsziele:

STDOUT	Ausgabe auf <i>stdout</i> .
STDERR	Ausgabe auf <i>stderr</i> .
STATION	Clients, die über ein PTERM oder einen TPOOL mit PTTYPE=TTY angeschlossen sind.
SYSLINE	Systemzeile des Terminals, dabei bleibt der Bildschirminhalt erhalten. Soll eine Meldung in der Systemzeile erscheinen, so ist SYSLINE und STATION als Meldungsziel anzugeben.
CONSOLE	Konsole des Systemoperators, der Anwendungsname wird mit ausgegeben.
PARTNER	Clients, die über ein PTERM oder einen TPOOL mit PTTYPE=APPLI oder SOCKET angeschlossen sind.
SYSLOG	System-Protokolldatei SYSLOG (siehe Seite 153).
MSGTAC	MSGTAC-Service (siehe Steueranweisung TAC im openUTM-Handbuch „Anwendungen generieren“ sowie Abschnitt „Event-Service MSGTAC“ im openUTM-Handbuch „Anwendungen programmieren mit KDCS“).
USER-DEST-1 bis USER-DEST-4	Benutzer-spezifische Meldungsziele, denen Sie bei der Generierung als konkrete Meldungsziele eine USER-Queue, eine TAC-Queue, einen Asynchron-TAC oder einen LTERM-Partner zuordnen können. WinAdmin und WebAdmin können aus diesen UTM-Queues, zyklisch oder auf Anforderung des Anwenders, Meldungen abholen, diese in Listen ausgeben und gegebenenfalls in der Konfigurationsdatenbank abspeichern. Die Zuordnung zwischen USER-DEST-1..4 und dem konkreten Ziel wird über die KDCDEF-Anweisung MSG-DEST festgelegt.

U-Meldungen gehen nur nach *stdout* und *stderr*. Meldungsziele können deshalb nicht geändert werden.

4.3.1 Ausgabeform der Meldungen

Je nach Meldungsziel gibt openUTM die Meldungen in unterschiedlicher Form aus:

- an SYSLOG und MSGTAC:
den Meldungskopf und die aktuellen Werte der Parameter, wie im [Abschnitt „Aufbau der UTM-Systemmeldungen“ auf Seite 156](#) beschrieben,
- an CONSOLE:
den Anwendungsnamen, die Meldungsnummer und den Meldungstext mit den Textparametern,
- an USER-DEST-1...USER-DEST-4:
 - den Meldungskopf und die aktuellen Werte der Inserts, wie in [Abschnitt „Aufbau der UTM-Systemmeldungen“ auf Seite 156](#) beschrieben, falls USER-DEST-*n* mit MSG-FORMAT=FILE generiert ist,
 - Datum/Uhrzeit gefolgt von Meldungsnummer, Meldungstext und Inserts, falls USER-DEST-*n* mit MSG-FORMAT=PRINT generiert ist,
- an alle anderen Ziele:
die Meldungsnummer und den Meldungstext mit Textparametern, so weit sie im Meldungstext enthalten sind.

In bestimmten Fehlersituationen im Lauf der Anwendung (abnormale Beendigung der Anwendung, abnormale Programm- oder Prozessbeendigung) ist es möglich, dass openUTM eine Meldung nur an die Ziele STDERR, STDOUT oder CONSOLE ausgibt, auch wenn die Meldung für weitere Ziele (z.B. SYSLOG) bestimmt ist. Dies geschieht, um eventuelle Folgefehler zu verhindern.

Aus dem gleichen Grund entnimmt openUTM in bestimmten Fehlersituationen den Meldungstext auch nicht dem Meldungsmodul, den der Anwender verändern kann, sondern dem Standardmeldungsmodul.

Header für Meldungen auf STDERR / STDOUT

Bei der Ausgabe von Meldungen an die Meldungsziele STDERR und STDOUT stellt openUTM den Meldungen standardmäßig einen Header mit Datum und Uhrzeit voran.

Dieser Header enthält auch die PID des Prozesses, der die Meldung erzeugt hat. Mit diesen Informationen lassen sich z.B. Fehlersituationen leichter diagnostizieren.

Dieser Header hat folgende Form:

```
pid jjjj-mm-tt hh:mm:ss
```

Die Meldungen K038 und K044 werden immer ohne Datum und Uhrzeit ausgegeben. Mit der Umgebungsvariable `UTM_MSG_DATE=NO` lässt sich die Ausgabe von Datum und Uhrzeit auch für alle anderen Meldungen unterdrücken.

Mit der Umgebungsvariable `UTM_MSG_PID=NO` lässt sich die Ausgabe der PID unterdrücken.

4.3.2 UTM-Meldungen an die Konsole

In Meldungen, die auf Konsole (Meldungsziel `CONSOLE`) ausgegeben werden, trägt `openUTM` zusätzlich zum Meldungstext noch den Anwendungsnamen ein.

Der Systemverwalter muss dabei unter Unix-Systemen sicherstellen, dass die entsprechende Benutzerkennung (`USER`) tatsächlich auf die Konsole schreiben darf. Im Bedarfsfall muss der Systemverwalter die Zugriffsrechte für die Konsole ändern (Kommando `chmod` auf `/dev/console`).

Unter Windows-Systemen wird im Verzeichnis *filebase* eine Datei `console.txt` angelegt, in die Meldungen mit dem Ziel `CONSOLE` geschrieben werden.

4.3.3 UTM-Meldungen an eine TS-Anwendung

Sind UTM-Meldungen für eine Transportsystem-Anwendung vom Typ `PTYPE=APPLI` oder `SOCKET` bestimmt (Meldungsziel = `PARTNER`), so ist darauf zu achten, dass diese Anwendung die Meldungen erkennt und sinnvoll darauf reagiert. Geschieht dies nicht, kann es z.B. bei der Kopplung zweier UTM-Anwendungen als TS-Anwendungen dazu kommen, dass die Anwendungen sich gegenseitig endlos Meldungen übermitteln wie
K009 Der Transaktionscode K009 ist ungueltig.

USP-Header bei UTM-Meldungen an eine Socket-Anwendung

Für UTM-Meldungen an eine Socket-Anwendung (`PTYPE=SOCKET`, Meldungsziel = `PARTNER`) können Sie bei der Generierung festlegen, ob `openUTM` der Meldung einen USP-Header (`openUTM-Socket-Protokoll-Header`) voranstellen soll. Der USP-Header dient u.a. dazu, dem Socket-Partner die Länge der empfangenen Nachrichten auszugeben.

Dazu legen Sie bei der `KDCDEF`-Generierung im Operanden `USP-HDR=` der `PTERM-` oder `TPOOL-`Anweisung folgendes fest:

```
USP-HDR = MSG oder USP-HDR = ALL
```

Falls Sie `USP-HDR = NO` (Standardeinstellung) generieren, so wird kein USP-Header erzeugt.

Eine Beschreibung des USP-Headers finden Sie im `openUTM`-Handbuch „Anwendungen programmieren mit `KDCS`“.

4.3.4 UTM-Meldungen an Benutzer-spezifische Meldungsziele

Bei Auftreten einer Meldung, für die als Meldungsziel USER-DEST-1 ...USER-DEST-4 vereinbart wurde, erzeugt UTM intern einen Asynchron-Auftrag an dieses Meldungsziel. Diesem Asynchron-Auftrag wird als Verursacher der Benutzer KDCMSGUS und der LTERM-Partner KDCMSGLT zugeordnet. Wird der Asynchron-Auftrag zurückgewiesen, z.B. weil das Meldungsziel gesperrt ist (STATUS=OFF), geht die Meldung für das Meldungsziel verloren. Wird wieder eine Meldung für dieses Meldungsziel erzeugt, versucht UTM erneut, einen Asynchron-Auftrag zu erzeugen.

Ist als Meldungsziel ein Asynchron-TAC generiert, dann startet UTM das dem TAC zugeordnete Programm jedes Mal, wenn die betreffende Meldung erzeugt wurde (in einem Programmablauf kann immer nur eine Meldung mit FGET gelesen werden).

4.3.5 UTM-Meldungen an MSGTAC

MSGTAC ist ein spezielles Asynchron-Teilprogramm, das der Anwender selbst programmieren kann, siehe openUTM-Handbuch „Anwendungen programmieren mit KDCS“.

Ist ein MSGTAC-Programm vorhanden und tritt eine Meldung auf, für die das Meldungsziel MSGTAC vereinbart wurde, dann wird der Asynchron-Vorgang MSGTAC gestartet. MSGTAC liest die Meldung und kann entsprechend darauf reagieren. Das MSGTAC-Programm kann in einem Teilprogrammablauf mehrere Meldungen lesen.

Das MSGTAC-Teilprogramm läuft unter der internen UTM-Benutzerkennung KDCMSGUS mit KSET=MASTER und PERMIT=ADMIN. Es muss in einer TAC-Anweisung definiert werden mit `TAC KDCMSGTC,PROGRAM=...`

4.4 Aufbereitung der Meldungen durch openUTM

Initialisierung

Damit U-, K- und P-Meldungen in der gleichen Sprache ausgegeben werden, initialisiert openUTM bzw. der Dialog-Terminalprozess zunächst die Standardmeldungskataloge. Anwendungs-spezifische Meldungskataloge einer Sprache werden nur dann verwendet, wenn für die entsprechende Sprache auch die Standardmeldungskataloge vorhanden sind und lesender Zugriff erlaubt ist:

- `utmsys.cat` (Unix-System) bzw. `utmsys.dll` (Windows-System)
- `utmxprog.cat` (Unix-System) bzw. `utmxprog.dll` (Windows-System)

Existiert für eine Sprache z.B. nur der Standardmeldungskatalog für die K-Meldungen (`utmsys.cat` bzw. `utmsys.dll`), dann wird die NLS-Meldungsbearbeitung nicht initialisiert.

Die eingebundenen Meldungsmodule werden in folgenden Situationen verwendet:

- wenn Sie keine eigenen NLS-Kataloge einsetzen
- zur Auswahl der Meldungsziele
- als Rückfallstufe im Fehlerfall

Existiert für die `$LANG`-Variable eines Users kein Meldungskatalog, dann wird wie bisher auf den in der Anwendung eingebundenen Meldungsmodul zurückgegriffen.

Auswertung der Meldungstexte unter Unix-Systemen

Sofern Meldungskataloge verwendet werden, sucht eine UTM-Anwendung nach dem Start in folgender Reihenfolge nach den Meldungstexten:

1. Zunächst sucht eine Anwendung im Verzeichnis `filebase/nls/msg/lang` bzw. nach einem, eventuell vom Anwendungsbetreiber, modifizierten NLS-Meldungskatalog mit dem Namen `utmsys.cat`, wobei `lang` dem Inhalt der `$LANG`-Variable des betreffenden Users entspricht. Falls unter `filebase/nls/msg` kein Verzeichnis `lang` vorhanden ist, wird `lang` auf den Sprachbestandteil der Variablen `$LANG` verkürzt und der so entstandene Name als aktueller Katalogname verwendet. So wird z.B. `En_US.ASCII` auf `En` verkürzt, wenn kein Verzeichnis `.../En_US.ASCII` vorhanden ist.
2. Existiert im Verzeichnis `filebase/nls/msg/lang` kein Meldungskatalog, dann sucht die Anwendung im UTM-Verzeichnis `utmpfad` unter `nls/msg/lang` einen Meldungskatalog. Im Fehlerfall wird der Wert von `lang` wie oben auf den Sprachbestandteil reduziert. In diesem Verzeichnis befinden sich NLS-Meldungskataloge, die für alle Anwendungen auf dem Rechner gemeinsam benützt werden, d.h. wenn ein Meldungskatalog unter diesem Verzeichnis modifiziert wird, so gilt diese Änderung für alle Anwendungen auf dem Rechner.

3. Kann die Anwendung auch unter *utmpfad/nls/msg/lang* keinen Meldungskatalog finden, so greift die UTM-Anwendung auf den eingebundenen Standardmeldungsmodul bzw. einen zusätzlich eingebundenen Anwendungs-spezifischen Meldungsmodul zurück.

Auswertung der Meldungstexte unter Windows-Systemen

In Windows-Systemen können nur Meldungsziele geändert werden (mit dem Tool KDCMMOD). Änderungen an den Meldungstexten sind nicht möglich, d.h. es können nur die ausgelieferten Meldungskataloge für deutsche und englische Meldungen verwendet werden. Diese Meldungskataloge werden als DLLs ausgeliefert und befinden sich in den Dateiverzeichnissen *%UTMPATH%\nls\msg\lang*. Dabei ist *lang* der Wert der Umgebungsvariable *%LANG%*.

Für *%LANG%* können Sie die Werte „De“ für Deutsch und „En“ für Englisch setzen. Andere Werte werden wie „En“ behandelt.

openUTM lädt daher beim Start entweder den deutschen oder den englischen Meldungskatalog, abhängig davon, welcher Wert für *%LANG%* gesetzt ist.

Aufbereitung der Meldungen

Liegen keine NLS-Kataloge vor, oder besteht keine Übereinstimmung zwischen dem Wert der Umgebungsvariablen *%LANG%* und den existierenden NLS-Katalogen, so wird das eingebundene Standardmeldungsmodul verwendet. Da der Dialog-Terminalprozess die Meldungen des Transaktionsmonitors selbst aufbereitet, hat jeder Anwender sein „eigenes Meldungswesen“. Hat z.B. ein Anwender *LANG=Französisch* eingetragen und gibt es im System nur deutsche und englische, aber keine französischen NLS-Kataloge, so werden für diesen Anwender die Standard-Meldungen des eingebundenen Meldungsmodul ausgegeben. Für alle anderen Anwender werden die Meldungen gemäß den Einträgen in den NLS-Katalogen aufbereitet.

Ausnahmen bei der Meldungs Ausgabe

openUTM nimmt bei der Ausgabe von UTM-Meldungen im laufenden Betrieb den Meldungstext und die aktuellen Meldungsziele aus dem Standardmeldungsmodul. Davon gibt es jedoch folgende Ausnahmen:

- Für einige Meldungen, die von ROOT-Modulen ausgegeben werden, sind englische Meldungstexte einprogrammiert, z.B. für K078. Für diese Meldungen kann die Ausgabe nicht verändert werden.
- In der Start- und Ende-Behandlung der Prozesse kann openUTM aus technischen Gründen nicht auf den Meldungsmodul der Anwendung zugreifen. openUTM nimmt in diesen Fällen den Meldungstext und die Meldungsziele aus dem deutschen Meldungsmodul.

4.5 Gestaltung der Meldungsausgabe durch den Anwender

In der Meldungsdefinitionsdatei ist festgelegt, wie openUTM standardmäßig Meldungen ausgibt, d.h. mit welchen Texten, an welche Ziele, usw. sowie die Einschränkungen für deren Änderbarkeit. Wenn Sie diese Standardeigenschaften ändern wollen, müssen Sie ein eigenes Meldungsmodul erzeugen und zu der UTM-Anwendung binden. Sie können die K- und P-Meldungen der Work-Prozesse und die U-Meldungen ändern. Folgende Änderungen sind möglich:

- Hinzufügen oder Weglassen von Meldungszielen (mit KDCMMOD) wie z.B. Protokollieren weiterer Meldungen nach STDERR oder Eintragen des Ziels MSGTAC als Voraussetzung für den Einsatz eines MSGTAC-Service.
- Meldungstexte in eine andere Sprache übersetzen und mit dem Tool KDCMTEXT in die Meldungsdefinitionsdatei einbringen ([Seite 135](#)).
- Ändern von Meldungstexten mit dem Tool KDCMMOD ([Seite 141](#)) wie z.B. Hinzufügen oder Weglassen von (erlaubten) Inserts in einem Meldungstext, Texte in Klein-/Groß-Buchstaben umsetzen usw.

Hinweis

In Windows-Systemen können nur Meldungsziele verändert werden. Änderungen an Meldungstexten sind nicht möglich ([Seite 133](#)).

- Definieren von Textkonstanten. Diese können in vielen Meldungstexten verwendet werden. Hierzu gehören auch Steuerzeichen.

Die individuelle Gestaltung der Meldungsausgabe gilt jeweils nur für die entsprechende UTM-Anwendung. Sie hat keine Auswirkung auf andere UTM-Anwendungen auf dem selben Rechner und erfordert keine Änderungen im Betriebssystem.

Die Modifikationen dürfen Sie nur mit Hilfe der Tools KDCMTEXT und KDCMMOD durchführen.

Bei Übersetzung aller Meldungstexte in eine andere Sprache sollte das Tool KDCMTEXT verwendet werden. Wenn nur wenige Texte eines vorhandenen Meldungskataloges geändert werden, kann auch das Tool KDCMMOD verwendet werden.

Erzeugen eines eigenen Meldungsmoduls

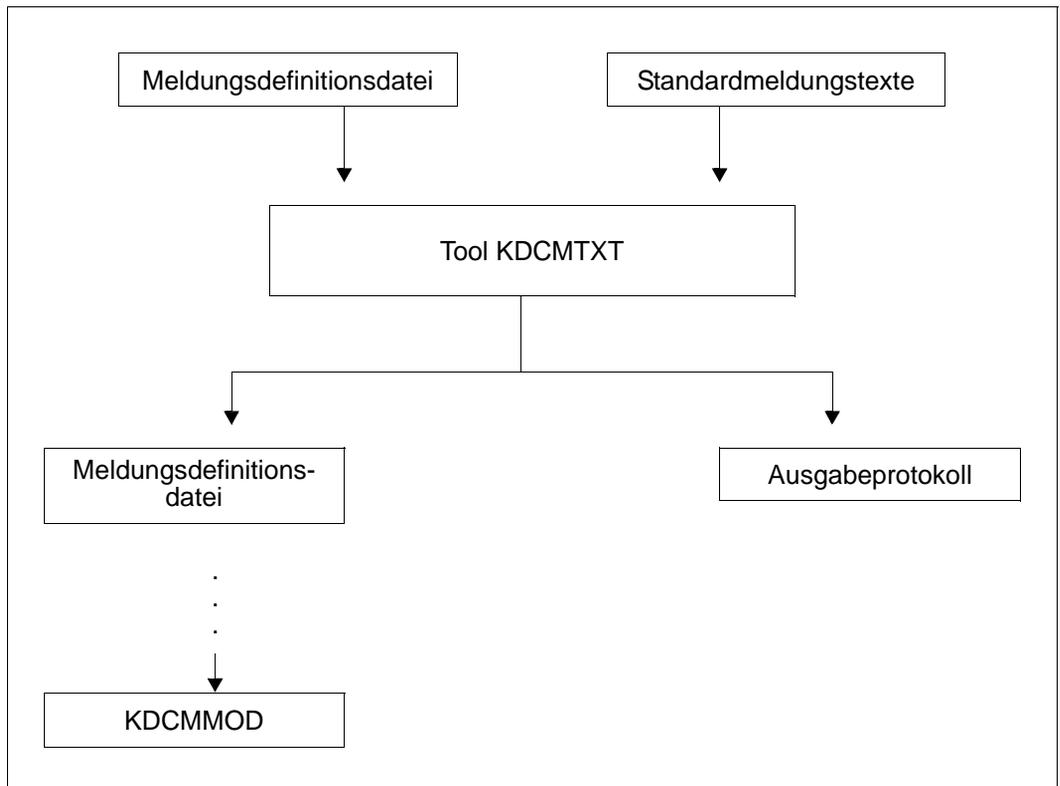
Ein eigenes Meldungsmodul muss mit dem Tool KDCMMOD erzeugt werden, siehe [Seite 141](#). KDCMMOD erstellt aus der Meldungsdefinitionsdatei und einer vom Anwender verfassten Änderungsbeschreibung ein C-Quellprogramm sowie eine NLS-Sourcedatei. Durch Übersetzung dieses C-Programms erhalten Sie das Anwendungs-spezifische Meldungsmodul, das statt des Standardmeldungsmoduls mit den Teilprogrammen einer Anwendung zusammen gebunden wird.

4.5.1 Meldungen in anderen Sprachen - Tool KDCMTXT

Standardmäßig werden englische Meldungstexte ausgegeben (deutsche Meldungstexte werden mit KDCMMOD erzeugt). Das Programm KDCMTXT ist ein Werkzeug, um die Meldungsdefinitionsdatei `utmpfad/msgdescription` mit Meldungstexten in weiteren Sprachen zu ergänzen.

KDCMTXT benötigen Sie, wenn die Standardmeldungstexte in einer anderen Sprache als Deutsch oder Englisch ausgegeben werden sollen. Mit KDCMTXT fügen Sie Ihre Übersetzungen in die Meldungsdefinitionsdatei ein und erzeugen eine NLS-Sourcedatei, aus der Sie mit `genclat` einen NLS-Meldungskatalog erzeugen können. Aus der erweiterten Meldungsdefinitionsdatei können Sie mit KDCMMOD ein C-Quellprogramm für einen Meldungsmodul erzeugen.

Das folgende Bild veranschaulicht die Eingaben und Ausgaben von KDCMTXT.



Meldungen in anderen Sprachen erstellen mit KDCMTXT

4.5.1.1 Aufruf von KDCMTXT

Das Tool KDCMTXT rufen Sie auf mit:

utmpfad/ex/kdcmtxt

4.5.1.2 KDCMTXT-Steueranweisungen

KDCMTXT kennt folgende Steueranweisungen:

OPTION	Namen der Meldungsdefinitionsdatei angeben
FU	Funktionseinheit und Landessprache definieren
MSGBASE	Auswahl der Meldungsgruppe bei der FU SYS: UTM (K-Meldungen) oder XAPTP (P-Meldungen)
STDTXT	Anfang des Standardmeldungstextes
ENDTXT	Ende des Standardmeldungstextes
END	Ende der Eingabe von Steueranweisungen

Bei den Eingaben der Anweisungen gelten folgende Regeln:

- Kommentarzeilen werden durch einen Stern (*) in der 1. Spalte gekennzeichnet
- Endet eine Zeile mit einem Komma, dann interpretiert KDCMTXT die folgende Zeile als Fortsetzungszeile der Anweisung.

Zweckmäßigerweise schreiben Sie die Steueranweisungen für KDCMTXT in eine Datei, die Sie beim Aufruf des Tools als Standardeingabe zuweisen. Mit openUTM wird eine Datei *utmpfad/mtxtin* ausgeliefert. Diese Datei enthält die deutschen und englischen Standardmeldungstexte in der Syntax der KDCMTXT-Steueranweisungen.

Die Datei *mtxtin* kann Ihnen als Vorlage für eine eigene Eingabedatei dienen.

Hinweis

Bei der Definition von Meldungstexten darf ein für die jeweilige Meldung zugelassenes Insert nur einmal vorkommen.

Anweisung OPTION

Die Anweisung OPTION muss, wenn sie angegeben wird, die erste Steueranweisung sein. Mit ihr kann man den Namen der Meldungsdefinitionsdatei angeben, die durch KDCMTXT geändert werden soll.

Operation	Operanden
OPTION	MSGFILE=filename

filename ist der Name der Meldungsdefinitionsdatei. Er darf bei vollqualifizierter Angabe maximal 54 Zeichen und bei teilqualifizierter Angabe maximal 14 Zeichen lang sein.

Wenn Sie keine OPTION-Anweisung angeben, verwendet KDCMTXT die standardmäßig mit openUTM ausgelieferte Meldungsdefinitionsdatei *utmpfad/msgdescription*.

Anweisung FU

Mit dieser Anweisung kann man Funktionseinheit und Landessprache definieren.

Operation	Operanden
FU	[funktionseinheit][,][LANG=sprachkennzeichen]

funktionseinheit

Funktionseinheit, für die die neue Landessprache eingetragen werden soll. Es sind folgende Werte erlaubt:

SYS: K-Meldungen (K001-K399) bzw. P-Meldungen (P001 - P049)

XPROG: U-Meldungen (U101-U550)

Standard: SYS

LANG=sprachkennzeichen

Das Sprachkennzeichen ist maximal 3 Zeichen lang und kennzeichnet die Landessprache. Das Sprachkennzeichen kann frei vergeben werden, es muss jedoch innerhalb einer Meldungsdefinitionsdatei eindeutig sein.

Als Sprachkennzeichen können z.B. die Kfz-Länderkennzeichen oder die in ISO IS/R639 festgelegte Sprachkennzeichen verwendet werden.

KFZ-Kennz.	ISO IS/R639	Sprache
DK	Da	Dänisch
D	De	Deutsch
GB	En	Englisch
E	Es	Spanisch
FI	Fi	Finnisch
F	Fr	Französisch
GR	Gr	Griechisch
I	It	Italienisch
NL	Nl	Niederländisch
N	No	Norwegisch
P	Pt	Portugiesisch
S	Sv	Schwedisch

Standard: GB

Das Komma muss nur dann angegeben werden, wenn sowohl *funktionseinheit* als auch `LANG=sprachkennzeichen` angegeben werden., z.B. `FU XPROG,LANG=DK`.

Für eine Funktionseinheit müssen alle Meldungstexte für eine Landessprache definiert werden, damit die Meldungstexte für diese Sprache und Funktionseinheit in die Meldungsdefinitionsdatei übernommen werden.

Nach Abarbeitung der Anweisung FU wird implizit die Anweisung MSGBASE UTM ausgeführt.



VORSICHT!

Die ausgelieferte Meldungsdefinitionsdatei enthält bereits die deutschen und die englischen Meldungstexte mit Sprachkennzeichen D bzw. GB! Geben Sie eines dieser Kennzeichen an, überschreibt KDCMTXT die entsprechenden Standardmeldungstexte in der Meldungsdefinitionsdatei mit den neuen Meldungstexten.

Anweisung MSGBASE (nur bei FU SYS)

openUTM unterscheidet innerhalb der FU SYS zwei Meldungsgruppen: Meldungen der Gruppe UTM, die mit dem Buchstaben K beginnen, und Meldungen der Gruppe XAPTP, die mit dem Buchstaben P beginnen.

Mit der Anweisung MSGBASE können Sie die Meldungsgruppe auswählen, auf die sich die nachfolgenden STDTXT-Anweisungen bis zur nächsten MSGBASE- bzw. FU-Anweisung beziehen.

Operation	Operanden
MSGBASE	{ UTM XAPTP }

UTM Auswahl der Meldungsgruppe UTM (K-Meldungen).

XAPTP Auswahl der Meldungsgruppe XAPTP (P-Meldungen).

Anweisungen STDTXT und ENDTXT

Die Steueranweisung STDTXT leitet die Definition des Standardtextes für eine Meldung ein. Die nachfolgenden Eingabezeilen beschreiben den Standardtext.

Die ENDTXT-Anweisung schließt die Definition des Meldungstextes ab.

Operation	Operanden
STDTXT	<i>msg-nr</i> text
ENDTXT	

msg-nr Meldungsnummer, für die der Standardtext bestimmt ist.

msg-nr gibt an, welche K- bzw. P-Meldung modifiziert werden soll. Die Meldungsnummer muss im Meldungsnummernbereich der spezifizierten Funktionseinheit liegen. Die Meldungsnummer ist dreistellig und **ohne** das vorangestellte **K** bzw. **P** anzugeben.

Pflichtoperand.

text Hier ist der Meldungstext anzugeben, die Syntax ist dieselbe wie beim Programm KDCMMOD (siehe [Seite 148](#)); ebenso gelten die gleichen Längenbeschränkungen.

Die Steueranweisungen STDTXT und ENDTXT müssen jeweils in einer neuen Zeile beginnen. Die STDTXT / ENDTXT-Anweisungen für eine Funktionseinheit und eine Sprache müssen alle hinter der FU-Anweisung für diese Funktionseinheit stehen und vor der nächsten FU-Anweisung.

KDCMTXT übernimmt die Meldungstexte für eine weitere Sprache nur dann in die Meldungsdefinitionsdatei, wenn die Eingabedaten Meldungstexte für alle Meldungen dieser Sprache enthalten, und wenn alle Eingaben fehlerfrei sind.

Sind die Meldungstexte einer Sprache bereits in der Meldungsdefinitionsdatei vorhanden, dann übernimmt KDCMTXT auch einzelne Meldungen. Die NLS-Sourcedatei enthält in diesem Fall nur die geänderten Meldungen.

Hinweis

Zum Ändern von Meldungstexten verwenden Sie das Tool KDCMMOD.

Anweisung END

Diese Anweisung beendet die Eingabe von Steueranweisungen.

4.5.1.3 Protokoll von KDCMTXT

Nachdem das Tool KDCMTXT alle Änderungen bearbeitet und eine neue Meldungsdefinitionsdatei erstellt hat, gibt es als Protokoll eine aufbereitete Liste aller Meldungen nach *stdout* und *stderr* aus (Ausgabeprotokoll). In der Liste werden die Meldungstexte in der Form aufbereitet, in der sie an eine Datensichtstation ausgegeben würden. Dabei werden die Inserts je nach Typ in der entsprechenden Länge wie folgt gefüllt:

Typ	Füllzeichen
CHAR	@
INT	#
HEXA	X

4.5.2 Erstellen eines eigenen Meldungsmoduls mit KDCMMOD

Wollen Sie das ausgelieferte Standardmeldungsmodul mit englischen Meldungstexten nicht verwenden, dann bietet openUTM Ihnen die Möglichkeit, ein eigenes Meldungsmodul zu erzeugen. Damit können Sie, in gewissen Grenzen, das Standardmeldungsmodul ändern, indem Sie nur die Veränderungen gegenüber dem Standard beschreiben.

Möglich sind Änderungen für die K-Meldungen K001 - K149, die P- und die U-Meldungen.

Sie können bei Verwendung eines eigenen Meldungsmoduls für K- und P-Meldungen

- die Ziele einzelner Meldungen ändern
- Meldungen auf die Konsole ausgeben
- eine MSGTAC-Routine einsetzen
- eigene Meldungstexte ausgeben
- die Meldungstexte verändern, z.B. in eine andere Sprache übersetzen.

Hinweis

In Windows-Systemen können nur Meldungsziele verändert werden. Änderungen an Meldungstexten sind nicht möglich ([Seite 133](#)).

Für die U-Meldungen können Sie

- eigene Meldungstexte ausgeben
- die Meldungstexte verändern, z.B. in eine andere Sprache übersetzen.

Mit dem Tool KDCMMOD können Sie auch eigene Meldungstexte erstellen, sowie Inserts und Meldungsziele verändern. Jedes zugelassene Insert darf allerdings nur einmal in der Meldung vorkommen. KDCMMOD erstellt aus der Meldungsdefinitionsdatei und Ihren Angaben eine C-Quelldatei und eine NLS-Sourcdatei, mit den geänderten Texten für den entsprechenden NLS-Katalog. Durch Übersetzung der C-Quelldatei erstellen Sie einen Anwender-spezifischen neuen Meldungsmodul. Dieser kann dann zusätzlich zum Standardmeldungsmodul mit dem Anwendungsprogramm gebunden und verwendet werden.

Das Tool KDCMMOD legt das erzeugte C-Quellprogramm in der Datei *modul-name.c* im aktuellen Dateiverzeichnis ab, wobei *modul-name* der in der GEN-Anweisung festgelegte Name des Meldungsmoduls ist.

Da die Meldungsaufbereitung von openUTM primär über die NLS-Kataloge durchgeführt wird, hat eine Textänderung im Anwendungs-spezifischen Meldungsmodul (außer im Fehlerfall) für den Ablauf einer UTM-Anwendung keine Wirkung. In diesem Fall, d.h. wenn Anwendungs-spezifisch Meldungstexte modifiziert werden sollen, muss unter dem Verzeichnis *filebase/nls/msg/lang* ein NLS-Katalog *utmsys.cat* für die K-Meldungen bereitgestellt werden. Dieser NLS-Katalog muss nur die geänderten Meldungen enthalten.

Wenn für eine Anwendung nur Meldungstexte, aber keine Meldungsziele geändert werden sollen, kann die von KDCMMOD erzeugte C-Sourcdatei gelöscht werden.

Hinweis

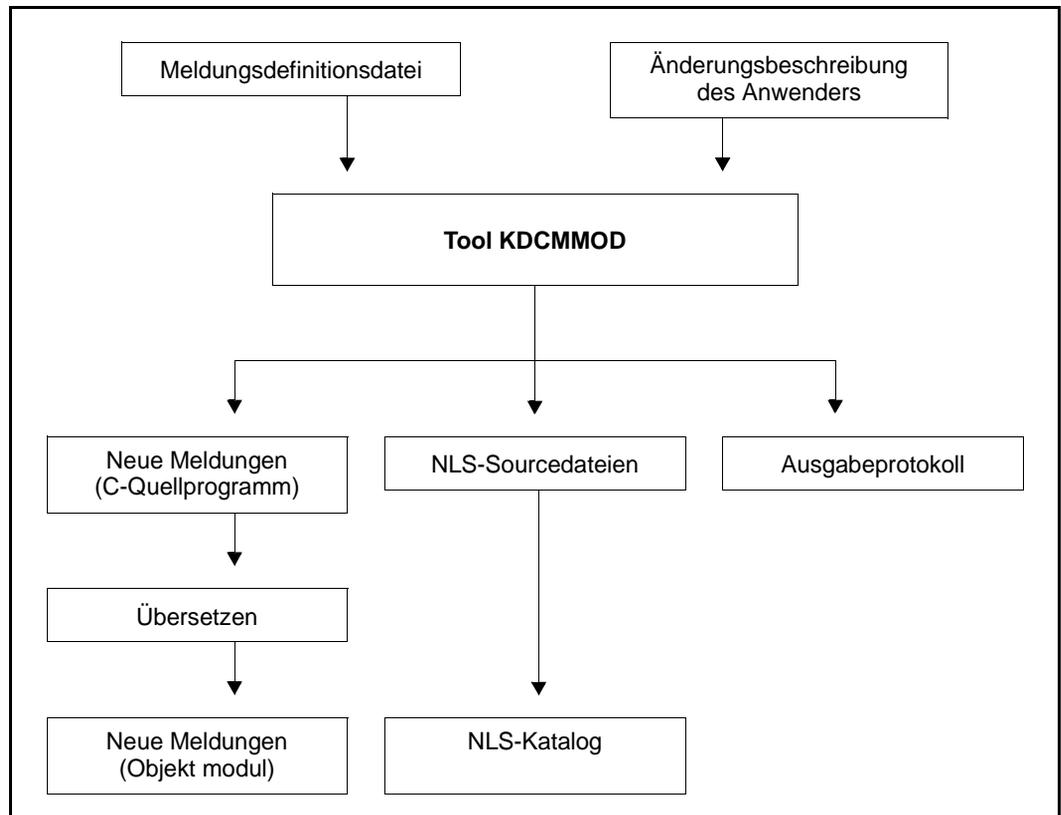
Um unnötige Katalogzugriffe zu vermeiden, sollte ein Anwendungs-spezifischer NLS-Katalog alle Meldungen enthalten. Dazu kopieren Sie den Standard-NLS-Katalog der gewünschten Sprache nach *filebase/nls/msg/lang*, erzeugen mit dem Tool KDCMMOD eine NLS-Sourcedatei mit den geänderten Meldungstexten und mischen diese in den Anwendungs-spezifischen NLS-Katalog ein:

Die Meldungstexte können in einem weit gesteckten Rahmen verändert werden, wobei die Längenbeschränkungen zu beachten sind, siehe [Seite 148](#).

Die zur Verfügung stehenden Inserts finden Sie ab [Seite 385](#). Es ist möglich, Inserts, die im Standardtext enthalten sind, aus dem Meldungstext zu entfernen oder Inserts einer Meldung, die im Standardtext nicht enthalten sind, in den Text aufzunehmen. Außerdem ist es möglich, die Reihenfolge der Inserts innerhalb des Textes zu verändern. Auch die Meldungsnummer kann an eine andere Stelle im Meldungstext verschoben oder sogar ganz aus diesem entfernt werden. Dies sollte aber nur in besonderen Ausnahmefällen gemacht werden, da dies eine Diagnose anhand der Meldungen erschweren kann.

Die Meldungsziele können nur innerhalb gewisser Grenzen geändert werden (siehe dazu [Abschnitt „Ziele der UTM-Meldungen“ auf Seite 398ff](#)). Beachten Sie dabei, dass für jede Meldung unterschiedliche Ziele vorgeschrieben bzw. verboten sein können.

Ein- und Ausgaben von KDCMMOD



Ablauf beim Erstellen eines eigenen Meldungsmoduls

Nachdem das Tool KDCMMOD alle Änderungen bearbeitet und eine neue Meldungsdatei erstellt hat, gibt es als Protokoll auf *stdout* eine aufbereitete Liste aller Meldungen aus. Wenn Sie Meldungen geändert haben, wird auch eine NLS-Sourcedatei erstellt. Außerdem erstellt das Programm eine Übersicht der Meldungsdefinitionen (Insertliste, Ziele, Compression), entstanden aus den Rahmendefinitionen und Modifikationen. In der Ausgabelliste werden die definierten Standardtexte in der Form aufbereitet, in der sie an eine Datensichtstation ausgegeben würden. Dabei werden die Inserts je nach Typ in der entsprechenden Länge wie folgt gefüllt:

CHAR mit @-Zeichen,
 INT mit #-Zeichen,
 HEXA mit X-Zeichen.

Die Meldungen von KDCMMOD finden Sie ab [Seite 350](#).

4.5.2.1 Aufruf von KDCMMOD

Das Tool KDCMMOD rufen Sie auf mit:

utmpfad/ex/kdcmod

4.5.2.2 Steueranweisungen von KDCMMOD

KDCMMOD kennt folgende Steueranweisungen:

CONSTANT	Definieren von Konstanten
END	Ende der Eingabe der Steueranweisungen
ENDMSG	Ende der Meldungsdefinition
GEN	Generieren von Meldungen für eine Funktionseinheit
MODMSG	Ändern von Meldungen
MSGBASE	Auswahl der Meldungsgruppe: UTM (K-Meldungen) oder XAPTP (P-Meldungen)
OPTION	Namen der Meldungsdefinitionsdatei festlegen

Bei Eingabe der Steueranweisungen von KDCMMOD ist zu beachten:

1. Das Tool KDCMMOD liest die Anweisungen der Änderungsbeschreibung von *stdin*. Es werden einzelne Zeilen gelesen, wobei
 - eine Kommentarzeile durch das Zeichen „*“ (Stern) in der 1. Spalte gekennzeichnet wird,
 - für Zeilen mit einem Komma als letztem Zeichen die nächste Zeile als Fortsetzungszeile interpretiert wird.Alle anderen Zeilen werden vom Programm einzeln analysiert.
2. Die OPTION-Anweisung kann nur als erste Steueranweisung angegeben werden.
3. Die CONSTANT-Anweisung muss in der Eingabedatei stehen, bevor eine Textdefinition Bezug auf sie nimmt.
4. Die GEN-Anweisung muss in der Eingabedatei vor allen MODMSG-/ENDMSG-Steueranweisungen stehen.
5. Die END-Anweisung muss am Ende der Eingabedatei stehen.

Anweisung CONSTANT

Die CONSTANT-Anweisung definiert eine Konstante, die man anschließend in den Textdefinitionen der Meldungen verwenden kann.

Operation	Operanden
CONSTANT	constant-name, constant-wert

constant-name

legt den maximal 8 Zeichen langen Namen einer Konstanten fest. Falls eine Konstante mit dem gleichen Namen bereits existiert, wird die CONSTANT-Anweisung als fehlerhaft abgewiesen.

Pflichtoperand.

constant-wert

ordnet *constant-name* einen Wert zu. Der Wert ist entweder als hexadezimale Größe (X'.....') oder abdruckbar in der Form '...' anzugeben. Die maximale Länge beträgt 50 Zeichen.

Pflichtoperand.

Die Konstanten für New Line (NL) und New Page (NP) sind bereits in der Meldungsdefinitionsdatei enthalten.

Anweisung END

Mit der END-Anweisung werden die Anweisungen für KDCMMOD beendet. END muss als letzte Anweisung eingegeben werden.

Operation	Operanden
END	

Ohne Operanden.

Anweisung ENDMSG

Mit der ENDMSG-Anweisung wird eine Meldungstextdefinition abgeschlossen.

Operation	Operanden
ENDMSG	

Ohne Operanden.

Die ENDMSG-Anweisung muss immer in einer eigenen Zeile stehen.

Anweisung GEN

Die GEN-Anweisung gibt an, für welche Funktionseinheit und in welcher Landessprache die Meldungen generiert werden sollen. In einem Programmlauf darf diese Anweisung nur **einmal** gegeben werden und muss vor allen MODMSG-Anweisungen stehen.

Operation	Operanden
GEN	[funktionseinheit] [, LANG=sprachkennzeichen] ,MODULE=objekt-module-name

funktionseinheit

Name der Funktionseinheit, für die Meldungen generiert werden sollen.
Mögliche Funktionseinheiten:

SYS für K- und P-Meldungen

XPROG für U-Meldungen

Standard: SYS

LANG=sprachkennzeichen

bezeichnet die Landessprache, für welche die Meldungstexte generiert werden sollen. Es muss ein Sprachkennzeichen angegeben werden, für das Meldungstexte in der Meldungsdatei vorhanden sind.

Standardwert: GB

MODULE=objekt-modul-name

bezeichnet den Namen des Meldungsmoduls. Er entspricht dem MODULE-Namen in der MESSAGE-Anweisung des Tools KDCDEF (siehe openUTM-Handbuch „Anwendungen generieren“).

Pflichtoperand.

Anweisung MODMSG

Mit der MODMSG-Anweisung ändern Sie eine Meldung der Funktionseinheit, die in der GEN-Anweisung angegeben wurde. Die Anweisung MODMSG leitet die Angaben zur Änderung einer Meldung ein, die Anweisung ENDMSG schließt diese ab.

Operation	Operanden
MODMSG	<pre> msg-nr [,BEL= {YES NO}] ¹ [,COMPRESSION= {YES NO}] [,CONSOLE= {YES NO}] [,MSGTAC= {YES NO}] [,PARTNER= {YES NO}] [,STATION= {YES NO}] [,SYSLINE= {YES NO}] [,SYSLOG= {YES NO}] [,STDOUT= {YES NO}] ² [,STDERR= {YES NO}] ² [,USER-DEST-1= {YES NO}] [,USER-DEST-2= {YES NO}] [,USER-DEST-3= {YES NO}] [,USER-DEST-4= {YES NO}] [text] ³ </pre>

¹ Die Operanden der MODMSG-Anweisung müssen durch ein Komma getrennt werden. Bei Fortsetzungszeilen in einer MODMSG-Anweisung muss das Komma immer als letztes Zeichen in der vorangehenden Zeile stehen (als Fortsetzungszeichen).

² Für STDOUT ist auch das Synonym SYSLST und für STDERR das Synonym SYSOUT zulässig.

³ Die Zeile vor [text] darf nicht mit einem Komma abgeschlossen werden, siehe Beispiel auf [Seite 151](#).

msg-nr gibt an, welche Meldung modifiziert werden soll. Die Meldungsnummer muss im Meldungsnummernbereich der im GEN-Kommando spezifizierten Funktionseinheit liegen. Die Meldungsnummer ist **ohne** das vorangestellte **K** bzw. **P** anzugeben.

Pflichtoperand.

BEL gibt an, ob bei der Ausgabe der Nachricht an das Meldungsziel STATION oder SYSLINE ein akustisches Signal (BELL) ausgelöst werden soll.

YES Ein akustisches Signal wird ausgelöst.

NO Es wird kein akustisches Signal ausgelöst.

Standardwert: NO.

COMPRESSION=

YES Überflüssige Leerzeichen werden aus der Meldung entfernt.

NO Überflüssige Leerzeichen verbleiben in der Meldung.

Der Standardwert ist für die einzelnen Meldungen unterschiedlich. Welcher Wert für welche Meldung voreingestellt ist, entnehmen Sie dem [Abschnitt „Ziele der UTM-Meldungen“ auf Seite 398](#).

Standard: Rahmendefinition

Meldungsziele CONSOLE usw.

bezeichnet die Meldungsziele, an welche die Meldung gesendet (YES) bzw. nicht gesendet (NO) werden soll. Es dürfen nur die Meldungsziele angegeben werden, die in den Rahmendefinitionen mit 'erlaubt' (+) oder 'Standard' (S) angegeben sind. Für alle Meldungsziele, die nicht im MODMSG-Kommando angegeben werden, gelten die Rahmendefinitionen. Dabei bedeuten:

Die Zuordnung der Benutzer-spezifischen Meldungsziele USER-DEST-1,..., USER-DEST-4 zum konkreten Meldungsziel müssen Sie über die KDCDEF-Anweisung MSG-DEST vornehmen.

Die einzelnen Meldungsziele sind auf [Seite 128](#) näher beschrieben.

text

Im MODMSG-Kommando kann ein neuer Meldungstext definiert werden. Wird kein neuer Text definiert, dann gilt der Text, wie im [Kapitel „UTM-Meldungen“ auf Seite 157](#) beschrieben.

Ein neuer Meldungstext wird in einer oder mehreren Zeilen definiert. Er besteht aus einer Folge von Textelementen, die entweder durch Komma oder durch Zeilenende voneinander getrennt sind.

Der Meldungstext sollte kein ASCII'00'-Zeichen enthalten.

Text = Textelement,Textelement,.....

Textelement,.....

Das erste Textelement muss immer in einer neuen Zeile stehen.

Längenbeschränkungen

Der Meldungstext darf inklusive Meldungsnummer und eventuell vorhandener Inserts nicht länger als 512 Zeichen sein. Die Längen der Inserts entnehmen Sie bitte der Tabelle auf [Seite 385](#). Ist der Text länger als 512 Zeichen, dann lehnt KDCMMOD ihn mit der Meldung K686 ab. Die Meldungsdefinitionsdatei wird dann nicht verändert.

Meldungstexte mit dem Meldungsziel SYSLINE (=Systemzeile) dürfen nicht länger als 40 Zeichen sein. Ist der Text länger als 40 Zeichen, so warnt KDCMMOD mit der Meldung K687. Bei der Ausgabe werden längere Meldungen auf 40 Zeichen begrenzt, der Rest wird abgeschnitten.

Folgende Textelemente können verwendet werden:

- Literal** Ein Literal ist eine Zeichenkette, die in Hochkommata eingeschlossen ist. Doppelte Hochkommata in einem Literal ergeben das Zeichen Hochkomma im Literal.
- Insert** Das Textelement Insert ist der Name eines Feldes (Insert) im Meldungstext, dem das Zeichen „&“ vorangestellt wird. Im Text dürfen nur die Feldnamen (Inserts) verwendet werden, die in der Rahmendefinition der betreffenden Meldung angegeben sind.
- Insert-Felder sind Bereiche in Meldungstexten, in die vor Ausgabe der Meldung aktuelle Werte eingetragen werden.
- Konstante** Das Textelement Konstante ist der Name einer Konstanten, der das Zeichen „#“ vorangestellt ist.
- Built-in-Function** Das Textelement Built-in-Function ist eine Funktion, die an der betreffenden Stelle ausgeführt wird. Es gibt die folgende Built-in-Function:
- MSGID**
Zur Erleichterung der Diagnose sollte jeder Meldungstext mit MSGID beginnen.
- Die Funktion liefert für die Meldungsnummer die Zeichenkette für die Meldungs-Identifikation.

Der Meldungstext sollte kein ASCII'00'-Zeichen enthalten.

Jedes MODMSG-Kommando muss mit einer ENDMSG-Anweisung abgeschlossen werden.

Anweisung MSGBASE (für FU SYS)

openUTM unterscheidet innerhalb der FU SYS zwei Meldungsgruppen: Meldungen der Gruppe UTM, die mit dem Buchstaben K beginnen, und Meldungen der Gruppe XAPTP, die mit dem Buchstaben P beginnen.

Mit der Anweisung MSGBASE können Sie die Meldungsgruppe auswählen, auf die sich die nachfolgenden MODMSG-Anweisungen innerhalb der Funktionseinheit SYS beziehen sollen.

Operation	Operanden
MSGBASE	{ <u>UTM</u> XAPTP }

UTM Voreinstellung. Auswahl der Meldungsgruppe UTM (K-Meldungen).

XAPTP Auswahl der Meldungsgruppe XAPTP (P-Meldungen).

Anweisung OPTION

Mit der OPTION-Anweisung legt man den Namen der Meldungsdefinitionsdatei fest, die das Tool KDCMMOD bearbeiten soll.

Operation	Operanden
OPTION	MSGFILE=dateiname

dateiname Name der zu bearbeitenden Meldungsdefinitionsdatei. Er darf bei vollqualifizierter Angabe maximal 54 Zeichen lang sein. Bei teilqualifizierter Angabe darf er maximal 14 Zeichen lang sein.

Wird keine OPTION-Anweisung angegeben, dann wird die Datei mit dem Namen `msgdescription` verwendet.

Beispiel

Der Anwender möchte für die Meldung K006 das Meldungsziel MSGTAC einschalten, für die Meldung K008 den Meldungstext ändern, und ansonsten die unveränderten Meldungen mit englischen Standardtexten verwenden.

Zunächst wird eine Datei `eingabe` erstellt, die folgende KDCMMOD-Steueranweisungen enthält:

```
GEN  SYS,LANG=D,MODULE=mymsgsd
MODMSG 006, MSGTAC=YES
ENDMSG
MODMSG 008
'Hallo ',&USER,', Was willst Du denn schon wieder?'
ENDMSG
END
```

Anschließend wird das Tool KDCMMOD aufgerufen:

```
utmpfad/ex/kdcmmod < eingabe
```

Als Ergebnis wird das C-Quellprogramm `mymsgsd.c` erzeugt, welches mit dem C-Compiler in einen Objektmodul übersetzt werden muss. Beim Binden von `utmwork` (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen unter Unix und Windows“) muss dieser Objektmodul zusätzlich zum Standardmeldungsmodul mit eingebunden werden.

Bei der Übertragung der Änderungen in das Meldungswesen beachten Sie bitte Folgendes:

Wenn Sie mit NLS-Katalogen arbeiten und mit KDCMMOD nur Meldungstexte verändert haben (keine Meldungsziele), dann übertragen Sie mit Hilfe des Unix-Kommandos `gencat` und der NLS-Sourcedatei die Änderungen in den entsprechenden NLS-Meldungskatalog. Sie können so einen modifizierten NLS-Katalog erzeugen, der für alle UTM-Anwendungen im System verwendet wird, oder einen Anwendungs-spezifischen NLS-Katalog, den Sie im Dateiverzeichnis `filebase/nls/msg/lang` ablegen müssen (`filebase` = Dateiverzeichnis der Anwendung).

Die C-Quelldatei für das Meldungsmodul können Sie löschen.

4.5.3 NLS-Meldungskataloge unter Windows-Systemen

In Windows-Systemen werden die Meldungskataloge als DLLs ausgeliefert und befinden sich in den Dateiverzeichnissen %UTMPATH%\nls\msg\de (deutsch) und %UTMPATH%\nls\msg\en (englisch).

Welcher Meldungskatalog verwendet wird, kann über die Umgebungsvariable %LANG% eingestellt werden. Für %LANG% können Sie die Werte „De“ für Deutsch und „En“ für Englisch setzen. Andere Werte werden wie „En“ behandelt.

4.6 UTM-Protokolldatei SYSLOG

openUTM führt für jede UTM-Anwendung eine eigene Protokolldatei, die SYSLOG-Datei. In diese Datei protokolliert openUTM alle Meldungen, die das Meldungsziel SYSLOG haben. Welche Meldungen das sind, können Sie in gewissen Grenzen selbst bestimmen (siehe dazu auch den [Abschnitt „Ziele der UTM-Meldungen“ auf Seite 398](#)).

Die SYSLOG-Datei der Anwendung liegt immer unter dem Dateiverzeichnis *filebase*, wobei *filebase* das Dateiverzeichnis ist, unter dem die Anwendung installiert ist (Basisname der KDCFILE; definiert in MAX KDCFILE).

Die System-Protokolldatei SYSLOG können Sie anlegen als:

- Einzeldatei: Datei mit dem Namen *filebase/SYSLOG*. Sofern es diese Datei beim Start der Anwendung nicht gibt und auch keine FGG eingerichtet wurde, wird diese Datei von openUTM angelegt.
- Dateigenerationsverzeichnis (FGG): Dieses Verzeichnis muss vom Anwender vor dem Start der Anwendung mit dem Tool `kdcsllog` eingerichtet werden. Dabei genügt es, wenn Sie das Dateigenerationsverzeichnis anlegen; die einzelnen Dateien der FGG legt openUTM für Sie an.
Wenn Sie eine automatische Größenüberwachung der SYSLOG-Datei generiert haben, muss die SYSLOG als FGG angelegt werden.

In der SYSLOG-Datei werden Ereignisse aus dem Lauf der Anwendung in Form von UTM-Meldungen protokolliert, die für die laufende Überwachung oder für spätere Kontrollen nützlich sein können (z.B. die Meldungen K033, K070). Insbesondere für die Diagnose liefert die SYSLOG-Datei wichtige Informationen.

Bei jedem Folgestart der Anwendung wird die SYSLOG-Datei von openUTM überschrieben. Die Protokollinformation aus dem vorherigen Anwendungslauf geht verloren. Nach dem Ende eines Anwendungslaufs sollten Sie deshalb, falls nötig, den Inhalt sichern.

4.6.1 SYSLOG-Datei auswerten

openUTM erstellt die SYSLOG-Datei in einem Format, das nicht direkt auswertbar ist. Um sie lesbar ausdrucken oder mit einem Programm bearbeiten zu können, muss man sie aufbereiten. Dazu gibt es die zwei Aufbereitungstools: `KDCCSYSL` und `KDCPSYSL`.

4.6.1.1 Das Tool KDCCSYSL - SYSLOG-Datei aufbereiten

Das Aufbereitungstool KDCCSYSL wandelt eine SYSLOG-Datei in eine andere Datei um, die mit einem benutzereigenen Auswerteprogramm oder mit dem Tool KDCPSYSL weiter verarbeitet werden kann. Jede Meldung dieser neuen Datei besteht aus einem Satz variabler Länge und ist in folgendem Format abgelegt:



SL 2 Byte; Inhalt: Satzlänge binär (Länge der Meldung + 4)

FIL 2 Byte; Inhalt: Leerzeichen

Der Aufbau der einzelnen UTM-Meldungen wird durch die Datenstrukturen *kcmsg.h* (C) oder *KCMSGC* (Cobol) beschrieben. Die Meldungen enthalten keinen Meldungstext.

Die Datenstrukturen für die UTM-Meldungen finden Sie unter:

Unix-Systeme: *utmpfad/include/kcmsg.h* bzw.
 utmpfad/copy-cobo185/KCMSGC (Micro Focus COBOL-Compiler) bzw.
 utmpfad/netcobo1/KCMSGC (NetCOBOL-Compiler)

Windows-Systeme: *utmpfad\include\kcmsg.h* bzw.
 utmpfad\copy-cobo185\KCMSGC (Micro Focus COBOL-Compiler) bzw.
 utmpfad\netcobo1\KCMSGC (NetCOBOL-Compiler)

Aufruf von KDCCSYSL

Unix-Systeme:

utmpfad/ex/kdccsysl file_in file_out

Windows-Systeme in einem DOS-Fenster:

utmpfad\ex\kdccsysl file_in file_out

Bedeutung der Parameter:

file_in ist hierbei die SYSLOG-Datei

file_out ist die von KDCCSYSL aufbereitete SYSLOG-Datei

Die Meldungen von KDCCSYSL werden nach *stderr* ausgegeben.

4.6.1.2 Das Tool KDCPSYSL - Meldungstexte einfügen

KDCPSYSL liest aus der mit dem Aufbereitungstool KDCCSYSL erzeugten Datei und erzeugt eine weitere Datei. Diese enthält zu jeder Meldung die Meldungsnummer und den Meldungstext mit den aktuellen Textparametern, Datum und Uhrzeit sowie zusätzliche aktuelle Parameter, die nicht im Meldungstext enthalten sind. Einige Inserts werden abdruckbar aufbereitet.

Die mit KDCPSYSL erzeugte Datei ist eine normale Textdatei, die pro Meldung eine oder mehrere Zeilen enthält. Diese Datei, die z.B. zur schnellen Übersicht dienen kann, können Sie mit einem Editor ansehen und evtl. weiter verarbeiten oder ausdrucken. Die Sätze der Datei haben eine Länge von bis zu 136 Zeichen (Spalten). Dies ist beim Ausdrucken zu berücksichtigen.

Aufruf von KDCPSYSL

Unix-Systeme:

```
utmpfad/ex/kdcpsysl
```

Windows-Systeme in einem DOS-Fenster:

```
utmpfad\ex\kdcpsysl
```

KDCPSYSL arbeitet mit den festen Dateinamen `slogin` für die mit KDCCSYSL erzeugte Eingabedatei und `slogout` für die aufbereitete Ausgabedatei. Gegebenenfalls ist daher vor dem Aufruf von KDCPSYSL die Datei `file_out` vom KDCCSYSL-Aufruf in `slogin` umzubenennen.

KDCPSYSL bereitet die Daten in `slogin` mit Hilfe von NLS in der durch die Umgebungsvariable LANG eingestellten Sprache auf, sofern unter `utmpfad/nls/msg/eingestellte-Sprache` (Unix-Systeme) bzw. `utmpfad\nls\msg\eingestellte-Sprache` (Windows-Systeme) ein Meldungskatalog `utmsys.cat` existiert. Andernfalls wird der eingebundene Meldungskatalog verwendet.

4.6.1.3 Meldungen von KDCCSYSL und KDCPSYSL

Die Aufbereitungstools KDCCSYSL und KDCPSYSL schreiben ihre Meldungen nach `stderr` (siehe [Abschnitt „Meldungen des UTM-Tools KDCPSYSL“ auf Seite 349](#)).

KDCCSYSL und KDCPSYSL bereiten ihre Meldungen mit Hilfe von NLS in der durch die Umgebungsvariable LANG eingestellten Sprache auf, sofern ein NLS-Meldungskatalog `utmsys.cat` unter folgendem Pfad existiert:

```
utmpfad/nls/msg/eingestellte-Sprache (Unix-Systeme) bzw.  
utmpfad\nls\msg\eingestellte-Sprache (Windows-Systeme)
```

Andernfalls wird der eingebundene Meldungskatalog verwendet.

4.7 Aufbau der UTM-Systemmeldungen

Aufbau der Meldungen, wie sie an SYSLOG oder MSGTAC geschickt werden:

Meldungskopf	Daten der Meldung, abhängig von der Meldungsnummer
1	28 29

Aufbau der Meldungen:

Byte	Format	Bedeutung
1	X' 20'	Leerzeichen
2-5	C' Kxxx' od. C'Pxxx'	Meldungsnummer
6	X' 20'	Leerzeichen
7-17	C' mm/tt/jjddd'	Datum mit mm=Monat, tt=Tag im Monat, jj=Jahr, ddd=Tag im Jahr
18	X' 20'	Leerzeichen
19-24	C' hhmmss'	Uhrzeit mit hh=Stunde, mm=Minute, ss=Sekunde
25-28	C'yyyy'	4-stellige Jahresangabe
ab 29		Daten der Meldung

Die kompletten Datenstrukturen, die dem Aufbau des Meldungskopfes entsprechen, finden Sie unter:

- Unix-Systeme: *utmpfad*/include/kcmsg.h (C-Complier) bzw.
utmpfad/copy-cobo185/KCMSGC (Micro Focus COBOL-Compiler) bzw.
utmpfad/netcobo1/KCMSGC (NetCOBOL-Compiler)
- Windows-Systeme: *utmpfad*\include\kcmsg.h (C-Complier) bzw.
utmpfad\copy-cobo185\KCMSGC (Micro Focus COBOL-Compiler) bzw.
utmpfad\netcobo1\KCMSGC (NetCOBOL-Compiler)

5 UTM-Meldungen

Die folgenden Aufstellungen enthalten alle Meldungen, die von UTM ausgegeben werden können. Die Meldungen der UTM-Dienstprogramme sind in diese Liste integriert. Die Beschreibungen wurden um „Zusatzinformationen“ erweitert, die Ihnen die Reaktion auf die betreffende Meldung erklären sollen.

Vor dem Namen eines Inserts steht ein „&“. UTM-Meldungen können mehr Inserts beinhalten als die Standard-Meldungen. Die Bedeutung der Inserts finden Sie in [Abschnitt „Inserts in Meldungen“ auf Seite 385ff](#) und [Abschnitt „Ziele der UTM-Meldungen“ auf Seite 398ff](#).

5.1 Meldungen des Transaktionsmonitors

- K001** Verbunden mit Anwendung &APPL - Bitte Eingabe
- K002** Verbunden mit Anwendung &APPL - Bitte KDCSIGN
- K003** Das Kommando &CMD ist in dieser Situation nicht erlaubt
- K004** Die Benutzerkennung &USER ist ungueltig - Bitte KDCSIGN
- K005** Die Benutzerkennung &USER ist gesperrt - Bitte KDCSIGN
- K006** Ungueltiges Passwort - Bitte KDCSIGN
- K007** Der Benutzer &USER ist bereits angeschlossen - Bitte KDCSIGN
- K008** KDCSIGN akzeptiert - Bitte Eingabe
- K009** Der Transaktionscode &TAC ist ungueltig (&RCDC) - Bitte Eingabe
Das Insert &RCDC enthält den inkompatiblen Returncode KRCDC, siehe [Abschnitt „Interner Returncode KRCDC“ auf Seite 422](#).
- K010** Der Transaktionscode &TAC ist gesperrt - Bitte Eingabe
- K011** Die Transaktion &ATAC1 wurde akzeptiert - Bitte Eingabe
- K012** &NUMMSGs Nachricht(en) vorhanden
- K013** Fehler bei &CMD - Bitte Eingabe

K016 Die Anwendung wird beendet - Bitte mit KDCOFF beenden

K017 Vorgang &TCVG durch UTM beendet (&RCCC/&RCDC) - Bitte Eingabe

Mit dieser Meldung wird ein Benutzer am Terminal über das anormale Ende des von ihm gestarteten Vorgangs informiert.

Die K017-Meldung besitzt folgende Inserts:

PTRM	Name des PTERMs, von dem aus der Vorgang gestartet wurde.
PRNM	Name des Prozessors, an dem das Terminal angeschlossen ist.
BCAP	Name der Anwendung, an die sich der Benutzer angemeldet hat.
LTRM	Name des LTERMs, von dem aus der Vorgang gestartet wurde.
USER	Name des Benutzers, der den Vorgang gestartet hat.
TCVG	Name des Vorgangs-TACs des Vorgangs.
RCCC	Wert des kompatiblen Returncodes KCRCCC.
RCDC	Wert des inkompatiblen Returncodes KCRDCD.
TAC	Nur bei Returncodes RCDC=KMxx: Name des TACs, der zu dem Vorgangsabbruch geführt hat. Sonst enthält dieses Insert Leerzeichen.

K018 KDCOFF von Anwendung &APPL akzeptiert - Bitte Anmelden

K019 KDCOFF akzeptiert

K020 Es ist keine Nachricht vorhanden

K021 Eine Eingabe ist nicht in der vorgegebenen Zeit erfolgt

K022 Die folgende Nachricht von &BCAP wurde eventuell bereits gesendet

K023 &OMSG2

Die Meldung wird durch das Administrationskommando KDCSHUT WARN bzw. den entsprechenden Aufruf an der Administrator-Schnittstelle erzeugt.

K024 Die Eingabenachricht ging verloren - Bitte wiederholen

K025 Die Ausgabenachricht der Anwendung &BCAP an LTERM <RM wurde verkuerzt

Eine Nachricht an einen LTERM-Partner konnte nicht vollständig gesendet werden, die Nachricht wurde verkürzt ausgegeben.

Mögliche Maßnahme zur Behebung des Fehlers: Vergrößern der Längenangabe im Operanden TRMSGLTH in der KDCDEF-Generierung.

K026 Mitteilung an <RM akzeptiert - Bitte Eingabe

K027 LTERM-Partner <RM gesperrt - Administrator verstaendigen oder KDCOFF eingeben

K028 Bitte Passwort eingeben

K029 Bitte Karte einlegen

K030 Es ist ein Kartenleser erforderlich - Bitte Anmelden

K031 Falsche Karte - Bitte Anmelden

K032 UTM-D Verbindungsmeldung : &CON/&PRNM/&BCAP/&LPAP/&USER; Grund1: &RCF1B; Grund2: &RCF2B

Bei der folgenden Beschreibung der Returncodes &RCF1B und &RCF2B ist PLU (Primary Logical Unit) die Anwendung, in der SESCHA PLU=NO generiert wurde, und SLU (Secondary Logical Unit) die Anwendung, in der PLU=YES angegeben wurde.

Das Insert &USER enthält den UTM-Sessionnamen der Verbindung (LSES-Name).

&RCF1B	Bedeutung	
C01	BIND_REJECTED	Der BIND-Request oder die BIND-Response wurde abgelehnt. Session kann nicht zustandekommen.
C02	BIND_CHANGED	BIND-Parameter wurden modifiziert. Der BIND-Request darf von SLU modifiziert werden, PLU kann die BIND Response nicht verändern, nur ablehnen. Kann die BIND-Response von PLU nicht akzeptiert werden, kommt keine Session zustande.
C03	CONNECTION / SESSION FAILED	Eine Verbindung bzw. Session kann nicht aufgebaut werden.
C04	SESSION_FAILED	Sessionwarmstart konnte nicht durchgeführt werden.
C05	STSN_CHANGED	SLU kann den vorgeschlagenen Aufsetzpunkt nicht akzeptieren.

&RCF2B	Bedeutung und evtl. Maßnahme
CR00	Session und Connection gehören nicht zum gleichen LPAP. Maßnahme: Die KDCDEF-Generierung überprüfen.
CR01	Beim Sessionwarmstart konnten die Sessions keinen gültigen Aufsetzpunkt vereinbaren. Die Anwendungen wurden wahrscheinlich mit verschiedenen KDCDEF-Ständen gestartet. Maßnahme: Überprüfen, ob eine der beiden Anwendungen neu generiert wurde.

&RCF2B	Bedeutung und evtl. Maßnahme
CR02	SLU hat einen neuen Aufsetzpunkt vorgeschlagen. Kommt die Session nicht zustande, wurden die Anwendungen wahrscheinlich mit verschiedenen KDCDEF-Ständen gestartet. Maßnahme: Überprüfen, ob eine der beiden Anwendung neu generiert wurde.
CR03	QUIET-Kommando gegeben.
CR04	Keine geeignete Session verfügbar; Gründe: <ul style="list-style-type: none"> – Es wurden mehr Verbindungen als Sessions generiert – KDCLSES-Kommando mit ACT=QUIET gegeben – Eine Session wurde noch nicht abgebaut.
CR05	Verbindungsaufbauanforderung von Transit-CD oder Partner-Anwendung erzeugt für diese Verbindung NEA-Protokolle
CR06	Verbindungsaufbau in Bearbeitung.
CR07	Inkonsistenz bzgl. PLU-Angabe in der Verbindungsnachricht vom Partner und in der Generierung. Evtl. wurden beide Partner als PLU generiert. Maßnahme: Generierungen überprüfen.
CR08	Inkonsistenz bzgl. PLU-Angabe in der Verbindungsnachricht vom Partner und in der Generierung. Evtl. wurden beide Partner als PLU generiert. Maßnahme: Generierungen überprüfen.
CR09 CR10 CR11 CR12	Fehlerhafter Aufbau einer Verbindungsnachricht.
CR13	Betriebsmittelengpass: kein Slot für dynamische Tabellen vorhanden.
CR14	In der Verbindungsnachricht ist der Name der PLU Session falsch Maßnahme: Generierungen überprüfen.
CR15 CR16 CR19 CR21	Session ist noch aufgebaut.
CR17 CR18 CR20	Keine geeignete Session verfügbar; Gründe: <ul style="list-style-type: none"> – Es wurden mehr Verbindungen als Sessions generiert – KDCLSES-Kommando mit ACT=QUIET gegeben – Eine Session wurde noch nicht abgebaut.
SC00	Kein Grund angegeben.

&RCF2B	Bedeutung und evtl. Maßnahme
SC01 ¹	Der Partner meldet sich mit einem falschen FM-Profil. UTM unterstützt nur FM-Profile 18.
SC02 ¹	Der Partner meldet sich mit einem falschen TS-Profile. UTM unterstützt TS-Profile 4.
SC03 ¹	UTM kann eine Nachricht in kleineren Teilen (Request Units) bearbeiten. Deshalb muss PLU sog. „Multiple-RU-Chains“ unterstützen.
SC04 ¹	Eine evtl. angeforderte Antwort muss gekommen sein, bevor die nächste Anforderung geschickt werden darf. PLU muss deshalb im „Immediate-Request-Mode“ arbeiten.
SC05 ¹	Abhängig von dem Nachrichtentyp fordert UTM immer eine Quittung bzw. nur eine negative Quittung. PLU muss deshalb sowohl „definite“ als auch „exception response“ akzeptieren.
SC07 ¹	Bei der Nachrichtenübertragung wird Datenkomprimierung nicht unterstützt.
SC08 ¹	PLU muss immer die Möglichkeit haben, eine Bracket zu schließen.
SC09 ¹	Entspricht Reason SC03, aber für SLU.
SC10 ¹	Entspricht Reason SC04, aber für SLU.
SC11 ¹	Entspricht Reason SC05, aber für SLU.
SC12 ¹	UTM unterstützt kein „2-Phase-Commit“.
SC13 ¹	Entspricht Reason SC07, aber für SLU.
SC14 ¹	Entspricht Reason SC08, aber für SLU.
SC15 ¹	FM-Header müssen erlaubt sein.
SC16	Obwohl ein Partner einen Session-Kaltstart für ausreichend hält, besteht die andere Session auf einem Session-Warmstart.
SC17	Der Partner hat die Session als nicht abgeschlossen (in Bracket) bezeichnet, obwohl er keinen Warmstart wünscht. Die Session wird als abgeschlossen angenommen (BETB).
SC18 ¹	Ein Vorgang wird erst abgeschlossen, wenn er von beiden Partnern als abgeschlossen bezeichnet ist. D.h UTM benutzt die sog. „Bracket Termination Rule 1“.
SC19	Die USER-DATA im BIND sind nicht korrekt. Vermutlich sendet der Partner einen fehlerhaften BIND. Zur Diagnose wird der UTM-BCAM-Trace benötigt.
SC23 ¹	Bei VTV wird mit „Half-duplex flip-flop“ Sende- bzw. Empfangsmodus gearbeitet. D.h. zu jedem Zeitpunkt hat nur ein Partner Sendeerlaubnis.
SC24 ¹	Beide Partner müssen für evtl. Sessionwiederanlauf verantwortlich sein.

&RCF2B	Bedeutung und evtl. Maßnahme
SC25	Inkonsistenz bezüglich Contention-Winner im BIND und der KDCDEF-Generierung. Maßnahme: Beide Generierungen müssen bezüglich der Angabe Contention-Winner überprüft werden (KDCDEF-Steueranweisung SESCHA CONTWIN=Y/N)
SC26	Beim Session-Warmstart will PLU die Sendeberechtigung bekommen. Kommt die Session nicht zustande, wurden die Anwendungen wahrscheinlich mit verschiedenen KDCDEF-Ständen gestartet. Maßnahme: Überprüfen, ob eine der beiden Anwendungen neu generiert wurde.
SC27	Beim Session-Warmstart will SLU die Sendeberechtigung bekommen. Kommt die Session nicht zustande, wurden die Anwendungen wahrscheinlich mit verschiedenen KDCDEF-Ständen gestartet. Maßnahme: Überprüfen, ob eine der beiden Anwendungen neu generiert wurde.
SC28 ¹	UTM unterstützt „Two-Stage-Pacing“.
SC29	Inkonsistenz der Pacing-Counter. D.h. SLU-Receive-Counter ist ungleich PLU-Send-Counter bzw. PLU-Receive-Counter ist ungleich SLU-Send-Counter Maßnahme: Beide Generierungen müssen bezüglich des Pacing-Counter (=Window-Size) überprüft werden. (KDCDEF-Steueranweisung SESCHA PACCNT=)
SC30 ¹	In der BIND-Response schlägt SLU als „maximale Teilnachrichtenlänge von PLU“ einen nicht erlaubten Wert vor.
SC31 ¹	In der BIND-Response schlägt SLU als „maximale Teilnachrichtenlänge von SLU“ einen nicht erlaubten Wert vor.
SC32	In dem BIND-Request schlägt PLU als „maximale Teilnachrichtenlänge von SLU“ einen nicht erlaubten Wert vor.
SC33	In dem BIND-Request schlägt PLU als „maximale Teilnachrichtenlänge von PLU“ einen Wert vor, der von SLU modifiziert wurde.
SC34 ¹	Presentation-Service (PS) Usage field format muss als „basic format“ definiert sein.
SC35 ¹	Als Session-Type ist nur LU-Session Type 6 erlaubt.
SC36 ¹	Nachrichtenverschlüsselung wird nicht unterstützt.
SC37	In der BIND-Response ist der Name der PLU-Session falsch. Maßnahme: Beide Generierungen müssen bezüglich der Verknüpfung LSES - RSES überprüft werden (KDCDEF-Steueranweisung LSES RSES = Name)
SC38	In dem BIND-Request ist der Name der PLU-Session falsch. Maßnahme: Siehe Reason SC37

&RCF2B	Bedeutung und evtl. Maßnahme
SC39	In der BIND-Response ist der Name der SLU-Session falsch. Maßnahme: Siehe Reason SC37
SC40	In dem BIND-Request ist der Name der SLU-Session falsch. Maßnahme: Siehe Reason SC37
SC41 ¹	User Request Correlation (URC) wird von UTM-D nicht unterstützt.
SY01	Session termination pending. Der Abbau der Session ist noch nicht abgeschlossen.
SY02	PET no send request Die Session ist im Zustand PTC und es existiert keine Nachricht für diesen Partner.
SY03	PEND RS pending. Die PEND RS-Behandlung ist noch nicht abgeschlossen.
SY04	job-submitting conversation active. Der Auftraggeber-Vorgang ist aktiv.
SY05	job submitter not available. Der Auftraggeber ist nicht angemeldet oder der Auftraggeber-Vorgang ist gequeued.
SY06	RESTART_VIA_SEND_RQ. Die Session befindet sich im Zustand IN_BRACKET und es liegt keine Nachricht an diesen Partner vor.
SY07	Session active Die Session ist bereits aktiv.

¹ Nur bei heterogenen Kopplungen

K033 USER / LSES aktiv : &PTRM/&PRNM/&BCAP/<RM/&USER &REST,
&GLOBALSG

Die Meldung K033 wird ausgegeben

- nach erfolgreichem Verbindungsaufbau, wenn ohne USER gearbeitet wird,
- nach erfolgreicher Anmeldung, wenn mit USERn gearbeitet wird,
- nach erfolgreichem Binden einer Session bei einer VTV-Verbindung.

&REST	Bedeutung
Y	Vorgangswiederanlauf
N	kein Vorgangswiederanlauf
U	undefiniert, nur bei UTM-D

Meldet sich ein Benutzer über eine OSI TP-Verbindung an, haben die ersten Inserts folgende Inhalte:

Insert	Inhalt
&PTRM	OSI-CON Name
&PRNM	8 Leerzeichen
&BCAP	ACCESS-POINT Name
<RM	OSI-LPAP Name

Das Insert &GLOBALSG 'Cluster Global Sign' ist nur in UTM-Cluster-Anwendungen versorgt und kann folgende Werte annehmen:

&GLOBALSG	Bedeutung
Y	Cluster-globale Anmeldung
N	Knoten-lokale Anmeldung
A	Der Benutzer war bereits Cluster-global an dieser Knoten-Anwendung angemeldet.
' '	Anmeldung für einen LU6.1-Session-User oder einen Verbindungs-User.

K034 Transaktion wurde zurueckgesetzt

K035 Vorgangswiederanlauf in Bearbeitung

K036 Verbindungsaufbau : &PTRM/&PRNM/&BCAP/<RM &RSLT, &REA1

&RSLT	Bedeutung
Y	Verbindung aufgebaut
N	Verbindung wurde nicht aufgebaut; die Ursache wird in &REA1 angegeben.

&REA1	Bedeutung
X'00'	Verbindung aufgebaut
X'01'	PTERM-/CON-Name unbekannt
X'02'	Prozessorname unbekannt
X'03'	PTYPE nicht akzeptiert
X'04'	kein LTERM zugeordnet
X'05'	falsches Verbindungspasswort angegeben
X'06'	STATUS=OFF für dieses PTERM/CON
X'07'	STATUS=OFF für dieses LTERM/LPAP
X'08'	Betriebsmittelengpass
X'0A'	SHUTDOWN der Anwendung
X'0B'	Partner bereits angeschlossen
X'0C'	Verbindungsabbau in Bearbeitung
X'0D'	Negativer Returncode vom Transportsystem, siehe zugehörige K065-Meldung.
X'0E'	der Partner ist an einer anderen Anwendung generiert
X'0F'	UTM-D Verbindungswunsch abgelehnt; z.B. wegen Generierungsfehler oder da QUIET-Kommando gegeben
X'10'	PTERM-Name unbekannt und kein LTERM-Pool für diesen Prozessornamen generiert
X'11'	kein LTERM-Pool für diesen PTYPE vorhanden und der PTERM-Name ist unbekannt
X'12'	kein weiterer freier Eintrag im Terminalpool vorhanden
X'13'	Partner ist bereits an den LTERM-Pool angeschlossen
X'14'	Verbindung abgelehnt wegen nicht hinreichender Eigenschaften der Transportverbindung (GROS)
X'16'	Benutzer ist bereits angemeldet
X'17'	Der Verbindungswunsch wurde wegen contention abgelehnt. Der Partner hat selbst einen Verbindungsaufbau initiiert
X'1B'	Beim Start der Anwendung konnte die IP-Adresse des Partner-Rechners nicht ermittelt werden.

&REA1	Bedeutung
X'1F'	LPAP ist auf quiet gesetzt
X'20'	Es ist keine geeignete Session verfügbar
X'21'	Die Session ist bereits aktiv
X'22'	Session Termination ist in Bearbeitung
X'23'	Lokale Seite ist im PTC und hat keine Ausgabe-Nachricht für diese Session; zum Sessionausbau muss auf den Wiederanlauf des lokalen Vorgangs gewartet werden.
X'24'	PEND RS ist in Bearbeitung
X'25'	Der AG-Vorgang ist aktiv
X'26'	Der AG-User ist nicht verfügbar
X'27'	Zum Sessionaufbau muss auf den Wiederanlauf des lokalen Vorgangs gewartet werden
X'29'	Der Terminalpool ist mit einem anderen UTM-Anwendungsnamen generiert
X'2E'	Die Verbindung ist noch nicht vollständig abgebaut.
X'45'	Während des Wartens auf eine Bestätigung des Verbindungsaufbaus traf ein Verbindungsabbauwunsch ein.
X'46'	PTERM/CON wurde per Administration gelöscht.
X'48'	Es ist schon ein PTERM mit dem Namen des Multi-Pool-LTERM generiert.
X'55'	Abgelehnter Socket-Verbindungsaufbau.
X'58'	Verbindungsaufbaubestätigung abgelehnt, da die Verbindung zwischenzeitlich wieder abgebaut wurde.

Mit dieser Meldung, die standardmäßig auf SYSLOG ausgegeben wird, können Probleme beim Verbindungsaufbau erkannt werden.

Bei Ausgabe der Meldung wird das Insert &REA1 abdruckbar aufbereitet, während es an der Programmschnittstelle zu MSGTAC / SYSLOG als Hexadezimalwert übergeben wird.

K038 Syntaxfehler &SYN

K039 Die Startparameter CLUSTER-FILEBASE und &STRTPAR schliessen sich gegenseitig aus!!

K040 Die Warnungsstufe &WLEV fuer &PGPOOL wurde unterschritten

K041 Die Warnungsstufe &WLEV fuer &PGPOOL wurde ueberschritten

K042 Fehler beim Lesen der Startparameter

K043 DMS Fehler &DMSE fuer Datei &FNAM aufgetreten

Im Insert &DMSE wird der DMS-Fehlercode ausgegeben. Die möglichen DMS-Fehlercodes sind auf [Seite 383](#) beschrieben.

Für UTM-Cluster-Dateien kann das Insert &DMSE neben den vom Betriebssystem gemeldeten Fehlercodes folgende weitere Fehlercodes enthalten:

ILCK Es ist ein Fehler beim Initialisieren eines Datei-Locks aufgetreten.

LOCK Es ist ein Fehler bei Anfordern eines Datei-Locks aufgetreten.

ULCK Es ist ein Fehler bei der Freigabe eines Datei-Locks aufgetreten.

DLCK Es ist ein Fehler beim Zerstören eines Datei-Locks aufgetreten.

K044 Lesen der Startparameter**K045** --- Kein Meldungstext ---

Als einziges Meldungsziel dieser Meldung ist MSGTAC zugelassen; daher ist zu dieser Meldung kein Meldungstext definiert.

Die Ausgabe der Meldung erfolgt, wenn die Ausgabe einer Nachricht im Quittungsmodus mit einer positiven Abdruckquittung beendet wurde.

Standardmäßig ist kein Nachrichtenziel für diese Meldung definiert.

K046 Fehler bei Druckausgabe: &PALTRM / &CID / &DPID / &ERPRT

Die Meldung wird nur für Drucker, die einer Druckersteuerstation zugeordnet sind, ausgegeben. Sie ist vor allem für eine programmierte Fehlerbehandlung mit Hilfe des MSGTAC-Teilprogramms bestimmt.

Weitere Inserts dieser Meldung sind: &PTRM, &PRNM, &BCAP, <RM, &IMSG2

Die Meldung wird in folgenden Fällen ausgegeben:

- Beenden einer Druckausgabe mit negativer Abdruckquittung
- Wiederholung einer Druckausgabe (zusätzlich zur K022-Meldung)
- Verbindungsaufbau zu einem Drucker wird abgelehnt

Standardmäßig ist für diese Meldung kein Meldungsziel definiert.

Das Insert &ERPRT gibt Auskunft über den vorliegenden Fehlerfall, im Insert &IMSG2 werden abhängig davon weitere Informationen zur Verfügung gestellt.

&ERPRT	Bedeutung und Inhalt von &IMSG2
A	Negative Abdruckquittung. Wenn der Drucker eine logische Abdruckquittung liefert (Rückmeldung), so enthält &IMSG2 die ersten 32 Byte dieser Abdruckquittung (ggf. mit Leerzeichen ergänzt). Der genaue Aufbau dieser Abdruckquittung ist in den entsprechenden Geräte-Benutzerhandbüchern nachzulesen.
R	Wiederholung einer Druckausgabe. &IMSG2 ist ohne Bedeutung.
C	Verbindungsaufbau wird abgelehnt &IMSG2 enthält das DIAGNOSTIC-WORD.

K047 Ungültiger SVC 134 - Aufruf (SVC fuer UTM): Opcode = &OPCD1, Returncode = &RTCD

Der SVC-Aufruf ist ein UTM interner Schnittstellen-Übergang. In der Meldung K047 kann der Returncode (RC) folgende Werte annehmen:

nn	Bedeutung
04	Die Adresse von ROOTDATA ist ungültig (UTM-Fehler ¹).
08	Der übersetzte Root-Source und die Bibliothek mit den UTM Systemmodulen passen nicht zusammen. Es sind zwei Lösungen dieses Problems möglich: <ul style="list-style-type: none"> <li data-bbox="472 966 1293 1021">– Den Root-Source mit den korrekten Includes aus utmpfad/include übersetzen <li data-bbox="472 1021 1293 1075">– die Anwendung mit der korrekten Bibliothek utmpfad/sys/libwork.a (UTM Systemmodule) binden
0C	Die Adresse der Parameterliste ist ungültig (UTM-Fehler).
10	Die Adresse des KDCS-Parameterbereichs ist ungültig (UTM-Fehler). Bei Verwendung der von openUTM zur Verfügung gestellten Include-Dateien für die KDCS-Parameterbereiche handelt es sich um einen UTM-Fehler. Werden im Teilprogramm selbst definierte Bereiche für den KDCS-Parameterbereich verwendet, handelt es sich um einen Anwenderfehler. Maßnahme: Ausrichtung und Adresse des Parameterbereiches überprüfen.
14	Die Adresse des KTA in ROOTDATA ist ungültig. Mögliche Ursachen: <ul style="list-style-type: none"> <li data-bbox="472 1357 1293 1385">– UTM-Fehler <li data-bbox="472 1385 1293 1412">– KDCS-Aufruf im START-EXIT-Teilprogramm <li data-bbox="472 1412 1293 1439">– KDCS-Aufruf im SHUT-EXIT-Teilprogramm <li data-bbox="472 1439 1293 1466">– KDCS-Aufruf nach UTM Term Application
18	Der Inhalt des KTA-Headers ist ungültig. Mögliche Ursache: UTM-Fehler
1C	ROOTDATA und KAA stimmen bezüglich des Applikationsnamens nicht überein oder der KAA-Header ist ungültig (UTM-Fehler).

nn	Bedeutung
20	ROOTDATA und KTA stimmen bezüglich des Applikationsnamens nicht überein (UTM-Fehler).
24	Die Adresse der Parameterliste ist nicht auf Wortgrenze ausgerichtet (UTM-Fehler).
28	Der KDCS-Parameterbereich ist nicht ausgerichtet. Bei Verwendung der von openUTM zur Verfügung gestellten Include-Dateien für die KDCS-Parameterbereiche handelt es sich um einen UTM-Fehler. Werden im Teilprogramm selbst definierte Bereiche für den KDCS-Parameterbereich verwendet, handelt es sich um einen Anwenderfehler (Ausrichtung auf Halbwort wird geprüft). Maßnahme: Ausrichtung und Adresse des Parameterbereiches überprüfen.

¹ Bei Returncodes vom Typ UTM-Fehler: Service soll Problemmeldung schreiben

K049 Fehler &RCCC2 bei Start der Anwendung aufgetreten

UTM gibt die Meldung K049 aus, wenn der Start eines UTM-Tasks wegen eines Fehlers abgebrochen wurde, der Fehlercode &RCCC2 zeigt die Ursache des Fehlers an. Die folgende Tabelle enthält die möglichen Fehlercodes, die Fehlerursache sowie mögliche Maßnahmen zur Behebung des Fehlers.

Start-Fehlercodes

Code	Fehlerursache	Maßnahme
2	Für den ROOTDATA-Bereich steht nicht ausreichend Speicherplatz zur Verfügung.	Generierung prüfen, Systemkonfiguration prüfen
3	Der ROOTDATA-Bereich ist nicht allokiert oder der String ' ROOTDATA ' ist nicht vorhanden.	Konsistente UTM-Version herstellen.
4	Die Versionen von KDCROOT und den UTM-Systemmodulen sind verschieden.	Konsistente UTM- Version herstellen.
5	Anwendung kann beim Name Manager nicht angemeldet werden.	Systemfehler, bitte Service informieren.
6	Der Start einer Folgetask wird abgebrochen, weil der Start der 1. Task ebenfalls abgebrochen wurde.	siehe Fehlercode des 1. Task
7	Der Meldungsmodul des Anwenders ist inkonsistent zu dem Meldungsmodul, der zum UTM-Systemcode gehört (z.B. falsche Anzahl von Inserts).	Meldungsmodul in Ordnung bringen und Anwendung neu binden.
10	Ein UTM-Task fordert 4 KB Speicher an, bekommt ihn aber nicht.	System-Generierung und Speicherbedarf der Anwendung überprüfen.

Code	Fehlerursache	Maßnahme
11	Fehler beim 1. OPEN-Aufruf für die KDCA-Datei von KDCFILE, evtl. falscher FILEBASE-Name in den Startparametern. Mögliche Ursache in UTM-Cluster-Anwendungen: Für den zu startenden Knoten läuft noch ein Online-Import.	siehe DMS-Fehlercode, Seite 383 .
12	DMS-Fehler beim Lesen des ersten 2KB-Bereichs im KAA (KDC Application Area) der KDCA-Datei	siehe DMS-Fehlercode, Seite 383 .
13	DMS-Fehler beim Lesen des ersten oder letzten Kontrollbereichs der KDCA-Datei, die Datei ist nicht verwendbar	Kopie nehmen oder neu generieren mit KDCDEF.
14	Die UTM-Version in der KAA (KDC Application Area), die aus der KDCFILE gelesen wird, stimmt nicht mit der Version des UTM-Systemcodes überein. Der Fehler kann auftreten, wenn Sie versuchen, zwei UTM-Anwendungen mit demselben Namen in verschiedenen UTM-Versionen parallel zu betreiben.	Name einer Anwendung ändern (MAX APPLNAME=...)
15	Inkonsistenz zwischen KDCDEF und UTM-Code hinsichtlich KAA-Struktur.	Neu generieren mit richtiger KDCDEF
16	Die Versionen von KDCDEF und UTM-Systemmodulen sind verschieden.	konsistente Versionen herstellen
17	Vor dem Start hat Programm KDCUPD auf KDCFILE gearbeitet und sich nicht normal beendet.	konsistenten KDCFILE herstellen mit KDCUPD oder KDCDEF
18	Die KDCFILE einer Folge-Task hat eine andere Generierungszeit als die der ersten Task.	Folge-Task mit gleicher KDCFILE starten wie 1. Task
20	Wegen Adressraumangel gelingt es der 1. Task der UTM-Anwendung nicht, den Shared Memory für KAA einzurichten.	Unix-Generierung und UTM Generierung überprüfen
21	Wegen Adressraumangel gelingt es einer Folgetask der UTM-Anwendung nicht, sich an den KAA Shared Memory anzuschließen. Oder Anwendung wird bereits beendet.	wie 20
22	DMS-Fehler beim Lesen eines NSR-Bereichs in KAA	siehe DMS-Fehlercode, Seite 383 .
23	Ein KAA NSR-Bereich ist zerstört, die KDCA-Datei ist nicht mehr verwendbar.	Neu generieren mit KDCDEF.
24	DMS-Fehler beim Lesen eines SR-Bereichs in KAA	siehe DMS-Fehlercode, Seite 383 .

Code	Fehlerursache	Maßnahme
25	Ein KAA SR-Bereich ist zerstört, die KDCA-Datei ist nicht mehr verwendbar.	Neu generieren mit KDCDEF.
26	Mindestens ein Programm für einen generierten Benutzerausgang oder einen ereignisgesteuerten Vorgang fehlt; d.h. das Programm ist weder eingebunden noch konnte es nachgeladen werden.	Programm in die bei der Generierung angegebene Bibliothek bringen, evtl. neu binden.
27	KCSBKAA lieferte einen schlechten Returncode. Beim Initialisieren der SYSLOG-Daten im KAA durch KCSSLOG ist ein Fehler aufgetreten.	Problemmeldung schreiben.
28	Bei einem (Folge-)Start der Anwendung ist der Generierungsparameter FORMSYS inkonsistent zwischen ROOT und KDCFILE-Generierung! (s. auch Codes 100 und folgende) Der Fehler tritt dann auf, wenn KDCFILE und Root-Source nicht zusammenpassen. UTM lehnt dies ab, um eventuelle Folgefehler zu verhindern.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen
29	– Das Teilprogramm der Administration (KDCADM) fehlt oder – der TAC KDCSHUT ist nicht generiert.	– Programm KDCADM in die Generierung und ins UTM-Anwendungsprogramm aufnehmen. – TAC KDCSHUT generieren.
30	Eine UTM-Task hat einen zusammenhängenden Bereich für die Task-spezifischen UTM-Tabellen (= KTA) im Kl.5 Speicher angefordert, ihn aber nicht bekommen.	System-Generierung und Speicherbedarf überprüfen, KTA kleiner machen durch Änderung von KDCDEF-Parametern.
31	Es wurde versucht, mehr Tasks zu starten, als generiert sind.	Anwenderfehler.
32	Es gelang nicht, die Task-Lock-Börse zu erzeugen. Dieser Fehler tritt auf, wenn zu wenig Semaphoreinträge für die UTM-Prozesse zur Verfügung stehen. Das kann vorkommen, wenn sich das Beenden eines Prozesses und das Nachstarten überlappen.	Mit größerer Semaphorezahl neu generieren.
33	Der Start einer weiteren Task für die Anwendung wird von UTM abgelehnt, weil die Anwendung schon beendet wird (normale Beendigung oder Abbruch).	normales Verhalten.
34	Die KDCFILE einer Folge-Task unterscheidet sich hinsichtlich der KAA-Größe von der ersten Task.	Folge-Task mit gleicher KDCFILE starten wie 1. Task

Code	Fehlerursache	Maßnahme
35	Beim Nachladen des Anwendungsprogramms stellt UTM fest, dass die Anwendung abgebrochen wird.	normales Verhalten.
36	Fehler beim Name Manager Aufruf für eine Folgetask einer Anwendung	Systemfehler, bitte Service informieren.
37	Die Anzahl zu startender Tasks ist für die Bearbeitung von PGWT-Aufrufen bzw. für eine UTM-Cluster-Anwendung nicht ausreichend.	Mindestens 2 Tasks starten.
39	Beim Nachladen des Anwendungsprogramms stellt UTM fest, dass mittlerweile die Generierung unzulässig geändert wurde.	Startprozedur und Generierung überprüfen.
40	Anzahl der Einträge in der Programm-, Lademodul- oder Meldungsmodultabelle im ROOT und in der KDCFILE stimmen nicht überein.	KDCFILE und/oder ROOT mit gleichen Generierungsparametern erzeugen.
41	Ein Attribut eines Programms im ROOT und in der KDCFILE stimmt nicht überein. (siehe vorangegangene K073-Meldung).	KDCFILE und/oder ROOT mit gleichen Generierungsparametern erzeugen.
44	Die Generierungen von KDCFILE und ROOT stimmen hinsichtlich MAX TRMSGLTH nicht überein.	KDCFILE oder ROOT mit gleichen Generierungsparametern erzeugen.
45	Fehler von KCSLKLK beim Setzen eines TAC-Klassen-Locks	Generierungsparameter MAX RESWAIT=(...,zeit2) größer machen.
46	UTM-Cluster-Anwendung: Eine Folge-Task wurde mit anderen Angaben bezüglich Cluster-Filebase gestartet als die erste Task.	1. Task und Folge-Task mit gleichen Startparametern starten
47	Die Generierungen von KDCFILE und ROOT stimmen hinsichtlich der Nutzung von Shared Objects / dlls nicht überein.	KDCFILE oder ROOT mit gleichen Generierungsparametern erzeugen.
48	Fehler beim Initialisieren des Buffer Managements in Start-Case 1 oder 2	Den virtuellen Adressraum erhöhen; wenn das nicht hilft, PM schreiben
50	Die 1. UTM-Task versucht, den Cache Shared Memory einzurichten, oder eine Folgetask will sich anschließen. Wegen Adressraum-mangel gelingt dies aber nicht.	System-Generierung und Speicherbedarf der UTM-Anwendung überprüfen.
51	Die Datei stammt nicht von KDCDEF. Dateiname siehe K082-Meldung.	Richtige KDCFILE beschaffen oder neu generieren.
52	Die Datei wurde mit einer anderen KDCFILE vertauscht, z.B. Pool-Datei statt KDCB-Datei. Dateiname siehe K082-Meldung.	wie 51.

Code	Fehlerursache	Maßnahme
53	Die Datei passt nicht zur KDCA-Datei. z.B: KDCDEF Uhrzeit verschieden. Dateiname siehe K082-Meldung. Ein Kurzprotokoll nach <i>stdout</i> zeigt die genaue Ursache.	wie 51.
54	Die Konsistenzprüfung ergibt, dass die Datei zerstört ist. Evtl. liegt ein Versionsmix vor. Dateiname siehe K083-Meldung.	wie 51. Versionen überprüfen.
55	DMS-Fehler bei KDCA-/KDCB-Datei	siehe DMS-Fehlercode, Seite 383 .
56	DMS-Fehler bei Pagepool-Datei	
57	DMS-Fehler bei Wiederanlauf-Datei	
58	Fehler im Zusammenhang mit der vom Benutzer bereitzustellenden SYSLOG-Datei. Mögliche Ursachen: – Das Directory <filebase>/SYSLOG ist leer oder es enthält keine Info-Datei für eine FGG. – Existiert kein SYSLOG-Directory, dann wird automatisch eine Datei „SYSLOG“ angelegt.	Bereitgestelltes SYSLOG-Directory löschen und/oder neu erzeugen.
59	Fehler beim Öffnen der SYSLOG-Datei	siehe DMS-Fehlercode, Seite 383 .
60	Fehler beim Anmelden der Anwendung oder einer Folgetask an BCAM bzw. Socket. Genaue Ursache siehe DIAGNOSTIC-WORD in der K065-Meldung (Standardziel SYSLOG) bzw. K154-Meldung (Standardziel SYSLOG). Anwenderfehler oder Systemfehler	K065 bzw. K154 in der SYSLOG-Datei auswerten
62	wie 70.	wie 70.
64	Die 1. Task der Anwendung fordert im Warmstart einen Arbeitsbereich von 8 KB an, sie bekommt ihn aber nicht.	System-Generierung und UTM-Generierung überprüfen.
65	wie 64 (Länge = Taskzahl * 2KB).	wie 64.
69	KDCFILE und ROOT passen bezüglich OSI TP-Generierung nicht zusammen. Ursache kann sein, dass eine neue KDCFILE erzeugt wurde (neu: mit OSI TP, alt: ohne OSI TP), die UTM-Anwendung aber nicht neu gebunden bzw. die ROOT-Source nicht neu übersetzt wurde.	Neu generieren, ROOT übersetzen und binden.
70	Die 1. Task der Anwendung entdeckt beim Warmstart inkonsistente Daten in KDCFILE.	Systemfehler, bitte Service informieren; KDCFILE mit KDCDEF neu generieren.
71	wie 70.	wie 70.

Code	Fehlerursache	Maßnahme
72	wie 70.	wie 70.
74	Datenbankproblem bei DB-Statusabfrage im Wiederanlauf, Ursache KU.. wird ausgegeben, siehe KCRCDC-Fehlercodes.	Datenbank in Ordnung bringen, Anwendung neu starten.
75	wie 70.	wie 70.
76	wie 70.	wie 70.
77	Eine Benutzererkennung für einen offenen Asynchron-Vorgang wurde nicht mehr gefunden	Problemmeldung schreiben.
79	Eine UTM-Task fordert Speicher an, bekommt ihn aber nicht.	System-Generierung und Speicherbedarf der Anwendung überprüfen.
80	Die 1. Task der Anwendung entdeckt beim Aufbau der Pagepool Belegungstabelle inkonsistente Daten in KDCFILE.	Systemfehler, bitte Service informieren; KDCFILE mit KDCDEF neu generieren.
81	wie 80.	wie 80.
82	Datenbankproblem beim Löschen der DB-Statusinformation, Ursache KU.. wird ausgegeben, siehe KCRCDC-Codes.	Datenbank in Ordnung bringen, Anwendung neu starten.
83	Datenbankproblem beim Rücksetzen einer DB-Transaktion, Ursache KU.. wird ausgegeben, siehe KCRCDC-Codes.	Datenbank in Ordnung bringen, Anwendung neu starten.
84	Die Task versucht, einen Slot für eine OSI-Vorgangstabelle anzulegen. Wegen Adressraumangel gelingt dies nicht.	System-Generierung und Speicherbedarf der UTM-Anwendung überprüfen.
85	Die Task versucht, einen Slot für einen offenen Vorgang anzulegen. Wegen Adressraumangel gelingt dies nicht.	System-Generierung und Speicherbedarf der UTM-Anwendung überprüfen.
86	Die 1. Task der Anwendung entdeckt freizugebende Pagepoolseiten noch im Cache (im Warmstart).	Systemfehler, PM schreiben.
90	Fehler beim Starten eines offenen verteilten Asynchron-Vorgangs	Systemfehler, PM schreiben.
91	Fehler beim Start von XAP-TP. Der Fehler wird in der Meldung K124 näher beschrieben.	Systemfehler, bitte PM schreiben und Service informieren bzw. Speicherbedarf überprüfen (siehe K124 auf Seite 265)
92	Beim Wiederanlauf für OSI TP mit XAP-TP wurde keine freie NODE-Tabelle gefunden.	Systemfehler, bitte PM schreiben und Service informieren.
93	Beim Wiederanlauf für OSI TP mit XAP-TP wurde eine Benutzererkennung für eine noch nicht beendete Transaktion nicht mehr gefunden.	Systemfehler, bitte PM schreiben und Service informieren.

Code	Fehlerursache	Maßnahme
94	Beim Wiederanlauf für OSI TP mit XAP-TP wurde eine inkonsistente NODE-Tabelle gefunden.	Systemfehler, bitte PM schreiben und Service informieren.
95	DMS-Fehler beim Lesen einer UTM- Cluster-Datei (Start-Case 1)	siehe DMS-Fehlercode
96	Eine UTM-Cluster-Datei passt nicht zur Cluster-Konfigurationsdatei; z.B. Erzeugungszeit verschieden. Dateiname siehe K082-Meldung. Ein Kurzprotokoll nach <i>stdout</i> zeigt die genaue Ursache.	Richtige UTM-Cluster-Datei einsetzen oder UTM-Cluster-Anwendung neu generieren.
97	DMS-Fehler beim Schreiben einer Cluster-Datei (Opcode CL_FIRST_NODE)	siehe DMS-Fehlercode
98	Die KDCFILE passt nicht zur Cluster-Konfigurationsdatei; z.B. Erzeugungszeit verschieden. Dateiname siehe K082-Meldung. Ein Kurzprotokoll nach <i>stdout</i> zeigt die genaue Ursache.	Richtige KDCFILE einsetzen oder KDCFILE neu generieren.
99	Datenbankproblem beim Warmstart über die XA-Schnittstelle.	Systemfehler, bitte PM schreiben und Service informieren.
100	Beim Starten der UTM-Anwendung wurde eine Inkonsistenz bzgl. der KB-Länge zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
101	Beim Start einer Folge-Task wurde eine Inkonsistenz bzgl. der KB-Länge zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
103	Beim Starten der UTM-Anwendung wurde eine Inkonsistenz bzgl. der NB-Länge zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
104	Beim Start einer Folge-Task wurde eine Inkonsistenz bzgl. der NB-Länge zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
109	Beim Starten der UTM-Anwendung wurde eine Inkonsistenz bzgl. der Generierung der Datenbank zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
110	Beim Start einer Folge-Task wurde eine Inkonsistenz bzgl. der Generierung der Datenbank zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.

Code	Fehlerursache	Maßnahme
112	Beim Starten der UTM-Anwendung wurde eine Inkonsistenz bzgl. des ersten Datenbank-Typen zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
115	Beim Starten der UTM-Anwendung wurde eine Inkonsistenz bzgl. des 2. Datenbank-Typen zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
116	Beim Start einer Folge-Task wurde eine Inkonsistenz bzgl. des 2. Datenbank-Typen zwischen KDCFILE und ROOT festgestellt.	KDCFILE und ROOT mit gleichen Generierungsparametern erzeugen.
118	UTM-Cluster-Anwendung: Fehler beim Lesen des Filebase-Namens der KDCFILE aus der Cluster-Konfigurationsdatei Mögliche Ursachen: – Speicherengpass – Fehler bei Zugriff auf Cluster-File – ungültige Cluster-File – eigener Knoten nicht gefunden Näheres siehe K043- und K190-Meldung	Siehe K043- oder K190-Meldung; Ggfs. Startparameter oder Generierung korrigieren
119	UTM-Cluster-Anwendung: Die KDCFILE wurde als UTM-Cluster-Anwendung generiert, aber es wurde kein Startparameter CLUSTER-FILEBASE angegeben.	Startparameter korrigieren
120	UTM-Cluster-Anwendung: Die KDCFILE wurde nicht als UTM-Cluster-Anwendung generiert, aber es wurde ein Startparameter CLUSTER-FILEBASE angegeben.	Startparameter korrigieren
122	UTM-Cluster-Anwendung: Fehler beim Bearbeiten der Cluster-Konfigurationsdatei zum Initialisieren des KAA	Siehe K043- und/oder K190-Meldung
123	UTM-Cluster-Anwendung: Fehler beim Registrieren des Knotens	Siehe K043- und/oder K190-Meldung
125	UTM-Cluster-Anwendung: Die Reihenfolge der Knoten-Namen in der KDCFILE ist verschieden von der in der Cluster-Konfigurationsdatei.	Generierung korrigieren; KDCFILE und ggf. UTM-Cluster-Dateien neu erzeugen
127	UTM-Cluster-Anwendung: Fehler beim Öffnen der Cluster-User-Datei beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043- und/oder K190-Meldung
128	UTM-Cluster-Anwendung: Fehler beim Öffnen der Cluster-User-Datei beim Start eines Folge-Prozesses (Start-Case 2)	Siehe K043- und/oder K190-Meldung

Code	Fehlerursache	Maßnahme
130	UTM-Cluster-Anwendung: Fehler beim Anfordern des Start-Locks zur Serialisierung durch KCSCONS (Start-Case 1)	Siehe K043- und/oder K190-Meldung
131	UTM-Cluster-Anwendung: Fehler beim Öffnen der Dateien des Administrations-Journals beim Start eines Folge-Prozesses (Start-Case 2)	Siehe K043-Meldung und K190-Meldung
132	UTM-Cluster-Anwendung: Fehler beim Öffnen der Dateien des Administrations-Journals beim Neuladen des Anwendungsprogramms nach einem Programmaustausch (Start-Case 3)	Siehe K043-/K190-Meldung
133	UTM-Cluster-Anwendung: Fehler beim Anlegen der Dateien des Administrations-Journals beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043-/K190-Meldung
134	UTM-Cluster-Anwendung: Fehler beim Schreiben der Online-Kopie beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043- und/oder K190-Meldung
135	UTM-Cluster-Anwendung: Fehler beim Einarbeiten der Online-Kopie beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043-/K190- und/oder K174-Meldung
136	UTM-Cluster-Anwendung: Fehler beim Bearbeiten der Cluster-User-Datei beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043- und/oder K190-Meldung
137	UTM-Cluster-Anwendung: Der Aufruf von KCCCTRL mit Operationscode REGISTER_NODE lieferte einen unbekanntem Returncode.	Systemfehler, bitte PM schreiben und Service informieren.
138	UTM-Cluster-Anwendung: Der Aufruf von KCCJCTL mit Operationscode WRITE_JOURNAL_PI lieferte einen schlechten Returncode.	Systemfehler, bitte PM schreiben und Service informieren.
139	UTM-Cluster-Anwendung: Sperre zur Start-Serialisierung wurde nicht angefordert.	Systemfehler, bitte PM schreiben und Service informieren.
140	UTM-Cluster-Anwendung: Fehler beim Abschluss der Registrierung des Knotens am Cluster	Siehe K043- oder K190-Meldung
141	UTM-Cluster-Anwendung: Der Aufruf von KCCJCTL mit Operationscode JFCT_SET_KAA_INFO lieferte einen schlechten Returncode	Systemfehler, bitte PM schreiben und Service informieren.

Code	Fehlerursache	Maßnahme
142	UTM-Cluster-Anwendung: Der Aufruf von KCCJCTL mit Operationscode JFCT_GET_KAA_INFO lieferte einen schlechten Returncode	Systemfehler, bitte PM schreiben und Service informieren.
143	UTM-Cluster-Anwendung: Fehler beim Öffnen der Dateien des Administrations-Journals beim Start des ersten Prozesses der Anwendung (Start-Case-1)	Siehe K043-Meldung und K190-Meldung
144	UTM-Cluster-Anwendung: Fehler beim Einarbeiten des Administrations-Journals	Systemfehler, bitte PM schreiben und Service informieren.
145	UTM-Cluster-Anwendung: Der Aufruf von KCCJCTL mit Operationscode JFCT_SET_KAA_INFO lieferte einen schlechten Returncode.	Systemfehler, bitte PM schreiben und Service informieren.
146	UTM-Cluster-Anwendung: Die Laufzeit-Konfiguration der startenden Knoten-Anwendung passt nicht zur Laufzeit-Konfiguration der laufenden Knoten-Anwendungen.	Siehe K174-Meldung
147	UTM-Cluster-Anwendung: Im Warmstart sollen globale Administrationsaktionen eingearbeitet werden. Die Dateien des Administrations-Journals können aber nicht geöffnet werden.	Prüfen, ob die Dateien des Administrations-Journals existieren, siehe K043-/K190-Meldung
148	UTM-Cluster-Anwendung: Im Warmstart sollen globale Administrationsaktionen eingearbeitet werden. Interner Fehler beim Lesen der Journaldateien.	Systemfehler, bitte PM schreiben und Service informieren.
149	UTM-Cluster-Anwendung: Interner Fehler beim Bereinigen Knoten-spezifischer Information in den Journaldateien	Systemfehler, bitte PM schreiben und Service informieren.
150	UTM-Cluster-Anwendung: Fehler beim Öffnen der Dateien des Administrations-Journals beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043-Meldung und K190-Meldung
151	UTM-Cluster-Anwendung: Fehler beim Anfordern der Sperre zur Start-Serialisierung der Knoten (im Warmstart)	Siehe K043- oder K190-Meldung
152	UTM-Cluster-Anwendung: Fehler beim Anfordern der Sperre zur Start-Serialisierung der Knoten	Siehe K043- oder K190-Meldung
153	UTM-Cluster-Anwendung: Verwaltungsdatei des Cluster-Pagepool fehlerhaft.	Siehe K190-Meldung

Code	Fehlerursache	Maßnahme
154	UTM-Cluster-Anwendung: Fehler beim Öffnen der LOCK-Datei beim Start eines Folge-Prozesses (Start-Case 2)	Siehe K043-Meldung
155	UTM-Cluster-Anwendung: Fehler beim Öffnen der LOCK-Datei beim Start des ersten Prozesses der Anwendung (Start-Case 1)	Siehe K043-Meldung
156	UTM-Cluster-Anwendung: Fehler beim Öffnen der LOCK-Datei beim Start eines Folge-Prozesses (Start-Case 3)	Siehe K043-Meldung
157	UTM-Cluster-Anwendung: Fehler beim Prüfen der Cluster-GSSB-Datei	Siehe K043- oder K190-Meldung
158	UTM-Cluster-Anwendung: Fehler beim Prüfen der Cluster-ULS-Datei	Siehe K043- oder K190-Meldung
161	UTM-Cluster-Anwendung: Fehler beim Aufheben der ULS-Locks in der Cluster-ULS-Datei	Siehe K190-Meldung
162	UTM-Cluster-Anwendung: Fehler beim Aufheben der GSSB-Locks in der Cluster-GSSB-Datei	Siehe K190-Meldung
164	UTM-Cluster-Anwendung: Fehler beim Öffnen der LOCK-Datei beim Warmstart	Siehe K043-Meldung
165	UTM-Cluster-Anwendung: Beim Warmstart einer Knoten-Anwendung konnte die Cluster-Lock-Datei nicht in der generierten Zeit gesperrt werden.	Start der Anwendung wiederholen
166	UTM-Cluster-Anwendung: Interner Fehler beim Rücksetzen einer Transaktion im Zustand PTC bei der Knoten-Recovery.	Systemfehler, bitte PM schreiben und Service informieren.
167	UTM-Cluster-Anwendung: Interner Fehler bei der Ausgabe vorhandener Transaktionen im Zustand PTC bei der Knoten-Recovery.	Systemfehler, bitte PM schreiben und Service informieren.
168	Wie 167.	Wie 167.
169	UTM-Cluster-Anwendung: Es wurde eine Knoten-Recovery für eine normal beendete Knoten-Anwendung versucht.	Prüfen, ob im Startparameter der falsche Knoten angegeben wurde.
170	UTM-Cluster-Anwendung: Cluster-Pagepool und KDCFILE mit unterschiedlicher BLKSIZE generiert.	Alle UTM-Cluster-Dateien neu generieren.
171	UTM-Cluster-Anwendung: Es wurde eine Knoten-Recovery im Dialog gestartet.	Knoten-Recovery im Batch starten.

K050 Warmstart fuer Anwendung &APPL mit UTM &VERS / &OST1 / &BMD1 erfolgreich

K051 Kaltstart fuer Anwendung &APPL mit UTM &VERS / &OST1 / &BMD1 erfolgreich

Die Meldung K050 bzw. K051 wird nach einem erfolgreichen Start der Anwendung ausgegeben. Dabei enthalten die Inserts &VERS die UTM-Version, &OST1 den Typ des Betriebssystems und &BMD1 den Bit-Modus (32/64).

Die beiden Meldungen besitzen die zusätzlichen Inserts &AMOD, &TERM, &ATYP und &FNOD, die nicht im Standardmeldungstext enthalten sind; diese haben die folgende Bedeutung:

&AMOD enthält den Application Mode der Anwendung:

"S" bei UTM-S und

"F" bei UTM-F

&TERM enthält den beim Start der Anwendung in der KDCFILE eingetragenen Termination Type; folgende Werte sind möglich:

"D": Die KDCFILE wurde neu erstellt mit KDCDEF.

"U": Die KDCFILE wurde mit KDCUPD aktualisiert.

"N": Der letzte Anwendungslauf wurde normal beendet.

"A": Der letzte Anwendungslauf wurde abnormal beendet.

&ATYP enthält :

"C" bei einer UTM-Cluster-Anwendung

"S" bei einer stand-alone-Anwendung

&FNOD enthält:

" " (Leerzeichen) bei einer stand-alone-Anwendung

"Y" beim Start der ersten Knoten-Anwendung einer UTM-Cluster-Anwendung

"N" beim Start jeder weiteren Knoten-Anwendung einer UTM-Cluster-Anwendung

K052 Start beendet - Task &PID fuer Anwendung &APPL, Version &PRGVERS aktiviert; System-Task: &STSK

Das Insert &PRGVERS ist nur bei einem Programmaustausch mit kdcprog von Bedeutung. Es gibt die Versionsnummer der FGG an, aus der das Anwendungsprogramm geladen wurde. Bei normalem Anwendungsstart enthält &PRGVERS den Wert Null.

&STSK gibt an, ob es sich um einen System-Prozess handelt (Y) oder nicht (N).

K053 Neue Userlogdatei wurde erzeugt; die alte Userlogdatei enthaelt &CNTR Saetze

K054 Die Kopien der Userlogdateien sind nicht identisch

K055 Asynchron-Vorgang &ATAC1 durch UTM beendet; KCRCCC=&RCCC; KCRCDC=&R CDC; Benutzer=&USER; LTERM=<RM

- K056** Die Task &PID wurde beendet und wird erneut gestartet: &RSLT
Das Insert &RSLT zeigt an, ob der Prozess nach PEND ER erneut gestartet wird (Y) oder nicht (N).
- K057** Der Anwendungslauf wurde beendet
- K058** Die Task &PID wurde abnormal beendet
- K059** Der Anwendungslauf wurde abnormal beendet
- K060** Der Anwendungslauf wurde abgebrochen; die Ursache ist &TRMA.

Fehlercodes im UTM-Dump

UTM erzeugt einen Speicherauszug, wenn eine UTM-Anwendung anormal beendet oder ein Speicherauszug angefordert wird, und zwar für jeden Workprozess der Anwendung (siehe [Kapitel „Der UTM-Dump“ auf Seite 59](#)). Sie finden den Fehlercode im UTM-Dump bei: REASON= auf jeder Seite des aufbereiteten UTM-Dumps in der zweiten Zeile.

Wird eine UTM-Anwendung abnormal beendet, führt openUTM - soweit möglich - vorher außerdem noch folgende Aktionen durch:

- alle Transaktionen sofort beenden, die gerade von den einzelnen Prozessen der UTM-Anwendung bearbeitet werden
- alle Verbindungen zu den Kommunikationspartnern der UTM-Anwendung abbauen
- Verbindungen zu externen Resource Managern (z.B. zu Datenbanksystemen) abbauen
- alle Dateien schließen
- alle Prozesse beenden

Aus Sicherheitsgründen wird bei einer abnormalen Beendigung einer UTM-Anwendung nicht mehr versucht, die KDCFILE, welche alle für den Ablauf einer UTM-Anwendung notwendigen Daten enthält, wieder in einen konsistenten Zustand zu bringen. Dies geschieht erst während des erneuten Starts.

Die Spalte **Grp.** (Gruppe) in der folgenden Tabelle beschreibt, welcher Ursachengruppe der Dumpcode-Fehlercode angehört. Es gibt folgende Gruppen:

- A Ursache ist ein Anwenderfehler, z.B. ein Fehler beim
- Generieren und administrieren von UTM-Anwendungen
 - Generieren des System (z.B. Aufteilung des Adressraums)
- U Ursache ist ein Fehler im UTM-Code.
- S Ursache ist ein Fehler in einer anderen Systemkomponente (Software oder Hardware).
- F Es handelt sich um einen Folgedump, ein anderer Task hat den Abbruch der Anwendung veranlasst.

- D Der UTM-Dump wurde für Diagnosezwecke erzeugt. Der UTM-Work-Prozess läuft weiter.
- M Ursache ist ein Speicherengpass.
- X Ursache ist ein Fehler im XAP-TP-Code.

Mehrfachnennungen sind möglich, z.B. ADS.

Fehler der Ursachengruppe M im XAP-TP-Baustein können u.a. auftreten, wenn bei der Generierung der Anwendung mit KDCDEF der Wert für den Parameter OSI-SCRATCH-AREA in der MAX-Anweisung zu klein gewählt wurde.

Bei allen Fehlern der Gruppen U, X und S und bei allen **nicht** in der folgenden Tabelle aufgelisteten Fehlercodes sollten Sie zur Klärung eine Problemmeldung schreiben. Für die Diagnose sind eine Reihe von Unterlagen nötig. Ausführliche Informationen mit einer Liste der benötigten Unterlagen finden Sie in [Abschnitt „Fehlerunterlagen erstellen“ auf Seite 44](#).

Sie sollten versuchen, die Fehler unter Verwendung statischer Bibliotheken zu reproduzieren

Code	Grp.	Beschreibung
AHQA00	MX	XAP-TP Baustein Modul KCOCOTA, Funktion QueueAnno(). Der Makro mGetBufferSize() lieferte den Returncode LB_NOMEM.
AHSA02	SUX	XAP-TP Baustein Modul KCOCOTA, Funktion SendAnno(). Der Aufruf des System-Interfaces KCOBRSE lieferte einen schlechten Returncode.
AINF01	S	Nach dem AINF-Aufruf ist der Returncode ungleich X'00'.
ALAxXX	ASU	Fehlernummer xxx beim Aufruf von shmat() in KCSALME
ALGxxx	ASU	Fehlernummer xxx beim Aufruf von shmget() in KCSALME. Mögliche Ursachen: Siehe Startfehler 33 oder KDCREM wurde vor Anwendungsstart nicht aufgerufen oder Shared Memory Engpass wegen zu großem CACHESIZE o. ä. Hinweis: ALG022 kann daran liegen, dass das Shared Memory, das eingerichtet werden soll, größer ist als das maschinenspezifische Maximum (Tuning Parameter SHMMAX). Maßnahme: KDCDEF Generierung ändern.
APFI00	A	FOBIB-Shellvariable konnte nicht eingerichtet werden, z.B. bei Speicherengpass Maßnahme: Speicherbedarf überprüfen.
ASA006	SUX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ATTACH" aufgerufen. Schlechter Returncode von "bBuildPAddr".

Code	Grp.	Beschreibung
ASA007	SUX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ATTACH" aufgerufen. Erster Task der Anwendung, aber OSS gibt den Returncode "NOTFIRST".
ASA009	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ATTACH" aufgerufen. OSS liefert beim "attach" Aufruf den Returncode "INVEREF".
ASA010	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ATTACH" aufgerufen. OSS liefert beim "attach" Aufruf einen schlechten Returncode.
ASA011	FSX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_DETACH" aufgerufen. OSS liefert beim "detach" Aufruf den Returncode "ERROR". Die Task ist schon in der Terminierungsphase, nachdem eine andere Task hie Anwendung abnormal beendet hat. OSS liefert einen schlechten Returncode, weil UTM eigentlich noch ein Event bei OSS abholen müsste.
ASA012	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_DETACH" aufgerufen. OSS liefert beim "detach" Aufruf den Returncode "INVAREF".
ASA013	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_DETACH" aufgerufen. OSS liefert beim "detach" Aufruf einen schlechten Returncode.
ASA033	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. Der gelieferte Application Context Name ist zu lang.
ASA034	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim "assin" Aufruf den Returncode "ERROR".
ASA035	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim "assin" Aufruf den Returncode "INVREF".
ASA036	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim "assin" Aufruf einen schlechten Returncode.
ASA043	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim positiven "assrs" Aufruf den Returncode "ERROR".
ASA044	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. Schlechter Returncode von "PutElement".

Code	Grp.	Beschreibung
ASA045	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim positiven "assrs" Aufruf einen schlechten Returncode.
ASA046	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim negativen "assrs" Aufruf den Returncode "ERROR".
ASA048	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. OSS liefert beim negativen "assrs" Aufruf einen schlechten Returncode.
ASA049	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_IND" aufgerufen. Schlechter Returncode von "PutElement".
ASA051	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. Der gelieferte Application Context Name ist zu lang.
ASA052	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert beim "asscf" Aufruf den Returncode "ERROR".
ASA053	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert beim "asscf" Aufruf den Returncode "INVREF".
ASA054	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert beim "asscf" Aufruf einen schlechten Returncode.
ASA060	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert beim "aborq" Aufruf den Returncode "ERROR".
ASA062	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert beim "aborq" Aufruf einen schlechten Returncode.
ASA064	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert einen ungültigen diagnostic Wert bei einer negativen association confirmation.
ASA065	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. OSS liefert einen ungültigen "result source" Wert bei einer negativen association confirmation.

Code	Grp.	Beschreibung
ASA066	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF_TIMEOUT" aufgerufen. OSS liefert beim "aborq" Aufruf den Returncode "ERROR".
ASA068	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF_TIMEOUT" aufgerufen. OSS liefert beim "aborq" Aufruf einen schlechten Returncode.
ASA071	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_IND" aufgerufen. OSS liefert beim "aboin" Aufruf den Returncode "ERROR".
ASA072	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_IND" aufgerufen. OSS liefert beim "aboin" Aufruf den Returncode "INVREF".
ASA073	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_IND" aufgerufen. OSS liefert beim "aboin" Aufruf einen schlechten Returncode.
ASA080	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_P_ABORT_IND" aufgerufen. OSS liefert beim "paboin" Aufruf den Returncode "ERROR".
ASA081	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_P_ABORT_IND" aufgerufen. OSS liefert beim "paboin" Aufruf den Returncode "INVREF".
ASA082	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_P_ABORT_IND" aufgerufen. OSS liefert beim "paboin" Aufruf einen schlechten Returncode.
ASA083	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. Schlechter Returncode von "PutElement".
ASA084	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF" aufgerufen. Schlechter Returncode von "PutElement".
ASA085	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ASS_CNF_TIMEOUT" aufgerufen. Schlechter Returncode von "PutElement".

Code	Grp.	Beschreibung
ASA088	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "relin" Aufruf den Returncode "ERROR".
ASA089	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "relin" Aufruf den Returncode "INVREF".
ASA090	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "relin" Aufruf einen schlechten Returncode.
ASA091	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert für "release reason" einen ungültigen Wert.
ASA092	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "aborq" Aufruf den Returncode "ERROR".
ASA094	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "aborq" Aufruf einen schlechten Returncode.
ASA095	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "relrs" Aufruf den Returncode "ERROR".
ASA097	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. OSS liefert beim "relrs" Aufruf einen schlechten Returncode.
ASA099	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_REQ" aufgerufen. OSS liefert für "abort diagnostic" einen ungültigen Wert.
ASA100	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_REQ" aufgerufen. OSS liefert beim "aborq" Aufruf den Returncode "ERROR".

Code	Grp.	Beschreibung
ASA101	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_IND" aufgerufen. Schlechter Returncode von "PutElement".
ASA102	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_REQ" aufgerufen. OSS liefert beim "aborq" Aufruf einen schlechten Returncode.
ASA104	AX	XAP-TP Baustein Die Funktion "bBuildPAddr" des Moduls "KCOASAM" wurde aufgerufen. Der Presentation Selector eines lokalen Accesspoints oder eines entfernten Partners ist zu lang.
ASA105	AX	XAP-TP Baustein Die Funktion "bBuildPAddr" des Moduls "KCOASAM" wurde aufgerufen. Der Session Selector eines lokalen Accesspoints oder eines entfernten Partners ist zu lang.
ASA116	SX	XAP-TP Baustein Die Funktion "BuildRemoteAet" des Moduls "KCOASAM" wurde aufgerufen. Der APT liegt in "form2" vor, der AEQ nicht.
ASA120	SX	XAP-TP Baustein Die Funktion "BuildRemoteAet" des Moduls "KCOASAM" wurde aufgerufen. Der APT liegt in "form1" vor, der AEQ nicht.
ASA122	SX	XAP-TP Baustein Die Funktion "BuildRemoteAet" des Moduls "KCOASAM" wurde aufgerufen. Der APT liegt weder in "form1" noch in "form2" vor.
ASA128	M	XAP-TP Baustein Die Funktion "CopyDefinedContext" des Moduls "KCOASAM" wurde aufgerufen. Schlechter Returncode von "PutElement"
ASA137	M	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_GO_IND" aufgerufen. Die Association ist nicht gesperrt.
ASA139	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_P_ABORT_IND" aufgerufen. Schlechter Returncode von "PutElement".
ASA151	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. Schlechter Returncode von "PutElement".

Code	Grp.	Beschreibung
ASA152	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_REQ" aufgerufen. Schlechter Returncode von "PutElement".
ASA153	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_REQ" aufgerufen. Schlechter Returncode von "SacfSeparator".
ASA155	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_ABORT_IND" aufgerufen. Schlechter Returncode von "SacfSeparator".
ASA156	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_P_ABORT_IND" aufgerufen. Schlechter Returncode von "SacfSeparator".
ASA157	SX	XAP-TP Baustein Der Modul "KCOASAM" wurde mit dem Operationscode "ASAM_RELEASE_IND" aufgerufen. Schlechter Returncode von "SacfSeparator".
ASIO01	AU	Datei ist als nicht geöffnet gekennzeichnet beim Aufruf von KCSASIO.
ASIO02	AU	Ungültige (d.h. zu große) Position für Schreiben in Datei beim Aufruf von KCSASIO erkannt. Zusatzinfo: Der Dateiname und die aktuelle Position wird in zusätzlicher "K078 KCSASIO ..." Meldung ausgegeben.
ASIO03	AU	Ungültiger (d.h. zu großer) Schreibauftrag für Schreiben in Datei beim Aufruf von KCSASIO erkannt. Zusatzinfo: Siehe ASIO02
ASIO04	AU	Ungültige (d.h. zu große) Position für Lesen aus Datei beim Aufruf von KCSASIO erkannt. Zusatzinfo: Siehe ASIO02
ASIO05	AU	Ungültiger (d.h. zu große) Leseauftrag für Lesen aus Datei beim Aufruf von KCSASIO erkannt. Zusatzinfo: Siehe ASIO02
ASIO06	AU	Allgemeiner Fehler bei Positionierung in einer Datei beim Aufruf von KCSASIO.
ASIS99	D	Normale Ausführung des Administrationskommandos "KDCSHUT KILL" bzw. des Aufrufs der Programmschnittstelle der Administration mit dem Opcode KC_SHUTDOWN und dem Subopcode1 KC_KILL.

Code	Grp.	Beschreibung
ATAxxx	AU	Fehler beim Anbinden eines shared-memories an den Workprozess Falls xxx der Wert für EINVALID ist, liegt ein Konflikt zwischen einem shared-memory des users und einem shared-memory der UTM-Anwendung vor. Wird ein shared object mit LOADMODE=ONCALL generiert und benutzt, kann ebenfalls ein Fehler mit dem Wert EINVALID auftreten. Maßnahme: In diesen Fällen kann der Fehler umgangen werden, indem beim Starten der Anwendung die shared-memory Überwachung ausgeschaltet wird.
BFMM05	M	UTM-Cluster-Anwendung Es konnte kein weiteres Puffer-Segment angefordert werden. Evtl. ist der virtuelle Adressraum zu klein.
BFMM21	ASU	UTM-Cluster-Anwendung Beim Aufheben einer Dateisperre kam es zu einem Timeout. Maßnahme: Siehe CCFG19
BFMM22	SU	UTM-Cluster-Anwendung Beim Aufheben einer Dateisperre wurde ein schlechter Returncode zurückgegeben.
BRSREM	F	UTM-Anwendung wurde mit Dienstprogramm KDCREM beendet.
CACHT1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
CACHT2	F	(Siehe CACHT1) Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
CACHT3	F	(Siehe CACHT1) Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
CACHT4	F	(Siehe CACHT1) Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
CACHT5	F	(Siehe CACHT1) Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
CC-...	D	Diagnosedump, der abhängig von einem primären KDICS-Returncode erzeugt wird. Nach dem Präfix CC- folgt der primäre KDICS-Returncode (z.B CC-84Z). Ein- und Ausschalten mit der Funktion Message-Dump.
CCFG07	SU	UTM-Cluster-Anwendung Modul KCCCFG, Entry KCCCGFB Schlechter Returncode von KCSGLHN
CCFG19	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_READ_CLUSTER_FILE Timeout bei Anforderung des Shared Lock. Maßnahme: Siehe Abschnitt "Maßnahme bei Sperren von UTM-Cluster-Dateien" im Anschluss an diese Tabelle.
CCFG20	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_READ_CLUSTER_FILE Schlechter Returncode bei Anforderung des Shared Lock.

Code	Grp.	Beschreibung
CCFG21	ASU	UTM-Cluster-Anwendung Modul KCCCFG Timeout bei Anforderung des Exclusive Lock. Maßnahme: Siehe CCFG19
CCFG22	SU	UTM-Cluster-Anwendung Modul KCCCFG Schlechter Returncode bei Anforderung des Exclusive Lock.
CCFG29	ASU	UTM-Cluster-Anwendung Modul KCCCFG Timeout bei Freigabe des Datei-Lock. Maßnahme: Siehe CCFG19
CCFG30	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_READ_CLUSTER_FILE Schlechter Returncode bei Freigabe des Datei-Lock.
CCFG32	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_REGISTER_COMPLETE Ungültige Version der Cluster-Konfigurationsdatei.
CCFG33	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_REGISTER_COMPLETE Korrumpierte Cluster-Konfigurationsdatei.
CCFG34	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_REGISTER_COMPLETE Ungültiger Anwendungsname in der Cluster-Konfigurationsdatei.
CCFG35	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_REGISTER_COMPLETE Die Cluster-Konfigurationsdatei wurde während des Betriebs neu erzeugt.
CCFG41	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_OPEN_CLUSTER_FILE Timeout bei Initialisierung des globalen Lock. Maßnahme: Siehe CCFG19
CCFG42	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_OPEN_CLUSTER_FILE Schlechter Returncode bei Initialisierung des globalen Lock.
CCFG43	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_CLOSE_CLUSTER_FILE Timeout bei Destroy des globalen Lock. Maßnahme: Siehe CCFG19
CCFG44	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_CLOSE_CLUSTER_FILE Schlechter Returncode bei Destroy des globalen Lock.

Code	Grp.	Beschreibung
CCFG45	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_INIT_KAA Schlechter Returncode von KCSGLHN
CCFG49	S	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE Schlechter Returncode beim Lesen der Cluster-Datei.
CCFG50	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE Korrupte Cluster-Datei.
CCFG51	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE Ungültige Version der Cluster-Konfigurationsdatei.
CCFG52	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE Ungültiger Anwendungsname in der Cluster-Konfigurationsdatei.
CCFG53	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE Die Cluster-Konfigurationsdatei wurde während des Betriebs neu erzeugt.
CCFG55	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode NODE_FAILURE Korrupte Cluster-Konfigurationsdatei.
CCFG56	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode NODE_FAILURE Ungültige Version der Cluster-Konfigurationsdatei.
CCFG57	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode NODE_FAILURE Ungültiger Anwendungsname in der Cluster-Konfigurationsdatei.
CCFG58	A	UTM-Cluster-Anwendung Modul KCCCFG, Opcode NODE_FAILURE Die Cluster-Konfigurationsdatei wurde während des Betriebs neu erzeugt.
CCFG62	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_RESET_START_SERIALIZATION Timeout bei Freigabe des Lock. Maßnahme: Siehe CCFG19
CCFG63	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_RESET_START_SERIALIZATION Schlechter Returncode bei Freigabe des Lock.
CCFG64	ASU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_RESET_START_SERIALIZATION Timeout bei Destroy des Lock. Maßnahme: Siehe CCFG19

Code	Grp.	Beschreibung
CCFG65	SU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_RESET_START_SERIALIZATION Schlechter Returncode bei Destroy des Lock.
CCFG71	AU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode READ_CLUSTER_FILE oder CHECK_CLUSTER_FILE Der lokale Knoten ist in der Cluster-Konfigurationsdatei als ausgefallen gekennzeichnet
CCFG72	AU	UTM-Cluster-Anwendung Modul KCCCFG, Opcode CCFG_NODE_FAILURE Der lokale Knoten ist in der Cluster-Konfigurationsdatei als ausgefallen gekennzeichnet.
CCKF02	SU	UTM-Cluster-Anwendung Modul KCCCKF Unerwarteter Returncode von KCCFILA beim Öffnen der KDCFILE.
CCKF03	SU	UTM-Cluster-Anwendung Modul KCCCKF Unerwarteter Returncode von KCCGFLK beim Unlock.
CCKF04	SU	UTM-Cluster-Anwendung Modul KCCCKF Unerwarteter Returncode von KCCFILA beim Schließen der KDCFILE.
CDTN02	M	XAP-TP Baustein. Modul KCOCOHF, Funktion CheckDtnidTnid(). Der Makro mGetBufferSize() lieferte den Returncode LB_NOMEM.
CFMM05	D	Beim Senden passt die Länge in DCF nicht zu der im Letterheader.
CLREST	D	UTM-Cluster-Anwendung Der Dump wird nur im Warmstart einer Knoten-Anwendung bei eingeschalt- etem Testmode erzeugt. Er dient der Diagnose nach eventuellen Fehlern im Cluster Warmstart.
COBOL6	A	Es wurde die interne NETCOBOL Funktion CBL_SETJMP innerhalb der libwork.so aufgerufen, obwohl NETCOBOL auf AIX nicht unterstützt wird.
COBOL7	A	Es wurde die interne NETCOBOL Funktion CBL_LONGJMP innerhalb der libwork.so aufgerufen obwohl NETCOBOL auf AIX nicht unterstützt wird.
COBOL8	A	Es wurde die interne Funktion coblongjump innerhalb der libwork.so aufge- rufen obwohl kein Coblol Programm generiert ist.
COBOL9	A	Es wurde die interne Funktion cobsavenv2 innerhalb der libwork.so aufge- rufen obwohl kein Coblol Programm generiert ist.
CONS03	A	Die KDCFILE wurde im laufenden Betrieb überschrieben. Mögliche Ursache: KDCDEF-Lauf in laufender Anwendung.

Code	Grp.	Beschreibung
CSND04	MX	XAP-TP Baustein Ungültiger Returncode nach Aufruf von PutElement() zum Anfordern eines dynamischen Puffers für Concatenator Sendedaten.
CSND05	SX	XAP-TP Baustein Ungültiger Returncode nach Aufruf einer OSS Presentation Funktion.
CTPF04	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
DC....	D	Diagnosedump, der abhängig von einem sekundären KDCS-Returncode erzeugt wird. Nach dem Präfix DC- folgt der sekundären KDCS-Returncode (z.B DCKS17). Ein- und Ausschalten mit der Funktion Message-Dump.
DIAGCL	D	UTM-Cluster-Anwendung Eine Knoten-Anwendung wurde abnormal beendet. Vor der Beendigung wurden, sofern noch möglich, alle anderen Knoten-Anwendung über das abnormale Ende informiert. Die informierten Knoten-Anwendungen schreiben einen Diagnose-Dump.
DIAGDP	D	Mit dem Administrationskommando "KDCDIAG DUMP=YES" bzw. durch den Aufruf der Programmschnittstelle der Administration mit Opcode KC_CREATE_DUMP wurde ein Diagnosedump erzeugt.
DMCA00	M	XAP-TP Baustein Modul KCOCODM, Funktion ConnectDynMemArea(). Die Funktion ConnectSharedMem() lieferte den Returncode MEM_NOMEM.
EHHP00	M	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Der Returncode von mGetBufferSize() war ungleich LB_OK.
EHHP01	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_datain() lieferte den Returncode P_ERROR.
EHHP02	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_datain() lieferte den Returncode P_INVREF.
EHHP03	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die oss Funktion p_datain() lieferte einen unbekanntem Returncode.
EHHP04	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_typein() lieferte den Returncode P_ERROR.
EHHP05	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_typein() lieferte den Returncode P_INVREF.

Code	Grp.	Beschreibung
EHHP06	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_typein() lieferte einen unbekanntem Returncode.
EHHP07	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Nach Aufruf der OSS Funktion p_typein() liefert die Funktion CalcUserDataLth() als Länge der User Daten 0.
EHHP08	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synin() liefert den Returncode P_ERROR.
EHHP09	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synin() lieferte den Returncode P_INVREF.
EHHP10	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synin() lieferte einen unbekanntem Returncode.
EHHP12	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HamdlePresEvent(). Nach Aufruf der OSS Funktion p_synin() liefert die Funktion CalcUserDataLth() als Länge der User Daten Null.
EHHP13	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synfc() liefert den Returncode P_ERROR.
EHHP14	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synfc() lieferte den Returncde P_INVREF.
EHHP15	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_synfc() lieferte einen unbekanntem Returncode.
EHHP17	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Nach Aufruf Der OSS Funktion p_synfc() lieferte die Funktion CalcUserDataLth() die User Daten Länge Null.
EHHP18	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkgin() lieferte den Returncode P_ERROR.
EHHP19	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkgin() lieferte den Returncode P_INVREF.

Code	Grp.	Beschreibung
EHHP20	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkgin() lieferte einen unbekanntem Returncode.
EHHP21	S	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkgin() lieferte nicht das Token S_T_MINOR.
EHHP22	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkpin() lieferte den Returncode P_ERROR.
EHHP23	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkpin() lieferte den Returncode P_INVREF.
EHHP24	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_tkpin() lieferte einen unbekanntem Returncode.
EHHP25	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die Funktion p_tkpin() lieferte nicht das Token S_T_MINOR.
EHHP26	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Nach Aufruf der OSS Funktion p_tkpin() lieferte die Funktion CalcUserDataLth() als User Daten Länge den Wert Null.
EHHP27	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minin() lieferte den Returncode P_ERROR.
EHHP28	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minin() lieferte den Returncode P_INVREF.
EHHP29	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minin() lieferte einen unbekanntem Returncode.
EHHP30	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Nach Aufruf der OSS Funktion p_minin() lieferte die Funktion CalcUserDataLth() die User Daten Länge Null.
EHHP31	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minfc() lieferte den Returncode P_ERROR.
EHHP32	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minfc() lieferte den Returncode P_INVREF.

Code	Grp.	Beschreibung
EHHP33	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion p_minfc() lieferte einen unbekanntem Returncode.
EHHP34	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Nach Aufruf der OSS Funktion p_minfc() liefert die Funktion CalcUserDataLth() den Wert Null als User Daten Länge.
EHHP35	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die Funktion o_event() lieferte das unerwartete Ereignis P_MAJIN.
EHHP36	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die Funktion o_event() lieferte das unerwartete Ereignis P_MAJCF.
EHHP37	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion HandlePresEvent(). Die OSS Funktion o_event() lieferte einen unbekanntem Ereignistyp.
EHRP01	MX	XAP-TP Baustein. Modul KCOXFEH, Funktion ReloadPresEvent(). Der Makro mGetBufferSize() lieferte einen Returncode ungleich LB_OK.
EHSP00	SX	XAP-TP Baustein. Modul KCOXFEH, Funktion StorePresEvent(). Es soll mittels StorePresEvent() ein Presentation Ereignis im DynMem gespeichert werden. Für die betreffende Association ist jedoch schon ein Event gespeichert (mValLth(g, &pAss->PendingEvnt.h) > 0).
EHSP01	M	XAP-TP Baustein. Modul KCOXFEH, Funktion StorePresEvent(). Die Funktion PutElement() lieferte den Returncode DM_NOMEM. Maßnahme: In der KDCDEF-Generierung die Größe der OSI-Scratch-Area erhöhen (Parameter MAX OSI-SCRATCH-AREA).
EKAA11	S	UTM-Cluster-Anwendung Timeout beim Freigeben der Datei-Sperre für die JKAA-Datei. Maßnahme: Siehe CCFG19
EKAA12	S	Schlechter Returncode von KCCGFLK beim Freigeben der Datei-Sperre für die JKAA-Datei.
EKAA13	S	UTM-Cluster-Anwendung Timeout beim Zerstören des Datei-Locks für die JKAA-Datei. Maßnahme: Siehe CCFG19
EKAA14	S	Schlechter Returncode von KCCGFLK beim Zerstören des Datei-Locks für die JKAA-Datei.

Code	Grp.	Beschreibung
ENCERR	DSU	UTM hat eine verschlüsselte Nachricht erwartet, aber eine unverschlüsselte empfangen. Es liegt kein Abbruch der Anwendung vor. Der Dump dient allein der Diagnose.
ENDE01	SU	Der KAA-Pointer im KTA zeigt auf keinen KAA.
ENDE02	AU	Unterbrechung bei gesperrtem KAA.
ENDE03	ASU	Abnormale Prozess-Beendigung während des Datenbank-Aufrufs zwischen dem Schreiben eines Confirmatory Record und dem (vorläufigen) Ende der UTM-Transaktion.
ENDE04	SU	Der Cache kann nicht freigegeben werden
ENDE05	SU	Der KAA kann nicht freigegeben werden.
ENDE06	ASU	Fehler beim Öffnen der SYSLOG-Datei.
ENDE11	DSU	Die Anwendung wurde normal beendet, während der Abmeldung der Accesspoints von OSS wurde von OSS ein Returncode ungleich OK geliefert. Wenn der Testmode eingeschaltet ist, wird ein UTM Diagnosedump gezogen.
ENDE12	ASU	Eine Task hat sich beendet (normal oder abnormal). Die verbliebenen Tasks reichen nicht aus, damit die Anwendung weiterarbeiten kann (z.B. weil sich Tasks im PGWT befinden).
ENDE14	F	Eine Task hat sich beendet (normal oder abnormal) und die Task hat noch nicht KCSTRMA aufgerufen. Der Status der Anwendung ist jedoch TERM_APPL. Um zu erreichen, dass bei abnormalem Anwendungsende alle Tasks einen UTM-Dump erzeugen, wurde KCSTRMA mit ENDE14 aufgerufen.
ENDE19	ASU	Es laufen nur noch System-Tasks für die Anwendung. Die normalen Tasks wurden alle beendet. Ursache ist entweder ein Fehler in der Startprozedur oder ein Startfehler beim Nachstarten der Tasks nach PEND ER oder Programmaustausch.
ENDPET	A	Eine UTM-D-Anwendung kann nicht normal beendet werden, weil es noch Vorgänge mit Transaktionsstatus PTC (prepare to commit) gibt oder weil für asynchrone Nachrichten an andere Anwendungen noch keine Quittungen eingetroffen sind. ES wird kein UTM-Dump erzeugt. UTM führt beim nächsten Start der Anwendung einen Warmstart durch.
EVGE00	M	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Der Returncode des Makros mGetBufferSize() war ungleich LB_OK.
EVGE01	S	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Nach dem Aufruf der OSS Funktion o_event() enthielt der Rückgabeparameter <o_echain> einen ungültigen Wert.

Code	Grp.	Beschreibung
EVGE03	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte als Returncode den Wert O_ACSE und es wurden noch nicht alle User Daten beim Aufruf empfangen.
EVGE05	ASX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_ERROR.
EVGE06	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_INVEREF.
EVGE07	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_TIMEINT.
EVGE08	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_WAKEINT.
EVGE09	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_SYSTEM.
EVGE10	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_LOOK.
EVGE11	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_TRANSPORT.
EVGE12	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte den Returncode O_SESSION.
EVGE13	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion GetOssEvent(). Die OSS Funktion o_event() lieferte einen unbekanntem Returncode.
EVNT03	ASX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_ERROR. Dies kann auch passieren wenn "BCAM" beendet wird ("BCEND").
EVNT04	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_INVEREF.

Code	Grp.	Beschreibung
EVNT05	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_TIMEINT.
EVNT06	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_WAKEINT.
EVNT07	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_SYSTEM.
EVNT08	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_TRANSPORT.
EVNT09	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_SESSION.
EVNT10	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_PRESENTATION.
EVNT11	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte den Returncode O_ACSE.
EVNT12	SX	XAP-TP Baustein. Modul KCOXFEV, Funktion aputm_event(). Die OSS Funktion o_event(O_EVALLOOK) lieferte einen unbekanntenen Returncode.
EXIT00	AU	Unerlaubter exit()-Aufruf wird während des Ablaufs von UTM-Systemcoding erkannt.
EXPI32	ASU	UTM-Cluster-Anwendung Schlechter Returncode von KCCGFLK beim Sperren der Cluster-GSSB-Datei. Maßnahme: Siehe CCFG19
EXPI34	ASU	UTM-Cluster-Anwendung Schlechter Returncode von KCCGFLK beim Sperren der Cluster-ULS-Datei. Maßnahme: Siehe CCFG19.

Code	Grp.	Beschreibung
EXPI35	ASU	UTM-Cluster-Anwendung Schlechter Returncode von KCCGFLK beim Sperren der Cluster-ULS-Datei. Maßnahme: Siehe CCFG19
EXPI36	ASU	UTM-Cluster-Anwendung Schlechter Returncode von KCCGFLK beim Sperren der GSSB-Datei. Maßnahme: Siehe CCFG19
EXPI51	A	Ein Periodic Write wurde von einem Task aufgerufen, da die Conf.Area des Tasks voll geworden ist. Der Task will eine weitere Conf.-Seite schreiben, aber durch den Periodic Write wurden keine Seiten frei. Ursache kann eine zu klein generierte Conf. Area, so dass die Wiederanlaufinformation einer einzigen Transaktion größer ist als die Conf. Area (z.B. Transaktion mit Zugriffen auf "sehr viele" Page-Pool Seiten). Maßnahme: in der Generierungsanweisung "MAX RECBUF=(anzahl,...) den Parameter anzahl hochsetzen.
EXPI95	A	Siehe EXPI51, tritt aber in RESET_TA_RTN auf.
FHCV03	S	Formatierungssystem wird nicht unterstützt.
FMMM10	A	Eine Eingabenachricht kann nicht abgespeichert werden, da der Pagepool voll ist. Maßnahme: Pagepool vergrößern, dazu in KDCDEF-Generierung MAX PGPOOL=(anzahl,...) größer machen.
FMSM05	DSU	KCDFMSM wurde mit Operations-code PRSP_BID aufgerufen, obwohl Session-state nicht gleich WAIT_OF_BID_RSP.
FMSM12	S	Unzulässiger LSES_WORK_STATE beim Empfang von RTR.
FREE01	AX	XAP-TP Baustein. Modul KCOXFFO, Funktion ap_free(). Es sind mehr als APFREE_MAX_TO_REL Speicherbereiche freizugeben.
GETR00	AU	Recordlänge <= Null beim Aufruf von KCSGETR Mögliche Ursache: ungültige Eingabe-Datei
GETR01	AU	Beim Lesen des Satzlängenfeldes eines Records wurden weniger als 4 Bytes eingelesen in KCSGETR. Mögliche Ursache: ungültige Eingabe-Datei
GETR02	AU	Gelesener Record hat eine Satzlänge < 5 nach dem Lesen des Satzlängenfelds in KCSGETR. Mögliche Ursache: ungültige Eingabe-Datei
GETR03	AU	Beim Lesen eines Records wurden weniger Bytes eingelesen als nach dem Lesen des Satzlängenfelds in KCSGETR erwartet. Mögliche Ursache: ungültige Eingabe-Datei
GETR04	AU	Beim Lesen eines Records mit KCSGETR wurde ein ungültiger Record erkannt. Mögliche Ursache: ungültige Eingabe-Datei

Code	Grp.	Beschreibung
GFLKT1	F	UTM-Cluster-Anwendung Lock Hierarchie nach Term Application verletzt (Folge-Dump).
GFLKT2	F	UTM-Cluster-Anwendung Siehe GFLKT1 (Folge-Dump).
GLHN03	A	Lokaler Rechnername länger als acht Zeichen.
GMDT11	SU	UTM-Cluster-Anwendung Modul KCCGMDT, Opcode UPDATE_AND_UNLOCK_GSSB. Unerwarteter Returncode von KCCGFLK (lock GSSB Datei). Maßnahme: Siehe CCFG19
GSYS00	S	XAP-TP Baustein Modul KCOCOHF, Funktion GetSystemInfo(). die Funktion uname() lieferte einen negativen Returncode.
INPERR	AD	Fehler im INPUT-Exit wurde erkannt.
IOyxxx	ASU	ASIO Return Code: Ein Fehler bei der Dateibearbeitung ist aufgetreten und konnte nicht korrigiert werden, yxxx = DMS-Fehlercode. Maßnahme: Siehe Kapitel "Fehlercodes bei Dateibearbeitung"
IPC000	ASU	IPC Shared Memory Segment ist überschrieben worden. Bei jedem KDCS-Aufruf findet eine Überprüfung statt.
IPC035	A	Fehler beim Sperren des IPC Shared Memory Segments, Ursache kann kdcrem während der laufenden Anwendung sein.
IPC037	FU	An der IUTMIPC wird erkannt, dass die Anwendung abnormal beendet wird (= Folge-Dump). Siehe auch U306 mit Insert UERRNO=37.
IPCEND	F	Ein Workprozess hat sich nicht unter Kontrolle von openUTM beendet. Die Anwendung wird deshalb beendet. Mögliche Ursache: Siehe Hinweis zu Meldung U231
IPCREM	F	UTM-Anwendung wurde mit Dienstprogramm KDCREM beendet.
ISLP11	SU	Protokoll Fehler.
ISLP12	AU	Taskspezifischer Puffer für Wiederanlaufinformation ist zu klein. Maßnahme: In KDCDEF-Generierung MAX RECBUF=(...,länge) vergrößern.
ISLPT1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
ISLPT4	F	Siehe ISLPT1
JFC011	ASU	UTM-Cluster-Anwendung Timeout bei KCCGFLK-Aufruf in KCCJFCT mit Opcode CLOSE_FILES. Maßnahme: Siehe CCFG19
JFC012	ASU	UTM-Cluster-Anwendung Fehler bei KCCGFLK-Aufruf in KCCJFCT mit Opcode CLOSE_FILES. Maßnahme: Siehe CCFG19

Code	Grp.	Beschreibung
JFC016	SU	UTM-Cluster-Anwendung Fehler bei 1. readControlPage-Aufruf in KCCJFCT mit Opcode CHECK_UNPROC_ENTRIES.
JFC018	SU	UTM-Cluster-Anwendung Fehler bei 2. readControlPage-Aufruf in KCCJFCT mit Opcode CHECK_UNPROC_ENTRIES.
JFC055	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT mit Opcode SET_NODE_PROCESSING_STATE.
JFC058	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode SET_NODE_PROCESSING_STATE.
JFC067	SU	UTM-Cluster-Anwendung Fehler beim Lesen der Verwaltungsseite in KCCJFCT mit Opcode SET_NODE_CR_SEQNR.
JFC071	SU	UTM-Cluster-Anwendung Fehler beim ersten readControlPage-Aufruf in KCCJFCT mit Opcode GET_NODE_CR_SEQNR.
JFC072	SU	UTM-Cluster-Anwendung Fehler beim zweiten readControlPage-Aufruf in KCCJFCT mit Opcode GET_NODE_CR_SEQNR.
JFC076	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT mit Opcode RESET_NODE_CR_SEQNR.
JFC082	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode RESET_DYNADM_LOCK.
JFC083	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT mit Opcode RESET_DYNADM_LOCK.
JFC400	ASU	UTM-Cluster-Anwendung Timeout bei KCCGFLK-Aufruf(Lock) in KCCJFCT, interne Funktion setFileLock. Maßnahme: Siehe CCFG19
JFC401	ASU	UTM-Cluster-Anwendung Fehler bei KCCGFLK-Aufruf(Lock) in KCCJFCT, interne Funktion setFileLock. Maßnahme: Siehe CCFG19
JFC402	SU	UTM-Cluster-Anwendung Fehler bei KCCGFLK-Aufruf(Lock) in KCCJFCT, interne Funktion setFileLock.

Code	Grp.	Beschreibung
JFC404	SU	UTM-Cluster-Anwendung Timeout bei KCCGFLK-Aufruf(Unlock) in KCCJFCT, interne Funktion releaseFileLock.
JFC405	ASU	UTM-Cluster-Anwendung Fehler bei KCCGFLK-Aufruf(Unlock) in KCCJFCT, interne Funktion release-FileLock. Maßnahme: Siehe CCFG19
JFC420	SU	UTM-Cluster-Anwendung Fehler bei KCCFILA-Aufruf(Write) 1. Journaldatei-Verwaltungsseite in KCCJFCT, interne Funktion initControlPages.
JFC421	SU	UTM-Cluster-Anwendung Fehler bei KCCFILA-Aufruf(Write) in KCCJFCT, interne Funktion initControlPages.
JFC430	SU	UTM-Cluster-Anwendung Fehler beim ersten readControlPage-Aufruf in KCCJFCT, interne Funktion switchFiles.
JFC431	SU	UTM-Cluster-Anwendung Fehler beim zweiten readControlPage-Aufruf in KCCJFCT, interne Funktion switchFiles.
JFC460	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT, interne Funktion check-ControlPages.
JFC501	SU	UTM-Cluster-Anwendung Fehler bei KCCFILA-Aufruf in KCCJFCT, interne Funktion readPages.
JFC507	SU	UTM-Cluster-Anwendung Fehler bei KCCFILA-Aufruf in KCCJFCT, interne Funktion writePages.
JFC512	SU	UTM-Cluster-Anwendung Fehler bei KCCFILA-Aufruf in KCCJFCT, interne Funktion writeControlPage.
JFC550	SU	UTM-Cluster-Anwendung Fehler beim Lesen einer Verwaltungsseite in KCCJFCT mit Opcode GET_JOURNAL_FILE_INFO.
JFC561	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT mit Opcode SET_COPY_STATE.
JFC570	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT mit Opcode SET_KAA_INFO.
JFC572	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode SET_KAA_INFO.

Code	Grp.	Beschreibung
JFC580	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT mit Opcode GET_KAA_INFO.
JFC581	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode GET_KAA_INFO.
JFC583	SU	UTM-Cluster-Anwendung Fehler beim 3. readControlPage-Aufruf in KCCJFCT mit Opcode GET_KAA_INFO.
JFC584	SU	UTM-Cluster-Anwendung Fehler beim 4. readControlPage-Aufruf in KCCJFCT mit Opcode GET_KAA_INFO.
JFC590	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT mit Opcode SET_GLOBAL_ADM_LOCK.
JFC591	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode SET_GLOBAL_ADM_LOCK.
JFC594	SU	UTM-Cluster-Anwendung Fehler beim 3. readControlPage-Aufruf in KCCJFCT mit Opcode SET_GLOBAL_ADM_LOCK.
JFC595	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT mit Opcode RESET_GLOBAL_ADM_LOCK.
JFC596	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode RESET_GLOBAL_ADM_LOCK.
JFC598	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT mit Opcode GET_GLOBAL_ADM_LOCK.
JFC599	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT mit Opcode GET_GLOBAL_ADM_LOCK.
JFC611	SU	UTM-Cluster-Anwendung Fehler beim 2. readControlPage-Aufruf in KCCJFCT, Routine checkAllProcessed.
JFC613	SU	UTM-Cluster-Anwendung Fehler beim 1. readControlPage-Aufruf in KCCJFCT, Routine checkAllProcessed.
JFC626	SU	UTM-Cluster-Anwendung Fehler beim readControlPage-Aufruf in KCCJFCT, Routine checkAndRepair.

Code	Grp.	Beschreibung
LATC01	FSU	In dem Zeitpunkt des Aufrufs von KCSLATC ist der Lock-Mechanismus kaputt. Siehe LATC02.
LATC02	ASU	Fehler bei Aufruf von KCXLOCK der ATC-Lock-Börse. Da bei openUTM in Unix-/Windows-Systemen die Locks mit Semaphoren realisiert werden, wurde ein Fehlercode bei einer Semaphor Operation gemeldet. Die Ursache kann sein, dass der Benutzer ein Semaphor der Anwendung gelöscht hat (entweder durch Unix-/Windows-Systemfunktion oder durch KDCREM).
LATC03	SU	Max. Anzahl Locks pro Task überschritten.
LATC04	SU	KCSLATC wurde unter IPC-Lock aufgerufen.
LATCT1	F	Beim Anfordern der Sperre steht der Status der Applikation auf TERM APPLICATION. Der Task beendet sich dadurch ebenfalls (= Folge-Dump).
LCAC01	FSU	In dem Zeitpunkt des Aufrufs von KCSLKAA / KCSLCAC / KCSPCMM ist der Lock-Mechanismus kaputt. Siehe LCAC02.
LCAC02	ASU	Fehler bei Aufruf von KCXLOCK der KAA/CACH/PCMM-Lock-Börse. Da bei openUTM in Unix-/Windows-Systemen die Locks mit Semaphoren realisiert werden, wurde ein Fehlercode bei einer Semaphor Operation gemeldet. Die Ursache kann sein, dass der Benutzer ein Semaphor der Anwendung gelöscht hat (entweder durch Unix-/Windows-Systemfunktion oder durch KDCREM).
LCAC03	SU	Max. Anzahl Locks pro Task überschritten.
LCAC04	SU	KCSLCAC wurde unter IPC-Lock aufgerufen.
LCACT1	F	Beim Anfordern der Sperre steht der Status der Applikation auf TERM APPLICATION. Der Task beendet sich dadurch ebenfalls (= Folge-Dump).
LKAA01	FSU	In dem Zeitpunkt des Aufrufs von KCSLKAA ist der Lock-Mechanismus kaputt. Siehe LKAA02.
LKAA02	ASU	Fehler bei Aufruf von KCXLOCK der KAA/CACH/PCMM-Lock-Börse. Da bei openUTM in Unix-/Windows-Systemen die Locks mit Semaphoren realisiert werden, wurde ein Fehlercode bei einer Semaphor Operation gemeldet. Die Ursache kann sein, dass der Benutzer ein Semaphor der Anwendung gelöscht hat (entweder durch Unix-/Windows-Systemfunktion oder durch KDCREM).
LKAA03	SU	Max. Anzahl Locks pro Task überschritten.
LKAA04	SU	KCSLKAA wurde unter IPC-Lock aufgerufen.
LKAAT1	F	Beim Anfordern der Sperre steht der Status der Applikation auf TERM APPLICATION. Der Task beendet sich dadurch ebenfalls (= Folge-Dump).

Code	Grp.	Beschreibung
LKLC26	FU	Bei Sperren eines Eintrags wird erkannt, dass der Vorgang bereits in einer Warteschlangenverkettung eingetragen ist. Systemfehler oder Folge-Dump, wenn die Anwendung gerade abnormal beendet wird (durch Fehler oder KDCSHUT KILL). Der Fall Folge-Dump ist daran zu erkennen, dass die Ursache des Abbruchs in Meldung K060 nicht LKLC26 ist, und bereits ein weiterer Dump erzeugt wurde.
LKLC42	AU	Börsen-Wartezeit abgelaufen. Maßnahme: KDCDEF-Generierung ändern, Parameter RESWAIT = (...zeit2) in MAX-Anweisung erhöhen. Eine Ursache für die Zeitüberschreitung kann auch das Erzeugen von Diagnoseinformation (gcore) unter Lock in einem anderen Workprozess sein.
LKLC51	AU	Siehe LKLC42.
LKLC64	AU	Siehe LKLC42.
LKLCT1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump)
LKLCT2	F	Siehe LKLCT1
LKLCT3	F	Siehe LKLCT1
LKLCT4	F	Siehe LKLCT1
LKMTC0	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump vor Aufruf von KCSBRSE).
LKMTC1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump nach Aufruf von KCSBRSE).
LKMTKP	D	Dieser Diagnosedump wird im Fall von TESTMODE=ON erzeugt, wenn beim unlock das PEND KP Flag im Lockfeld nicht zurückgesetzt ist.
LPCM01	FSU	In dem Zeitpunkt des Aufrufs von KCSPCMM ist der Lock-Mechanismus kaputt. Siehe LPCM02.
LPCM02	ASU	Fehler bei Aufruf von KCXLOCK der PCMM-Lock-Börse. Da bei openUTM in Unix-/Windows-Systemen die Locks mit Semaphoren realisiert werden, wurde ein Fehlercode bei einer Semaphor Operation gemeldet. Die Ursache kann sein, dass der Benutzer ein Semaphor der Anwendung gelöscht hat (entweder durch Unix-/Windows-Systemfunktion oder durch KDCREM).
LPCM03	SU	Max. Anzahl Locks pro Task überschritten.
LPCM04	SU	KCSLPCM wurde unter IPC-Lock aufgerufen.
LPCMT1	F	Beim Anfordern der Sperre steht der Status der Applikation auf TERM APPLICATION. Der Task beendet sich dadurch ebenfalls (= Folge-Dump).
LWRT02	SU	Fehler beim Öffnen des Benutzerlogfiles <filebase>/USLA/yyyy (Kopie A).
LWRT03	SU	Fehler beim Öffnen des Benutzerlogfiles <filebase>/USLB/yyyy (Kopie B).

Code	Grp.	Beschreibung
LWRT04	ASU	Fehler beim Positionieren in Benutzerlogfile <filebase>/USLA/yyyy. Evtl. wurden Schreibmanipulationen an der aktuellen Dateigeneration vorgenommen oder die Base geändert. Zum Umschalten während laufender Anwendung KDCLOG-Kommando benutzen. Maßnahme, falls der LWRT04 beim nächsten Anwendungsstart wieder auftritt: Die Benutzerprotokolldatei, d.h. das gesamte USLA Verzeichnis muss gelöscht und wieder neu erzeugt werden.
LWRT05	ASU	Fehler beim Positionieren in Benutzerlogfile <filebase>/USLB/yyyy. Grund kann gleich sein wie bei LWRT04.
LWRT06	SU	Fehler beim Positionieren in Benutzerlogfile <filebase>/USLB/yyyy beim Rückpositionieren auf Dateianfang nach einem nicht erfolgreich abgeschlossenen Schreibauftrag (z.B. wegen Plattenspeicherengpass).
LWRT07	SU	Fehler beim Positionieren in Benutzerlogfile <filebase>/USLA/yyyy beim Rückpositionieren auf Dateianfang nach einem nicht erfolgreich abgeschlossenen Schreibauftrag (z.B. wegen Plattenspeicherengpass).
LWRT08	SU	Fehler beim PUT-Aufruf zum Schreiben eines LPUT-Records auf den Benutzerlogfile <filebase>/USLA/yyyy
LWRT09	SU	Fehler beim PUT-Aufruf zum Schreiben eines LPUT-Records auf Benutzerlogfile <filebase>/USLB/yyyy
LWRT10	SU	Fehler beim Schließen eines Benutzerlogfile <filebase>/USLB/yyyy
LWRT11	SU	Fehler beim Schließen eines Benutzerlogfile <filebase>/USLA/yyyy
LWRT17	A	KCSLWRT fordert über KCSALME einen Puffer zum Schreiben auf den Benutzerlogfile an, der Puffer kann nicht bereitgestellt werden. Maßnahme: Speicherbedarf und Betriebssystem-Generierung überprüfen.
LWRT19	SU	Fehler beim Positionieren ans Ende der Datei für Kopie A des Benutzerlogfiles.
LWRT20	SU	Siehe LWRT19, jedoch für Kopie B.
MACF02	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MACF03	M	XAP-TP Baustein Der Returncode von SetTimer() war ungleich TI_OK.
MACF04	M	XAP-TP Baustein Der Returncode von GetLogRecord war ungleich MACF_OK.
ME....	D	Diagnosedump, der abhängig von einer bestimmten UTM-Meldung erzeugt wird. Nach dem Präfix ME folgt die Meldungsnummer der UTM-Meldung (z.B. MEK135). Ein- und Ausschalten mit der Funktion Message-Dump.
MFCR04	M	XAP-TP Baustein Der Returncode von GetLogRecord war MACF_NO_MEM.

Code	Grp.	Beschreibung
MFCR07	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR08	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR09	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR10	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR11	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR16	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR17	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR18	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR19	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR20	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR21	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFCR24	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFDM03	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFDM04	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFDM05	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFDM06	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM05	AX	XAP-TP Baustein Beim TP_RECOVER_REQ ist kein freier Dialogtabelleneintrag für einen Transaktionszweig vorhanden. Mögliche Ursache: die Anzahl der Associations im vorherigen Anwendungslauf war größer als die Anzahl Associations im aktuellen Anwendungslauf.
MFRM06	AX	XAP-TP Baustein Siehe MFRM05.

Code	Grp.	Beschreibung
MFRM07	AX	XAP-TP Baustein Siehe MFRM05.
MFRM08	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSize() war ungleich LB_OK.
MFRM09	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM10	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM11	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM12	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM13	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM14	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM15	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM16	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM17	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM18	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM19	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM21	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM24	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFRM25	AX	XAP-TP Baustein Kein freier Tabelleneintrag für einen Log-Damage-Record vorhanden. Maßnahme: Log-Damage-Records mit TP_UPDATE_LOG_DAMAGE_REQ löschen oder nMaxLogDamRec auf einen größeren Wert setzen.
MFT102	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSize() war ungleich LB_OK.
MFT103	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.

Code	Grp.	Beschreibung
MFT104	M	XAP-TP Baustein Der Returncode von ChangeDescriptor war ungleich DM_OK.
MFT105	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT106	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT107	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT108	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFT109	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT110	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT111	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT113	M	XAP-TP Baustein Der Returncode von ChangeDescriptor war ungleich DM_OK.
MFT114	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT115	M	XAP-TP Baustein Der Returncode von ChangeDescriptor war ungleich DM_OK.
MFT119	M	XAP-TP Baustein Der Returncode von GetLogRecord() war ungleich MACF_OK.
MFT120	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT121	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT122	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT123	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT124	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT126	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT127	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.

Code	Grp.	Beschreibung
MFT128	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT129	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT130	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT131	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT132	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT133	M	XAP-TP Baustein Der Returncode von CopyElement war ungleich DM_OK.
MFT134	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT135	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT138	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT139	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT141	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFT142	M	XAP-TP Baustein Die Funktion CopyElement hat einen Returncode ungleich DM_OK geliefert.
MFT147	M	XAP-TP Baustein Die Funktion PutElement hat einen Returncode ungleich DM_OK geliefert.
MFT151	M	XAP-TP Baustein Die Funktion PutElement hat einen Returncode ungleich DM_OK geliefert.
MFTP03	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFTP04	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP05	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFTP06	M	XAP-TP Baustein Der Returncode von PutElement war ungleich DM_OK.
MFTP07	M	XAP-TP Baustein Der Returncode von SetTimer war ungleich TI_OK.

Code	Grp.	Beschreibung
MFTP10	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP11	M	XAP-TP Baustein Der Returncode von RequestBuffer() war ungleich LB_OK.
MFTP12	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP15	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP16	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP17	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP18	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP19	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP20	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MFTP24	M	XAP-TP Baustein Der Returncode vom Makro mGetBufferSpace() war ungleich LB_OK.
MOVE03	AU	Ungültige Überlappung von Ziel und Quellbereich in KCSMOVE bzw. KCSMOVEP.
MSG000	DU	KCSCRMS wurde mit einer ungültigen Message-Id aufgerufen. Es wird ein Diagnose-Dump geschrieben und die Meldung "K000 MESSAGE NOT DEFINED" ausgegeben.
NDRCVY	D	Der Dump wird nur am Ende einer Knoten-Recovery bei eingeschaltetem Testmode erzeugt. Er dient der Diagnose nach eventuellen Fehlern nach einer Knoten-Recovery.
NET001	M	Fehler beim Anfordern von Speicherbereich für die Bcamappl-Tabelle im Netzprozess.
NET002	M	Fehler beim Anfordern von Speicherbereich für die Verbindungs-Tabelle im Netzprozess.
NET003	M	Fehler beim Anfordern von Speicherbereich für die Application-Tabelle im Netzprozess.
NET006	ASU	Mindestens ein Bcamappl konnte im Netzprozess nicht beim Transportsystem angemeldet werden.
NET007	M	Fehler beim Anfordern von Speicherbereich für die Kindprozess-Tabelle in utmnetm.
NET011	SU	Fehler beim Registrieren der CMX callback Routine in utmnet.

Code	Grp.	Beschreibung
NET014	SU	Fehler beim Aufruf von t_conin() im Netzprozess.
NET019	SU	Ungültiges DATAGO Event vom Transportsystem empfangen im Netzprozess.
NET020	SU	Fehler beim Aufruf von t_event() im Netzprozess.
NET022	SU	Fehler beim Aufruf von t_info() im Netzprozess.
NET023	SU	Fehler beim Aufruf von t_datarq() im Netzprozess.
NET031	SU	Fehler beim Initialisieren der Thread Attribute im Netzprozess.
NET032	SU	Fehler beim Setzen des Detach Attributes für den Thread im Netzprozess.
NET033	SU	Fehler beim Erzeugen des Threads für den gemeinsamen Wartepunkt im Netzprozess.
NET034	SU	Fehler beim Erzeugen des Empfangs-Socket für den gemeinsamen Wartepunkt im Netzprozess.
NET035	SU	Fehler beim Setzen des Socket Attributes REUSEADDR für Empfangs-Socket im Netzprozess.
NET036	SU	Fehler beim bind() Aufruf für Empfangs-Socket im Netzprozess.
NET037	SU	Fehler beim Ermitteln des Listenerports für Empfangs-Socket im Netzprozess.
NET038	SU	[W] Fehler beim Initialisieren der Socket Umgebung im Netzprozess.
NET039	A	[W] Auf dem Rechner ist nicht die für den Netzprozess benötigte Version der Socket Bibliothek installiert.
NET051	SU	Ungültiges Event empfangen vom Transportsystem in utmnetm.
NMTE00	M	XAP-TP Baustein Modul KCOCOHF, Funktion NewMemTabEntry(). Die Funktion RequestBuffer() lieferte den Returncode LB_NOMEM.
NMTE02	M	XAP-TP Baustein Modul KCOCOHF, Funktion NewMemTabEntry(). Der Makro mGetBufferSpace() lieferte den Returncode LB_NOMEM.
NRDBER	A	Die Knoten-Recovery muss wegen Fehler bei der Datenbank Recovery abnormal beendet werden. Es wird kein UTM-Dump erzeugt.
ODIA00	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssDiagInfo(). Die OSS Funktion o_error() lieferte den Returncode O_ERROR.
ODIA01	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssDiagInfo(). Die OSS Funktion o_error() lieferte den Returncode O_INVEREF.
ODIA02	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssDiagInfo(). Die OSS Funktion o_error() lieferte einen ungültigen Returncode.

Code	Grp.	Beschreibung
OREA00	SX	XAP-TP Baustein. Die OSS Funktion o_reason() lieferte den Returncode O_ERROR.
OREA01	SX	XAP-TP Baustein. Die OSS Funktion o_reason() lieferte den Returncode O_INVEREF.
OREA02	SX	XAP-TP Baustein. Die OSS Funktion o_reason() lieferte einen unbekanntem Returncode.
OSAFT2	F	Returncode APEXT_TERMAPPL von XAP-TP.
OSGO01	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssGo(). Die OSS-Funktion o_go() lieferte den Returncode O_INVCREF.
OSGO02	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssGo(). Die OSS-Funktion o_go() lieferte den Returncode O_ERROR.
OSGO03	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssGo(). Die OSS-Funktion o_go() lieferte einen unerwarteten Returncode.
OSTM01	AS	Schwerwiegender Fehler in Zusammenarbeit mit dem DB-System.
OSTM05	AU	Taskspezifischer Puffer für Wiederanlaufinformation ist zu klein. Maßnahme: in KDCDEF-Generierung MAX RECBUF=(...,laenge) vergrößern.
OSTM06	AS	Schwerwiegender Fehler in Zusammenarbeit mit dem DB-System.
OSTM07	A	Ein Log-Record kann nicht gesichert werden, da der Pagepool voll ist. Maßnahme: Pagepool vergrößern, dazu in KDCDEF-Generierung MAX PGPOOL=(anzahl,...) vergrößern.
OSTP01	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssStop(). Die OSS-Funktion o_stop() lieferte den Returncode O_INVCREF.
OSTP02	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssStop(). Die OSS-Funktion o_stop() lieferte den Returncode O_ERROR.
OSTP03	SX	XAP-TP Baustein Modul KCOCOHF, Funktion OssStop(). Die OSS-Funktion o_stop() lieferte einen unerwarteten Returncode.
PCMM05	AU	Bei KCSPCM mit dem Operationscode Get Pagechain ist der Parameter NR_PAGES ungültig oder größer als Anzahl aller Pagepoolseiten. Maßnahme: In KDCDEF-Generierung MAX PGPOOL=number erhöhen
PCTR00	M	XAP-TP Baustein Modul KCOCOHF, Funktion PrepareCtrlReq(). Der Makro mGetBufferSize() lieferte den Returncode LB_NOMEM.

Code	Grp.	Beschreibung
PEND02	A	Für "System PEND ER" kann kein TACB mehr geschrieben werden, da der Pagepool auf KDCFILE voll ist. Maßnahme: Pagepool vergrößern, dazu in KDCDEF-Generierung MAX PGPOOL=(anzahl,...) vergrößern.
PEND03	AS	Es wurde ein schwerwiegender Fehler vom DB-System gemeldet, siehe DIAGAREA im UTM-Dump. Dieser Fehler kann auftreten, wenn beim Datenbanksystem Oracle in der RMXA-Anweisung der Parameter SesTm auf einen Wert kleiner wie die KDCDEF Parameter zur PEND KP Überwachung gesetzt werden.
PEND04	S	Das DB-System lieferte einen unerlaubten Returncode, siehe DIAGAREA im UTM-Dump.
PEND05	AS	Siehe PEND03
PEND07	ASU	Signal zwischen Schreiben eines Confirmatory Record und dem (vorläufigen) Ende der UTM-Transaktion (KCSEXPI-Aufruf END_TA). Mögliche Ursache: Fehler im DB-Connection Modul. Ein Timer-Signal (SIGALRM) während dieser Zeit wird ignoriert.
PEND11	S	Inkonsistente XID
PEND26	A	Bei "System PEND ER" für einen Socket-Client kann keine K017-Meldung mehr für einen evtl. später notwendigen KDCDISP geschrieben werden, da der Pagepool auf KDCFILE voll ist. Maßnahme: Wie bei PEND02.
PEND97	A	Bei "System PEND ER" für einen Socket-Client kann keine K017-Meldung mehr für einen evtl. später notwendigen KDCDISP geschrieben werden, da der Cluster Pagepool voll ist. Maßnahme: Cluster Pagepool vergrößern, dazu in KDCDEF-Generierung CLUSTER PGPOOL=(anzahl,...) vergrößern.
PEND98	A	Für "System PEND ER" kann kein TACB mehr geschrieben werden, da der Cluster Pagepool voll ist. Maßnahme: Cluster Pagepool vergrößern, dazu in KDCDEF-Generierung CLUSTER PGPOOL=(anzahl,...) vergrößern.
PENDER	ADU	Ursache ist nicht ein Abbruch der Anwendung, sondern PEND ER des Benutzers, UTM-interner PEND ER bei KCRCCC >= 70Z oder ein Signal.
PENDT1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump vor Aufruf von KCSBRSE).
PENDT2	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump nach Aufruf von KCSBRSE).
PENDT3	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump nach Aufruf durch KDCROOT mit PEND ER)

Code	Grp.	Beschreibung
PIPE39	AU	Fehler beim Einrichten einer Instanz einer Named Pipe auf Windows-Systemen. Mögliche Ursache: Es gibt zwei gleichnamige UTM Anwendungen auf dem Rechner.
PLCA00	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ActivateCmxCallback(). Die CMX-Funktion t_callback() lieferte den Returncode T_ERROR.
PLCC00	SX	XAP-TP Baustein Modul KCOXFPL, Funktion CmxCallback(). Die Systemfunktion select() lieferte einen unerwarteten Returncode.
PLCC01	S	XAP-TP Baustein Modul KCOXFPL, Funktion CmxCallback(). Die Systemfunktion select() lieferte einen unbekanntem Returncode.
PLCD00	SX	XAP-TP Baustein Modul KCOXFPL, Funktion DeactivateCmxCallback(). Die CMX-Funktion t_callback() lieferte den Returncode T_ERROR.
PMIO20	ASU	Integrity IDs der Seite sowohl in Original wie auch in Duplikat File inkonsistent. Mögliche Ursache: Überschreiben der KDCFILE-Dateien im laufenden Betrieb durch Kopieren in Startprozedur. Maßnahme in diesem Fall: Startprozedur korrigieren Wenn dies als Ursache ausscheidet, PM schreiben. Als Diagnoseunterlagen sind erforderlich: UTM-Dump, KDCFILE-Dateien, Startprozedur und Task-/Prozess-Protokoll. Bei Cluster-Anwendung auch Cluster-Pagepool-Dateien.
PMIO22	ASU	Der Page Type, angegeben in der Cache Control Table, und der Page Type im Header der gelesenen Seite stimmen nicht überein. Mögliche Ursache: Siehe PMIO20 Maßnahme: Siehe PMIO20
PMIO23	ASU	Zu lesende bzw. schreibende Pagepoolseite ist nicht allokiert. Ursache und Maßnahme: Siehe PMIO20
PMIO32	SU	UTM-Cluster-Anwendung Timeout beim Sperren einer Datei des Cluster Pagepools. Maßnahme: Siehe Abschnitt "Maßnahme bei Sperren von UTM-Cluster-Dateien" im Anschluss an diese Tabelle.
POLL03	MX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Der Returncode des Makros mGetBufferSize() war ungleich LB_OK.
POLL05	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). die OSS-Funktion o_event() meldet ein ACSE Event und es wurden noch nicht alle User Daten beim o_event() empfangen.

Code	Grp.	Beschreibung
POLL07	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_ERROR.
POLL08	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_INVEREF.
POLL09	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die Oss-Funktion o_event() lieferte den Returncode O_TIMEINT.
POLL10	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_WAKEINT.
POLL11	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_SYSTEM.
POLL12	SX	XAP-TP Baustein Modul LCOXFPL, Funktion ap_poll(). Die OSS-Funktion ap_poll() lieferte den Returncode O_LOOK.
POLL13	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_TRANSPORT.
POLL14	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_SESSION.
POLL15	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). Die OSS-Funktion o_event() lieferte den Returncode O_SHUTDOWN.
POLL16	SX	XAP-TP Baustein Modul KCOXFPL, Funktion ap_poll(). die OSS-Funktion o_event() lieferte einen unerwarteten Returncode.
PPMM10	ASU	UTM-Cluster-Anwendung Timeout in Routine lockFile beim Versuch, eine Cluster Pagepool Datei zu sperren. Maßnahme: Siehe CCFG19
PPMM12	SU	UTM-Cluster-Anwendung Timeout in Routine unlockFile beim Versuch, eine Cluster Pagepool Datei zu entsperren. Maßnahme: Siehe CCFG19
PPMM14	SU	UTM-Cluster-Anwendung Schlechter Returncode von KCSBFMM in Routine readControlPage.

Code	Grp.	Beschreibung
PPMM15	ASU	UTM-Cluster-Anwendung Ungültiger Seiten-Header der Verwaltungsseite des Cluster Pagepools in Routine readControlPage.
PPMM41	SU	UTM-Cluster-Anwendung Fehler beim Lesen oder Schreiben der ersten CONS-Seite einer Cluster Pagepool Datei.
PPMM42	SU	UTM-Cluster-Anwendung Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers.
PUTR01	AU	In KCSPUTR konnten nicht genug Bytes geschrieben werden Mögliche Ursache: Plattenspeicherengpass
PWRT03	AMU	KCSPWRT hat (über KCSALME) Speicher verlangt, bekommt ihn aber nicht. Maßnahme: Speicherbedarf und Betriebssystem-Generierung überprüfen.
PWRT06	F	Bei KCSPWRT wurde während des Periodic Write von einem anderen UTM-Task der Anwendung term application aufgerufen (= Folgedump).
RALC01	AU	Die KDCFILE wurde z.B. mit KDCDEF überschrieben, während die Anwendung noch lief (Anwenderfehler). Index ist außerhalb des erlaubten Bereichs beim Aufruf von KCSRALC
RCV009	MX	XAP-TP Baustein. Modul KCOXFRV, Funktion ap_rcv(). Der Returncode der Funktion CopyElement() war ungleich DM_OK.
RCV012	MX	XAP-TP Baustein. Modul KCOXFRV, Funktion ap_rcv(). Inkonsistenz in den booleschen Variablen <bSwitchToNextTtnid> und <bClearTtnid>.
REME01	SU	Unzulässiger System-Returncode beim Aufruf von KCSREME
RESTRT	D	Der Dump wird nur im Warmstart bei eingeschaltetem Testmode erzeugt. Er dient der Diagnose nach eventuellen Fehlern im Warmstart.
RQOB00	M	XAP-TP Baustein Modul KCOCOHF, Funktion ReqOssInBuff(). Die Funktion RequestBuffer() lieferte den Returncode LB_NOMEM.
RVCS03	M	XAP-TP Baustein. Modul KCOXFRV, Funktion CheckSaRetc(). Der von der Funktion SetAttribute() gelieferte Returncode war SA_NOMEM.
SACT14	MX	XAP-TP Baustein Ungültiger Returncode nach Aufruf von PutElement() zum Anfordern eines dynamischen Speicherbereichs für SACF Aktion COPY.
SACT27	M	XAP-TP Baustein Ungültiger Returncode nach Aufruf von PutElement() zum Abspeichern in SACF Aktion FLUSHALL oder FLUSHPAR.

Code	Grp.	Beschreibung
SACT28	M	XAP-TP Baustein Ungültiger Returncode nach Aufruf von ReAllocElement() oder PutElement() in SACF Aktion QUEUE. Maßnahme: Den Wert des KDCDEF Parameters MAX OSI-SCRATCH-AREA erhöhen.
SC0005	SU	Die Half-Session ist als PLU generiert. Ein RU-Request ist gekommen, aber der Request-Code ist nicht SDT, STSN oder CLEAR.
SC0006	SU	Die Half-Session ist als PLU generiert. Eine RU-Response ist gekommen, aber der Request-Code ist nicht RQR.
SC0007	SU	#FSM.SESS ist nicht im Zustand "ACTIVE"
SC0008	SU	Die Half-Session ist als SLU generiert. Ein RU-Request ist gekommen, aber der Request-Code ist nicht RQR.
SC0009	SU	Die Half-Session ist als SLU generiert. Eine RU-Response ist gekommen, aber der Request-Code ist nicht SDT,STSN oder CLEAR.
SC0010	SU	Für mindestens eine der FSMs #FSM_DT, #FSM_STSN oder #FSM_RQR liegt ein Protokollfehler vor.
SC0011	SU	Mindestens eine der FSMs #FSM_DT oder #FSM_CNTL_IMMED_EXP setzt einen Sense-Code.
SC0014	SU	FSM_SESS_LU_LU hat einen unerlaubten Returncode zurückgegeben.
SC0019	ADU	Protokollfehler von FSM gemeldet. Maßnahme: In der KDCDEF-Generierung überprüfen, ob beide Partner als VTV-Partner generiert sind.
SC0020	ADU	Der Zustand der PLU-FSM ist nicht erlaubt. Maßnahme: In der KDCDEF-Generierung überprüfen, ob beide Partner als VTV-Partner generiert sind.
SC0021	ADU	Der Zustand der SLU-FSM ist nicht erlaubt. Maßnahme: In der KDCDEF-Generierung überprüfen, ob beide Partner als VTV-Partner generiert sind.
SC0023	DSU	Bei Überprüfung von UNBIND-RSP für FSM_SESS_LU_LU ist ein Fehler gefunden worden.
SC0024	DSU	Beim Schalten von UNBIND-RSP für FSM_SESS_LU_LU ist ein Fehler aufgetreten.
SC0025	DSU	Bei Überprüfung von UNBIND-Request für FSM_SESS_LU_LU ist ein Fehler gefunden worden.
SC0026	DSU	Beim Schalten von UNBIND-Request für FSM_SESS_LU_LU ist ein Fehler aufgetreten (Diagnosedump).
SC0027	DSU	FSM_STSN_SEND meldet S,RQ,STSN.
SC0028	DSU	Unerlaubte STSN-REQ für PLU_TO_SLU.
SC0029	DSU	Unerlaubte STSN-REQ für SLU_TO_PLU.

Code	Grp.	Beschreibung
SC0030	DSU	ACT_SC_RECEIVE konnte keinen Returncode setzen.
SC0032	A	Die Session kann nicht aktiviert werden, da keine freien Connections zur Verfügung stehen. Maßnahme: KDCDEF-Generierung ändern, mehr CON-Anweisungen aufnehmen.
SC0034	DSU	Der Partner hat beim STSN-Response einen Protokollfehler gemacht.
SDCS02	M	XAP-TP Baustein. Modul KCOXFSD, Funktion CheckSaRetc(). Der SetAttribute() Returncode war SA_NOMEM.
SDUI01	M	XAP-TP Baustein. Modul KCOXFSD, Funktion UserDataIn(). Der Makro mGetBufferSize() lieferte den Returncode LB_NOMEM.
SEND05	SU	Schwerer BCAM-Fehler bei der Verarbeitung eines BCAM-SENDLET-Calls. Fehler beim Canceln eines Timers für Output-Terminals.
SG-nnn	D	Diagnosedump, der abhängig von einem Returncode KCRSIGN1/2 nach einem SIGN ON Aufruf erzeugt wird. Nach dem Präfix SG- folgt der Returncode (z.B SG-U17). Ein- und Ausschalten mit der Funktion Message-Dump.
SHCxxx	SU	Fehlernummer xxx beim Aufruf von shmctl() in KCSSHMF
SHM002	A	Es wurde versucht, ein Shared Memory zu generieren, das schon vorhanden ist Maßnahme: UTM Generierung überprüfen
SIGND1	DU	Zum Zeitpunkt des Abmelden eines Benutzers, der über eine Socket-Verbindung angemeldet war, ist die Eingabe-Nachricht nicht gelöscht.
SIGxxx	AU	Signal mit Signalnummer xxx ist aufgetreten. Hinweis: Beendet sich die Anwendung mit SIG010/SIG011 (SIGBUS/SIGSEGV), kann die Ursache falsche oder fehlende Parameter beim Aufruf von KDCS sein. Ob dies der Fall ist, kann anhand der Stack-Traces im gcore-Dump unter Unix-Systemen bzw. Minidumps unter Windows-Systemen festgestellt werden. In dieser Situation ist die Funktion KCSAVAL auf dem Stack zu finden.
SLOG05	SU	Beim Versuch, die Message-Buffer zu sperren, hat KCSLKLC einen schlechten Returncode geliefert (eventuell TIMEOUT).
SLOG06	SU	Beim Versuch, die Message-Buffer zu entsperren, hat KCSLKLC einen schlechten Returncode geliefert.
SLOG07	SU	Beim Versuch, die SYSLOG-Datei zu sperren, hat KCSLKLC einen schlechten Returncode geliefert (eventuell TIMEOUT).
SLOG08	SU	Beim Versuch, die SYSLOG-Datei zu entsperren, hat KCSLKLC einen schlechten Returncode geliefert.

Code	Grp.	Beschreibung
SLOG09	SU	Das Schreiben des Message-Buffers in die aktuelle SYSLOG-Datei ist schiefgegangen (eventuell gibt der DMS-Fehlercode in der vorausgegangenen K043-Meldung Aufschluss über die Fehlerursache).
SLOG10	SU	Der Versuch eines Folgetasks, auf die gerade aktuelle SYSLOG-Dateigeneration umzuschalten, ist fehlgeschlagen (eventuell vorausgegangene K043-Meldungen beachten).
SLOG21	ASU	Beim Administrationsaufruf KDCSLOG INFO bzw. beim Aufruf der Programmschnittstelle der Administration mit Opcode KC_SYSLOG und Subopcode1 KC_INFO hat KCSSLOG die Komponente KCSFST aufgerufen, um die Attribute der Dateigenerationsgruppe der SYSLOG zu ermitteln (nur wenn die SYSLOG als FGG angelegt ist). KCSFST lieferte einen DMS-Fehler (bitte auch die vorangegangene K043-Meldung beachten).
SLOT07	AM	In KCSSLOT muss die Adresse eines Slot berechnet werden. Der Task kennt den Shared Memory noch nicht, in dem der Slot liegt. Der Versuch, sich an den Shared Memory anzuschließen, misslingt wegen Adressraumangel.
SMSG00	AU	SMSG00 ist ein Diagnose-Dump und führt nicht zum Ende des Anwendungslaufs; ein SMSG00-Dump wird nur gezogen, wenn der Testmode eingeschaltet ist. Ein SMSG-Dump wird erstellt, wenn KCSSMSG eine Meldung an ein Meldungsziel vom Typ MSG-DEST schicken möchte und dabei von KCSFPUT eine schlechten Returncode erhält. Der Returncode kann der Tabelle NKB des UTM-Dumps entnommen werden; auf Distanz X'5C' in dieser Tabelle finden sich der CC- und der DC-Code des DPUT-Aufrufs. Ein möglicher Grund kann eine Sperre der Queue/des TACs sein.
SMSG03	ASU	Die Komponente KCSSLOG lieferte beim Schreiben einer Meldung in die SYSLOG-Datei einen schlechten Returncode. Ursache siehe DMS-Returncode in K043-Meldung
SMSG09	D	SMSG09 ist ein Diagnose-Dump und führt nicht zum Ende des Anwendungslaufs; ein SMSG09-Dump wird nur gezogen, wenn der Testmode eingeschaltet ist. Bitte Systemdienst informieren.
SND005	M	XAP-TP Baustein. Modul KCOXFSD, Funktion ap_snd(). Die Funktion UserDataIn() lieferte den Returncode UDIN_NOMEM.
SND007	MX	XAP-TP Baustein. Modul KCOXFSD, Funktion ap_snd(). Der Returncode der Funktion GetVarLthAttr() war ungleich GA_OK.
STnnnn	ADSU	Fehler in der Start-Verarbeitung einer UTM-Task. Dabei ist nnnn die Nummer, die in der UTM-Meldung "K049 Fehler <nnnn> bei Start der Anwendung" die Fehlerursache anzeigt.
STPF10	SX	XAP-TP Baustein Ungültiger Indikator für HeurRep Parameter in TP-HEUR-REP-RI.

Code	Grp.	Beschreibung
STPF11	SX	XAP-TP Baustein Ungültiger Indikator für FuUnits Parameter in TP-BEG-DIAL(dialogue)-RC.
STPF12	SX	XAP-TP Baustein Ungültiger Indikator für Result Parameter in TP-BEG-DIAL(dialogue)-RC.
STPF13	SX	XAP-TP Baustein Ungültiger Indikator für Diagnostic Parameter in TP-BEG-DIAL(dialogue)-RC.
STPF14	SX	XAP-TP Baustein Ungültiger Indikator für user data Parameter in TP-BEG-DIAL(dialogue)-RC.
STPF15	SX	XAP-TP Baustein Ungültiger Indikator für Result Parameter in TP-BEG-DIAL(channel)-RC.
STPF16	SX	XAP-TP Baustein Ungültiger Indikator für Diagnostic Parameter in TP-BEG-DIAL(channel)-RC.
STPF17	SX	XAP-TP Baustein Ungültiger Indikator für HeurRep Parameter in TP-(ABORT-AND-)HEUR-REP-RI.
STPT02	SX	XAP-TP Baustein Ungültiger Wert in p_udt.p_udtnxt.
STPT10	SX	XAP-TP Baustein Ungültiger Indikator für InitTPSUT Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT11	SX	XAP-TP Baustein Ungültiger Indikator für RectPSUT Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT12	SX	XAP-TP Baustein Ungültiger Indikator für FuUnits Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT13	SX	XAP-TP Baustein Ungültiger Indikator für BegTrans Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT14	SX	XAP-TP Baustein Ungültiger Indikator für Conf Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT15	SX	XAP-TP Baustein Ungültiger Indikator für LastPartId Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT16	SX	XAP-TP Baustein Ungültiger Indikator für user data Parameter in TP-BEG-DIAL(dialogue)-RI.
STPT17	SX	XAP-TP Baustein Ungültiger Indikator für FuUnits Parameter in TP-BEG-DIAL(channel)-RI.
STPT18	SX	XAP-TP Baustein Ungültiger ChanUtil Parameter in TP-BEG-DIAL(channel)-RI.
STPT19	SX	XAP-TP Baustein Ungültiger Indikator für LastPartId Parameter in TP-BEG-DIAL(channel)-RI.
STPT20	SX	XAP-TP Baustein Ungültiger Indikator für CcrTokReq Parameter in TP-BID-RI.

Code	Grp.	Beschreibung
STPT21	SX	XAP-TP Baustein Ungültiger Indikator für LastPartId Parameter in TP-BID-RI.
STPT22	SX	XAP-TP Baustein Ungültiger Indikator für Result Parameter in TP-BID-RC.
STPT23	SX	XAP-TP Baustein Ungültiger Indikator für Conf Parameter in TP-END-DIAL-RI.
STPT24	SX	XAP-TP Baustein Ungültiger Indikator für ConfUrg Parameter in TP-HSK-RI.
STPT25	SX	XAP-TP Baustein Ungültiger Indikator für ConfUrg Parameter in TP-HSK-AND-GRT-CTRL-RI.
STPT26	SX	XAP-TP Baustein Ungültiger Indikator für Type Parameter in TP-DEFER-RI.
STPT27	SX	XAP-TP Baustein Ungültiger Indikator für Reason Parameter in TP-TOKEN-GIVE-RI.
STPT28	SX	XAP-TP Baustein Ungültiger Indikator für Correlator Parameter in TP-TOKEN-GIVE-RI.
STRT01	DU	Beim Austausch des Anwendungsprogrammms ist die Anzahl der Prozesse, die noch austauschen müssen, kleiner 0, d.h. der Zähler ist inkonsistent. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
SYPM01	AU	Taskspezifischer Puffer für Wiederanlaufinformation ist zu klein. Maßnahme: in KDCDEF-Generierung MAX RECBUF=(...,länge) vergrößern.
SYPM03	AS	Schwerwiegender Fehler in Zusammenarbeit mit dem DB-System (FITA).
SYPM11	AU	Taskspezifischer Puffer für Wiederanlaufinformation ist zu klein. Maßnahme: in KDCDEF-Generierung MAX RECBUF=(...,länge) vergrößern.
TADR03	SU	Die sogenannte Application-Reference ist im Announcement ungültig.
TC0040	DSU	Die vom Transportsystem gelieferte Eingabenachrichtenlänge stimmt nicht mit der in der Nachricht enthaltenen Länge überein.
TC0041	DSU	Die vom Transportsystem gelieferte Eingabenachrichtenlänge ist kürzer als der Wert im Data Count Field (DCF) im Transmission Header (TH).
TCPI13	SU	Ungültige Länge der IP-Adresse (TCP_GET_HOST_BY_ADDR).
TIMR02	SU	Fehler beim Sperren des IUTMIPC in KCSTIMR
TIMR03	SU	Fehler beim Freigeben des IUTMIPC in KCSTIMR
TIMR05	AU	Prozess utmtimer ist nicht vorhanden beim Senden an IUTMIPC in KCSTIMR
UATC02	SU	(Siehe UKAA02) Unlock misslungen.
UATC03	SU	(Siehe UKAA03) Max. Anzahl Locks pro Task überschritten.

Code	Grp.	Beschreibung
UCAC02	SU	(Siehe UKAA02) Unlock misslungen.
UCAC03	SU	(Siehe UKAA03) Max. Anzahl Locks pro Task überschritten.
UKAA02	SU	Unlock misslungen.
UKAA03	SU	Max. Anzahl Locks pro Task überschritten.
UMDT12	SU	KCCUMDT Opcode UMDT_LOCK_ULS Schlechter Returncode von KCSBFMM beim Lesen einer ULS-Verwaltungsseite
UMDT13	SU	KCCUMDT Opcode UMDT_LOCK_ULS Schlechter Returncode von KCCLKMT
UMDT15	ASU	UTM-Cluster-Anwendung KCCUMDT Opcode UMDT_UNLOCK_ULS Fehler beim Sperren der ULS-Datei. Maßnahme: Siehe Abschnitt "Maßnahme bei Sperren von UTM-Cluster-Dateien" im Anschluss an diese Tabelle.
UMDT32	SU	UTM-Cluster-Anwendung KCCUMDT Funktion readHdrPage Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers
UMDT33	ASU	UTM-Cluster-Anwendung KCCUMDT Funktion releaseFileLock Timeout beim Entsperrern der Cluster-ULS-Datei. Maßnahme: Siehe UMDT15
UMDT34	SU	UTM-Cluster-Anwendung KCCUMDT Funktion releaseFileLock Schlechter Returncode von KCCGFLK beim Entsperrern der Cluster-ULS-Datei. Maßnahme: Siehe UMDT15
UMDT42	SU	UTM-Cluster-Anwendung KCCUMDT Funktion addrUlsEntry Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers
UMDT44	SU	UTM-Cluster-Anwendung KCCUMDT Funktion setFileLock Schlechter Returncode von KCCGFLK beim Sperren der Cluster-ULS-Datei. Maßnahme: Siehe UMDT15
UMDT52	ASU	UTM-Cluster-Anwendung KCCUMDT Funktion allocateUlsEntry Schlechter Returncode von KCCUSF beim Sperren der Cluster-USER-Datei. Maßnahme: Siehe UMDT15

Code	Grp.	Beschreibung
UMDT63	ASU	UTM-Cluster-Anwendung CUMDT Entry KCCUGLI Die Cluster-ULS-Datei kann nicht gesperrt werden. Maßnahme Siehe UMDT15
UMDT73	SU	UTM-Cluster-Anwendung KCCUMDT Funktion checkFile Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers
UMDT74	SU	UTM-Cluster-Anwendung KCCUMDT Funktion checkFile Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers
UMDT77	SU	UTM-Cluster-Anwendung KCCUMDT Funktion checkFile Schlechter Returncode von KCSBFMM beim Anfordern eines Puffers
UPCM02	SU	(Siehe UKAA02) Unlock misslungen.
UPCM03	SU	(Siehe UKAA03) Max. Anzahl Locks pro Task überschritten.
USF002	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode OPEN_FILE Timeout beim Initialisieren des globalen Datei-Locks. Maßnahme: In der KDCDEF-Generierung den Wert des Parameters FILE-LOCK-TIMER-SEC bzw. FILE-LOCK-RETRY der CLUSTER-Anweisung erhöhen.
USF003	SU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode OPEN_FILE Ungültiger Returncode beim Initialisieren des globalen Datei-Locks
USF009	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode CUSF_RESET_PTC_FLAG Fünfmal Timeout beim Sperren der Cluster-User-Datei. Maßnahme: Siehe USF002
USF013	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode CUSF_SIGNOFF_USER_WITH_CONTEXT Fünf Mal Timeout beim Sperren der Cluster-User-Datei. Maßnahme: Siehe USF002
USF018	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode CLOSE_FILE Timeout beim Zerstören des Datei-Locks. Maßnahme: Siehe USF002
USF019	SU	UTM-Cluster-Anwendung Modul KCCUSF, Opcode CLOSE_FILE Ungültiger Returncode von KCCGFLK beim Zerstören des Datei-Locks

Code	Grp.	Beschreibung
USF021	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion setFileLock Ungültiger Returncode von KCCGFLK beim Anfordern des exklusiven Locks
USF023	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion releaseFileLock Timeout bei der Freigabe des Datei-Locks. Maßnahme: Siehe USF002
USF024	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion releaseFileLock Ungültiger Returncode von KCCGFLK bei der Freigabe des Datei-Locks
USF025	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion readPage Timeout bei der Anforderung des shareable Datei-Locks. Maßnahme: Siehe USF002
USF026	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion readPage Ungültiger Returncode von KCCGFLK bei Anforderung des shareable Datei-Locks
USF027	ASU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion readPage Timeout bei der Freigabe des Datei-Locks. Maßnahme: Siehe USF002
USF028	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion readPage Ungültiger Returncode von KCCGFLK bei der Freigabe des Datei-Locks
USF051	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion writePage Fehler beim Schreiben der Cluster-User-Datei
USF052	SU	UTM-Cluster-Anwendung Modul KCCUSF, Funktion readPage Fehler beim Lesen der Cluster-User-Datei
WAI102	DU	Diagnose-Dump
WAI112	DU	Beim Umschalten der System-Protokoll-Datei ist die Anzahl der Prozesse, die noch umschalten müssen, kleiner 0, d.h. der Zähler ist inkonsistent. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
WAI113	DU	Beim Austausch des Anwendungsprogrammms ist die Anzahl der Prozesse, die noch austauschen müssen, kleiner 0, d.h. der Zähler ist inkonsistent. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
WAI114	DU	Siehe WAI113

Code	Grp.	Beschreibung
WAI118	DU	Aktion PROC_ACCESS_FROM_USER_INDX: Ein UTM_INTERNAL_MSG_ANNO wurde empfangen aber es ist kein UTM_INTERNAL_MSG_ANNO, UTM_START_PROG_ANNO, UTM_ROLLBACK_PET_ANNO und kein UTM_COMMIT_PET_ANNO. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
WAI119	DU	Aktion PROC_ACCESS_FROM_USER_INDX: Ein UTM_INTERNAL_MSG_ANNO wurde empfangen aber es ist kein UTM_START_PROG_ANNO, UTM_ROLLBACK_PET_ANNO und kein UTM_COMMIT_PET_ANNO. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
WAI120	DU	Beim Verwalten der Diagnose-Trace-Einstellungen ist die Anzahl der Prozesse, die Verwaltung noch aktualisieren müssen, kleiner 0, d.h. der Zähler ist inkonsistent. Bei eingeschaltetem Testmodus wird die Anwendung abnormal beendet, ansonsten wird eine Diagnose-Dump geschrieben.
WAIT01	SU	Schlechter Returncode beim Aufruf von KCSBRSE für die Workbourse im Aktionsblock ENQUEUE_WORKBOURSE von KCSWAIT.
WAIT02	DS	Ungültiger Timer-Announcement Typ (Aktionsblock ENQUEUE_WORKBOURSE).
WAIT03	S	Ungültiger Typ im Announcement und kein Timer-Announcement (Aktionsblock ENQUEUE_WORKBOURSE).
WAIT28	SU	Schlechter Returncode vom Modul KCSSLOG (vor dem ENQUEUE_WORKBOURSE).
WAIT38	S	KCSWAIT startet eine TU-Aktion für einen Vorgang, der im PGWT steht.
WAIT55	SU	Ungültiger Code für ein internes Timer-Announcement (Aktionsblock ANALYSE_INTERNAL_TIMER).
WAIT78	ASU	Anwenderfehler oder schwerer Fehler beim Aufruf der internen BCAM-Schnittstelle für Socket. Mögliche Ursachen: 1. Es wurde ein aktiver Verbindungsaufbau für einen BCAMAPPL versucht, der vom Netzprozess nicht angemeldet werden konnte, weil z.B. die Portnummer für diesen lokalen Endpunkt bereits benutzt wird oder weil es sich um eine reservierte Portnummer handelt. Maßnahme: Sicherstellen, dass die Portnummer nicht anderweitig benutzt wird bzw. keine reservierte Portnummer angeben. 2. Der für die Verbindung zuständige Netzprozess hat sich beendet. 3. Einer der Netzprozesse der UTM-Anwendung hat sich abnormal beendet.
WAITT1	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump vor Aufruf von KCSBRSE).

Code	Grp.	Beschreibung
WAITT2	F	Eine andere Task hat die Anwendung abnormal beendet (= Folge-Dump nach Aufruf von KCSBRSE).
XATT02	F	XAP-TP Baustein. Modul KCOXFEX, Funktion apext_att(). Die Funktion bCheckAndSetState() lieferte einen schlechten Returncode und der Anwendungsstatus war WAITING_DUMP_APPL. Eine andere Task hat bereits den Abbruch der Anwendung veranlasst.
XATT04	MX	XAP-TP Baustein. Modul KCOXFEX, Funktion apext_att(). Der Returncode der Funktion EstablishBuffer() war ungleich LB_OK.
XATT12	MX	XAP-TP Baustein. Modul KCOXFEX, Funktion apext_att(). Der Returncode der Funktion EstablishBuffer() war ungleich LB_OK.
XATT13	MX	XAP-TP Baustein. Modul KCOXFEX, Funktion apext_att(). Der Returncode der Funktion RequestBuffer() war ungleich LB_OK.
XFAG09	M	XAP-TP Baustein. Modul KCOXFHF, Funktion GetAttribute(). Die Funktion RequestBuffer() lieferte den Returncode LB_NOMEM.
XFGA07	SUX	XAP-TP Baustein. Modul KCOXFHF, Funktion GetAttribute(). Beim Lesen des Attributs AP_DTNID in der singletasking Variante lieferte die Funktion AllocUserMem() einen unerwarteten Returncode.
XFGA11	M	XAP-TP Baustein. Modul KCOXFHF, Funktion GetAttribute(). Der Makro mGetBufferSpace() lieferte den Returncode LB_NOMEM.
XFGE01	F	XAP-TP Baustein. Modul KCOXFHF, Funktion bCheckAndGetCallEnv(). Die Funktion bCheckAndSetState() lieferte einen schlechten Returncode und der Anwendungsstatus war WAITING_DUMP_APPL. Eine andere Task hat den Anwendungsabbruch veranlasst.
XFSA07	MX	XAP-TP Baustein. Modul KCOXFHF, Funktion SetAttribute(). Beim Setzen des Attributs AP_DTNID lieferte die Funktion PutElement() einen Returncode ungleich DM_OK.
XFTM01	SX	XAP-TP Baustein. Modul KCOXFHF, Funktion TraceMgmt(). Die OSS Funktion o_tron() lieferte den Returncode O_ERROR.
XFTM02	SX	XAP-TP Baustein. Modul KCOXFHF, Funktion TraceMgmt(). Die OSS Funktion o_tron() lieferte den Returncode O_INVEREF.

Code	Grp.	Beschreibung
XFTM03	SX	XAP-TP Baustein. Modul KCOXFHF, Funktion TraceMgmt(). Die OSS Funktion o_tron() lieferte einen unbekanntem Returncode.
XFTM04	SX	XAP-TP Baustein. Modul KCOXFHF, Funktion TraceMgmt(). Die OSS Funktion o_troff() lieferte den Returncode O_ERROR.
XFTM05	SX	XAP-TP Baustein. Modul KCOXFHF, Funktion TraceMgmt(). Die OSS Funktion o_troff() lieferte einen unbekanntem Returncode.
XINI06	ASX	XAP-TP Baustein. Modul KCOXFEX, Funktion apex_init(). Die OSS Funktion o_create() lieferte den Returncode O_ERROR. mögliche Ursachen: Falls das Insert XP1INFO in der zugehörigen UTM Meldung P001 den Wert 1 hat, konnte das OSS shared memory nicht eingerichtet werden.
XINI07	SX	XAP-TP Baustein. Modul KCOXFEX, Funktion apex_init(). Die OSS Funktion o_create() lieferte einen unbekanntem Returncode.

Maßnahmen bei Sperren von UTM-Cluster-Dateien

Bei allen Fehlern, die beim Anfordern oder beim Freigeben von Sperren auf UTM-Cluster-Dateien aufgetreten sind, wird folgende Umgehungsmaßnahme empfohlen:

- ▶ Erhöhen Sie in der der CLUSTER-Anweisung der KDCDEF-Generierung den Wert des Parameters FILE-LOCK-TIMER-SEC oder des Parameters FILE-LOCK-RETRY.

Zur weiteren Diagnose beachten Sie bitte auch die letzten K190-Meldungen.

K061 Die Dumpdatei &FNAM wurde erzeugt

K062 Die Dumpdatei konnte nicht erzeugt werden

K064 Die Nachricht wurde verworfen : &PTRM/&PRNM/&BCAP/<RM &DEVC &FIL1A &FIL2A &FIL3 &VTRC &CBRC &MSG2 &REA1

Bedeutung:

1. An der Workbourse wurde ein ungültiges oder unerwartetes Announcement empfangen. In diesem Fall hat das Insert &REA1 den Wert X'05' und das Insert &MSG2 enthält das empfangene Announcement.
2. Von einem LTERM-Partner wurde eine Nachricht empfangen, die aus Sicht von UTM logisch inkonsistent ist, z.B. weil der strenge Dialog nicht eingehalten wurde.
3. Von einem LU6.1 Partner wurde eine Nachricht empfangen, die keinen gültigen Transmission Header enthält.

Zur Diagnoseunterstützung werden die ersten 32 Zeichen der Nachricht ausgegeben. Bei fehlerhaften Eingabe-Nachrichten wird die Verbindung abgebaut oder ein automatischer KDCDISP durchgeführt.

Der standardmäßig auf SYSLOG ausgegebene Satz hat folgenden Aufbau:

INSERTS	Bedeutung
&DEVC	PTYPE laut KDCDEF-Generierung
&FIL1A	Status der Anwendung mit den Werten X'02' = Startphase X'03' = normaler Lauf X'04' = shut warn X'05' = shut grace X'07' = fast shut X'08' = term application
&FIL2A	Status des Client bzw. Druckers: X'00' = Verbindung wird abgebaut X'01' = Verbindung nicht aufgebaut X'02' = UTM wartet auf Bestätigung des Verbindungswunsches X'03' = Verbindung aufgebaut, Benutzer noch nicht angemeldet X'04' = UTM wartet auf Eingabe eines Ausweises X'05' = UTM wartet auf Eingabe eines Kennwortes X'06' = Anmeldeverfahren über MUX X'07' = 2. Teil des Anmelde-Vorgangs X'08' = LTERM mit USAGE=O: Drucker verbunden. Sonst: Benutzer ist angemeldet
&FIL3	Status des physikalischen Terminals bzw. Druckers, dient zur Diagnose in Fehlerfällen.
&VTRC	Interner UTM-Returncode, dient zur Diagnose in Fehlerfällen.
&CBRC	Interner UTM-Returncode, dient zur Diagnose in Fehlerfällen.

INSERTS	Bedeutung
&IMSG2	Die ersten 32 Zeichen der Nachricht
&REA1	Grund der K064-Meldung: X'02' = ungültiger Funktions-Key X'03' = keine positive Abdruckquittung X'04' = schlechter Returncode vom ISLP X'05' = inkonsistente Eingabe-Nachricht X'06' = LU6.1: Die Anno-Länge ist kürzer als das DCF im Transmission-Header X'08' = inkonsistente Nachricht von einem LU6.1 Partner X'09' = UPIC-Eingabe-Nachricht mit ungültigem Protokoll X'0A' = Inkonsistente physikalische oder logische Quittung X'0B' = Eingabe-Nachricht von einem als Ausgabepartner generierten Socket-Client X'0C' = Es wurde ein nicht erwartetes GO-Signal für eine UPIC- oder Socket-Verbindung empfangen

K065 Netzmeldung : &PTRM/&PRNM/&BCAP/<RM &FIL1B &FIL2B

Die Inserts &FIL1B und &FIL2B haben dabei folgende Bedeutung:

Inserts	Bedeutung
&FIL1B	BCAM-Call oder (UTM-)Announcement
&FIL2B	Diagnostic Word oder UTM-Announcement-Code

&FIL1B wird abdruckbar ausgegeben. Ein Wert zwischen X'F0' und X'FF' kennzeichnet ein UTM-Announcement. Dabei kennzeichnen die Werte X'E0', X'E2' und X'F0' bis X'FF' ein UTM-(Timer-)Announcement. Die Werte werden in der folgenden Tabelle erklärt:

&FIL1B	Bedeutung	Bedeutung von &FIL2B
X'01' - X'4C'	BCAM-Call oder BCAM-Announcement ¹	BCAM-Infoword (abdruckbar)
X'E0'	Idle-Timeout für eine LU6.1-Session (siehe KDCDEF-Anweisung SESCHA, Parameter IDLETIME)	ohne Bedeutung

&FIL1B	Bedeutung	Bedeutung von &FIL2B
X'E2'	Timeout beim Empfang von Nachrichtenfragmenten auf einer Socket-Verbindung, da die Nachricht nicht innerhalb der erlaubten Zeitspanne von 10 Minuten (interner Wert) komplett empfangen wurde	ohne Bedeutung
X'F0'	UTM-Anno	ohne Bedeutung
X'F1' - X'FA'	UTM-Anno	ohne Bedeutung
X'FB'	UTM-Anno: Timeout beim Warten auf eine Quittung von einem Drucker oder von einer TS-Anwendung nach dem Senden einer Asynchron-Nachricht (siehe KDCDEF-Anweisung MAX, Parameter LOGACKWAIT)	ohne Bedeutung
X'FC' - X'FE'	UTM-Anno	ohne Bedeutung
X'FF'	UTM-Anno: Timeout	Das erste Byte von &FIL2B gibt die Ursache des Timeout an:
		X'21' Ablauf des Timers, der den Aufbau einer LU6.1-Session überwacht. Siehe KDCDEF-Anweisung UTMD, Parameter CONCTIME=(time1,...).
		X'22' Ablauf des Timer, der den Empfang der Quittung für eine über eine LU6.1-Session gesendete Asynchron-Nachricht überwacht. Siehe KDCDEF-Anweisung UTMD, Parameter CONCTIME=(...,time2).
		X'30' Ablauf des PEND-KP-Timer. Siehe KDCDEF-Anweisung MAX, Parameter TERMWAIT oder PGWTTIME.

&FIL1B	Bedeutung	Bedeutung von &FIL2B
		X'40' Ablauf des PTERM-Idle-Timer. bis Siehe KDCDEF-Anweisung X'43' PTERM bzw. TPOOL, Parameter IDLETIME oder KDCDEF- Anweisung MAX, Parameter PGWTTIME.

¹ Wert und Bedeutung von BCAM-Call und -Announcement können Sie den BCAM-Diagnoseunterlagen entnehmen bzw. beim Systemdienst erfragen

Die Werte des Diagnostic Word sind hexadezimal angegeben und werden in der folgenden Tabelle erklärt.

Byte 1-4	Bedeutung	Maßnahme
00000008	ungültige Parameter	PM
00000014	Connection letter zu lang	PM
00000028	vorausgegangene Sendeaktion noch nicht beendet (Senden von Daten)	normales Verhalten
0000002C	Nachricht konnte nicht vollständig empfangen werden	UTM-Generierung überprüfen
00000030	interner Fehler	siehe Dump-Fehlercode WAIT 41 bei der K060 Meldung
00000038	vorausgegangene Sendeaktion noch nicht beendet (Verbindungsauf-/abbau).	normales Verhalten
0400001C	Betriebsmittelengpass	UTM-Generierung prüfen
04000020	Anwendung ist nicht angemeldet	UTM-Generierung prüfen
04000024	Verbindung bereits abgebaut	normales Verhalten
08000024	Verbindung bereits aufgebaut	normales Verhalten
10000024	UTM Netzanbindung noch nicht aktiv	normales Verhalten
1C080000	Anwendungsende durch Dienstprogramm kdcshut mit Zeitangabe	normales Verhalten
1C0C0000	Anwendungsende durch Dienstprogramm kdcshut ohne Zeitangabe	normales Verhalten
24000004	Verbindung bereits abgebaut	normales Verhalten
24000018	negative Antwort auf Verbindungsaufbauwunsch	normales Verhalten
24000038	Verbindungsabbau-Anforderung	normales Verhalten
30000020	Fehler beim Anmelden	PM

K066 Der Pflichtparameter FILEBASE fehlt

K067 Fehler in Verbindung mit Language-Connection-Modul &MOD: Errorcode = &ERCD1 &ERCD2; Opcode = &OPCD2

Die Inserts der Meldung haben folgende Bedeutung:

&MOD	Bedeutung
KDCCC	Verbindungsmodul für C
KDCCCOB2	Verbindungsmodul für Micro Focus COBOL
KDCCCOBN	Verbindungsmodul für NetCOBOL

&OPCD2	Bedeutung
INITIALISE	Initialisierung der Sprachumgebung
DEACTIVATE	Freigabe der Sprachumgebung
S; XXXXXXXX	Starten des Teilprogramms ' XXXXXXXX'
E; XXXXXXXX	Durchführung der Ende-Behandlung für das Teilprogramm ' XXXXXXXX'

Der ERRORCODE setzt sich aus zwei mal 4 Zeichen zusammen.

&ERCD1	Bedeutung und evtl. Maßnahme
0000	Auftrag wurde korrekt durchgeführt
0004	Auftrag wurde nicht korrekt durchgeführt. Fehlerunterlagen erstellen und an Service schicken
0008	Fehler im Anwendungs-Programm
00FF	Der Language-Connectionmodul für eine generierte Sprache ist beim Start der Anwendung nicht eingebunden

Die Inserts in &ERCD2 liefern Zusatzinformationen zu aufgetretenen Fehlern und dienen nur zu Diagnosezwecken.

K068 Datenbank Verbindungsmodul &DBCON Version &DBV1 kann nicht mit KDCDB-Makro Version &DBV2 zusammenarbeiten

K069 Verbindungsabbau : &PTRM/&PRNM/&BCAP/<RM/&REA4/&REA6/&COTM

Das Insert &REA4 zeigt an, wer den Verbindungsabbau initiiert hat. &REA6 enthält die Ursache für den Verbindungsabbau.

&REA4	Bedeutung und evtl. Maßnahme
B	Das Transportsystem meldet Verbindungsverlust
L	Verbindungsverlust bei Socket-Verbindung
sonst	Die Verbindung wurde von UTM abgebaut, Ursache siehe &REA6

&REA6	Bedeutung
X'00'	Grund nicht spezifiziert
X'08'	Betriebsmittelengpass
X'09'	Verbindung abgebaut
X'0A'	SHUTDOWN der Anwendung
X'0D'	Negativer Returncode vom Transportsystem, siehe zugehörige K065-Meldung.
X'31'	Verbindungsabbau nach KDCOFF
X'32'	Verbindungsabbau nach SIGN OF
X'33'	Verbindungsabbau durch die Druckadministration
X'34'	Verbindungsabbau durch die Administration
X'35'	Keine Nachrichten für Drucker mit PLEV > 0
X'36'	Pagepool-Warnstufe 2 überschritten beim Empfangen einer asynchronen VTV-Nachricht.
X'39'	Ungültige Funktionstaste
X'3A'	Verbindungsabbau durch VTV bei noch nicht aufgebauter Session
X'3B'	Verbindungsabbau durch UPIC
X'3C'	Verbindungsabbau nach erfolglosem Anmeldeversuch
X'3D'	Verbindungsabbau nach erfolglosem Anmelde-Vorgang
X'3E'	Verbindungsabbau nach Timeout
X'3F'	Nachrichtenfragment empfangen
X'40'	Negative Transport- oder Abdruckquittung
X'41'	Unerwartetes Protokoll-Element
X'42'	Inkonsistente Eingabe-Nachricht
X'43'	Verbindungsabbau durch neuen Verbindungsaufbauwunsch
X'47'	Verbindungsabbau nach Ablauf des IDLE-Timers
X'49'	Socket - Verbindungsabbau wegen ungültiger Länge der eingehenden Nachricht (negativ oder >32000)

&REA6	Bedeutung
X'4A'	Socket - Verbindungsabbau wegen ungültiger Version im Protokoll Header
X'4B'	Socket - Verbindungsabbau wegen ungültigem Typ im Protokoll Header
X'4C'	Socket - Verbindungsabbau nach einem Fehler beim Schreiben der Nachricht in die Message Queue
X'4D'	Socket - Verbindungsabbau nach Timeout
X'4E'	Socket - Verbindungsabbau wegen Längenproblem: die generierte Nachricht ist größer als die generierte maximale Länge (vgl. Operand MAX TRMSGLTH in der KDCDEF Anweisung)
X'4F'	Socket - Verbindungsabbau wegen unbekanntem Identifier im Protokoll-Header
X'50'	Socket - Verbindungsabbau wegen ungültiger Version Minor im Protokoll-Header
X'51'	Socket - Verbindungsabbau wegen ungültigem Flag im Protokoll-Header
X'52'	Socket - Verbindungsabbau wegen Überschreitung der maximal möglichen Anzahl von Fragmenten
X'53'	Socket - Verbindungsabbau wegen falschem Messagetype FRAGMENT zu Beginn einer Nachricht
X'54'	Socket - Verbindungsabbau wegen fehlendem letzten Teil eines Fragments.
X'57'	Abbau der Verbindung zum Client, da nach Timeout an der PGWT-Börse nicht genug Prozesse aktiv sind, um den Vorgang normal fortzusetzen.
X'59'	Verbindungsabbau durch VTV bei aufgebauter Session.
X'60'	Socket-Verbindungsabbau wegen nicht mehr unterstützter Protokoll-Version 1.0 (Version Minor X'00') im Protokoll-Header.

K070 USER / LSES inaktiv : &USER, &GLOBALSG

Das Insert &GLOBALSG 'Cluster Global Sign' ist nur in UTM-Cluster-Anwendungen versorgt und kann folgende Werte annehmen:

&GLOBALSG	Bedeutung
Y	Cluster-globale Abmeldung
N	Knoten-lokale Abmeldung
T	Die Cluster-globale Abmeldung war nicht erfolgreich, weil die Cluster-User-Datei innerhalb der generierten Zeit nicht gesperrt werden konnte.
F	Die Cluster-globale Abmeldung war nicht erfolgreich, weil der Benutzer nicht in der Cluster-User-Datei gefunden wurde.
S	Die Cluster-globale Abmeldung war nicht erfolgreich, weil der Benutzer nicht Cluster-global angemeldet war.
O	Die Cluster-globale Abmeldung war nicht erfolgreich, weil der Benutzer nicht an diesem Knoten Cluster-global angemeldet war.
' '	Abmeldung für einen LU6.1-Session-User oder einen Verbindungs-User.

Die Meldung besitzt zusätzlich folgende Inserts:

Insert	Bedeutung
&COTM	Anzahl Sekunden seit dem Anmelden von &USER auf dieser Verbindung
&CPTM	Verbrauchte CPU-Zeit in msec in der Benutzererkennung &USER, wenn unter der Benutzererkennung &USER zur Zeit nur noch ein Benutzer angemeldet war. Wenn sich vorher Benutzer unter mehreren Verbindungen unter &USER angemeldet haben, dann enthält der Wert auch ihren CPU-Verbrauch. Anschließend wird der Wert wieder auf 0 zurückgesetzt. Wenn unter der Benutzererkennung &USER noch weitere Benutzer angemeldet sind, dann ist der Wert = 0.

K071 Interner Fehler im Datenbankanschluss: Operation=&OPCD1, Fehler=&ERCD3

Die Inserts der Meldung haben folgende Bedeutung:

DBCON = XA (Datenbankanschluss über XA)

OPCD1 DB-Operationscode siehe [Abschnitt „DB-DIAGAREA“ auf Seite 107](#)

ERCD Fehlercode KCR CDC siehe [Abschnitt „DB-DIAGAREA“ auf Seite 107](#)

DBTRAC DB-Trace-Information. Die möglichen Werte und deren Bedeutung sind bei dem jeweiligen Datenbanksystem beschrieben.

Bei einem Anschluss des DB-Systems über die XA-Schnittstelle haben die 4 Bytes der DB-Trace-Information folgende Bedeutung:

Byte 1	Index des Ressource-Managers (vgl. Reihenfolge in den Startparametern)
Byte 2 und 3	Opcode des XA-Aufrufs Ausgabe sedezimal und plattformabhängig (EBCDIC bzw. ASCII),
Byte 4	Returncode des XA-Aufrufs, Ausgabe sedezimal und plattformabhängig

K072 Die Anzahl der &STMT-Einträge im ROOT und in der KDCFILE stimmen nicht überein

Diese Meldung wird ausgegeben, wenn beim Start der Anwendung erkannt wird, dass die Generierungen von ROOT und KDCFILE hinsichtlich der Anzahl von Programmen nicht übereinstimmen. Der Start der Anwendung bzw. des Prozesses wird mit der Meldung K049, Code 40 abgebrochen.

Maßnahme: ROOT und/oder KDCFILE neu erzeugen

K073 Das Attribut &ATTR des &STMTs &PROG im ROOT und in der KDCFILE stimmt nicht überein

Diese Meldung wird ausgegeben, wenn beim Start einer Task festgestellt wird, dass die Einträge der PROGRAM Tabelle im ROOT und in der KDCFILE nicht übereinstimmen. Das erste Insert gibt das Attribut an, in dem die Einträge differieren.

Der Start der Anwendung bzw. des Prozesses wird mit Meldung K049, Code 41 abgebrochen.

Maßnahme: ROOT und/oder KDCFILE neu erzeugen

K074 Programmaustausch abgeschlossen; &CTYP &PROG &PVER

(siehe K075)

K075 Programmaustausch von Prozess &PID abgebrochen; &CTYP &PROG &PVER

K074 bzw. K075 werden nach positivem bzw. negativem Abschluss eines Programmaustauschs ausgegeben. Die Inserts geben Information darüber, welcher Teil des Anwendungsprogramms ausgetauscht wurde bzw. nicht ausgetauscht werden konnte. Es bedeuten:

&PID:	Prozess-ID des Prozesses, von dem der Programmaustausch abgebrochen wurde.
&CTYP:	APPL=Austausch des gesamten Anwendungsprogramms LMOD=Austausch eines Shared Objects
&PROG:	Name des Shared Objects (nur bei CTYP=LMOD)
&PVER:	Versionsbezeichner des Shared Objects (nur bei CTYP=LMOD)

K076 Fehler bei asynchroner Administration mit Transaktionscode &ADTC; KCRCCC=&RCCC, KCRCDC=&RCDC**K077** Interne UTM Cluster Kommunikation mit &PTRM/&PRNM/&BCAP/<RM erfolgreich: &CLSIGT

Die K077-Meldung wird nur bei eingeschaltetem Test-Modus (siehe [Seite 41](#)) ausgegeben.

In der internen Cluster-Kommunikation wurde eine Nachricht &CLSIGT an eine andere Knoten-Anwendung gesendet. &CLSIGT kann folgende Werte annehmen:

&CLSIGT	Bedeutung
1	Prüfen auf Verfügbarkeit der überwachten Knoten-Anwendung (check alive, ping)
2	Prüfen auf Änderungen in der Cluster-Konfiguration
4	Administrationsauftrag
8	Auftrag zum Wecken eines auf einen GSSB oder ULS wartenden Prozesses
16	Antwort an die überwachende Knoten-Anwendung auf eine Verfügbarkeits-Anfrage (check alive response, pong)

Es ist keine Maßnahme notwendig.

K078 &ERRNAME &ERRCODE: in &REA3

Die Meldung kann von openUTM beim Start der Anwendung, bei der Signalbehandlung oder bei der Beendigung eines Workprozesses ausgegeben werden.

Die Meldung K078 wird von UTM in mehreren Varianten ausgegeben, abhängig vom aufgetretenen Ereignis.

K078 yyyyyyyyy: IN modulname zzzzzzzz**K078** yyyyyyyyy: zzzzzzzz**K078** xxxxxxxx yyyyyyyyy: IN modulname text

Die zugehörigen Fehler sind nachfolgend beschrieben. Es bedeuten:

xxxxxxx

enthält eine Kurzbezeichnung des aufgetretenen Fehlers (siehe Tabelle).

yyyyyyyyy

Fehlercode der aufgerufenen Funktion (näheres siehe vorherige K078).

zzzzzzzz

spezifische, kontextabhängige Fehlermeldung.

modulname

enthält den Modulnamen des Moduls, in dem dieser Fehler aufgetreten ist (für interne Diagnose)

text enthält zusätzliche Informationen bei bestimmten Fehlern

xxxxxxx	Fehlerursache	Maßnahme
ALME	Beim Anfordern von Speicher kann nicht genug Speicherplatz bereitgestellt werden	Nicht genug Speicher, ggf. System tunen
APPLI 01	DMS-Fehler beim Zugriff auf die applifile. applifile steht unter <i>utmpfad/applifile</i> und enthält Namen und Status der UTM-Anwendung	Anwenderfehler
APPLI 02	Anwendung in applifile nicht bekannt, oder utmwork zum Testen gestartet, ohne dass utmain den Start vorbereitet hat	Anwenderfehler
APPLI 03	Anwendung in applifile nicht gestartet	Anwenderfehler

xxxxxxx	Fehlerursache	Maßnahme
ARG 01	Ungültige Anzahl von Argumenten beim Start von utmwork.	Anwenderfehler: utmwork mit richtigen Argumenten starten oder utmwork hat die falsche Version
ATEXIT 00	Bei Prozessbeendigung wurde die Funktion atexit() aufgerufen.	Informationsmeldung
ATEXIT 01	Systemteil fordert mit exit() die Prozessbeendigung an.	Informationsmeldung
ATEXIT 02	Anwendungsteilprogramm fordert mit exit() die Prozessbeendigung an.	Informationsmeldung, danach folgt PEND ERROR.
ATEXIT 03	Bei Prozessbeendigung wurde die Funktion atexit() rekursiv aufgerufen.	Informationsmeldung
ATEXIT 04	Systemteil fordert unerlaubt mit exit() die Prozessbeendigung an.	Informationsmeldung, danach folgt TERM APPLICATION.
BIND	Fehler beim Laden eines Lademoduls: Mehr Informationen gibt das Insert' LMOD: name / version ':	siehe vorherige K078
COB 01	cobtidy() wird unterdrückt nach exit() im Micro Focus COBOL-Anwendungsprogramm.	Informationsmeldung
COB 02	Start des COBOL-Anwendungsprogramms wird unterdrückt nach exit() im Micro Focus COBOL-Anwendungsprogramm.	Informationsmeldung
COB 03	JMPCINT3() und JMPCINT4() werden unterdrückt nach exit() im NetCOBOL-Anwendungsprogramm.	Informationsmeldung
COB 04	Start des COBOL-Anwendungsprogramms wird unterdrückt nach exit() im NetCOBOL-Anwendungsprogramm.	Informationsmeldung
CPUT 00	Die CPU Zeit für den Prozess konnte vom System nicht geliefert werden.	Informationsmeldung
CPUT 01	Die CPU Zeit für die Kindprozesse konnte vom System nicht geliefert werden.	Informationsmeldung
CPUT 02	Die CPU Zeit für den Prozess wurde auf den Maximalwert LONG_MAX gesetzt.	Informationsmeldung

xxxxxxx	Fehlerursache	Maßnahme
DIAG 01	Die Umgebungsvariable UTM_BREAK_BEFORE_KCSTRMA ist gesetzt.	Informationsmeldung (nur unter Windows-Systemen)
DIAG 02	Die Umgebungsvariable UTM_ABORT_WITH_EXCEPTION ist gesetzt.	Informationsmeldung
DIAG 03	Die Umgebungsvariable UTM_CORE_DUMP ist auf NO gesetzt.	Informationsmeldung
DIAG 04	Die Umgebungsvariable UTM_EXIT_CORE ist auf YES gesetzt	Informationsmeldung
DIAG 05	Die Umgebungsvariable UTM_ABORT_WITH_EXCEPTION ist gesetzt	Informationsmeldung
DIAG 06	exit()-Aufruf im Anwendungsprogramm führt zur Beendigung des utmwork-Prozesses	Informationsmeldung
DIAG 07	exit()-Aufruf im Systemteil führt zur Beendigung des utmwork-Prozesses	Informationsmeldung
DIAG 08	Die Umgebungsvariable UTM_MAIN_KILL_TIME ist gesetzt.	Informationsmeldung
DLCLOSE 01 SHL_UNLOAD 01 FreeLibrary 01	Fehler beim Schließen eines Shared Objects.	UTM-Generierung und Returncode überprüfen.
DLOPEN 01 SHL_LOAD 01 LoadLibrary 01	Fehler beim Nachladen eines Shared Objects.	UTM-Generierung und Returncode überprüfen. Umgebungsvariable
DLSYM 01 SHL_FINDSYM 01 GetProcAddress 01	Fehler beim Holen der Adresse zu einem Symbol in einem Shared Object.	LD_LIBRARY_PATH bzw. SHLIB_PATH auf HP-UX überprüfen.
dlclose shl_unload FreeLibrary	Diagnosemeldung beim Schließen eines Shared Objects.	Informationsmeldungen
dlopen shl_load LoadLibrary	Diagnosemeldungen beim dynamischen Nachladen von Programmen. Diese Meldungen treten nur bei eingeschaltetem Testmodus auf.	Informationsmeldungen
dlsym shl_findsym GetProcAddress	Diagnosemeldungen beim Holen der Adresse zu einem Symbol in einem Shared Object	Informationsmeldungen

xxxxxxx	Fehlerursache	Maßnahme
ENV 00	Angabe über den Typ der UTM-Anwendung: standalone / cluster / node_recovery / beanconnect	Informationsmeldung
ENV 01	Angaben über das Betriebssystem	Informationsmeldung
ENV 02	Angabe über den Bit-Modus der UTM-Anwendung	Informationsmeldung
ENV 03	Der compare- und swap-Algorithmus wurde ausgeschaltet.	Informationsmeldung
ENV 04	Die Umgebungsvariable UTM_NO_GCORE_DUMP ist auf YES gesetzt. D.h. beim Erzeugen eines UTM-DUMPs wird kein gcore zusätzlich erzeugt. Standardmäßig wird beim Erzeugen eines UTM-Dumps zusätzlich auch ein gcore erzeugt.	Informationsmeldung (nur auf Unix-Systemen)
EXIT xx	Signalnummer xx im Start oder Shutexit aufgetreten. Es wird im Zusatzfeld „STARTEXT“ oder „SHUTEXIT“ eingetragen.	Anwenderfehler
HOST 01	Die Umgebungsvariable UTM_NET_HOSTNAME ist gesetzt. Die Hostname-Datei <file> ist im Einsatz.	Informationsmeldung
HOST 02	Die Umgebungsvariable UTM_NET_HOSTNAME ist gesetzt. Die Hostname-Datei <file> konnte aber wegen DMS-Fehler <error> nicht verwendet werden.	Anwenderfehler
IPC 01	Fehler beim Einrichten des IPC Shared Memory.	UTM-Generierung überprüfen (MAX IPCTRACE) oder KDCREM aufrufen.
IPC 02	Speicherengpass beim Einrichten des IPC Shared Memory	UTM-Generierung überprüfen Unix-Systeme: Kernel Parameter SHMMAX tunen
IPC 03	Die Umgebungsvariable UTM_IPC_LETTER ist gesetzt	Informationsmeldungen
IPC 04	Die Umgebungsvariable UTM_IPC_LETTER ist gesetzt	Informationsmeldungen
IPC 05	Anzahl der IPC Letter wurde geändert	Informationsmeldungen

xxxxxxx	Fehlerursache	Maßnahme
IPC 06	Anzahl der IPC Elemente wurde geändert	Informationsmeldungen
IPC 07	Die Umgebungsvariable UTM_IPC_EXTP_LETTER ist gesetzt	Informationsmeldungen
KCSASIO	Vollständiger Text: K078 KCSASIO: error KCSTRMA ("&REAS>") file: "<&FNAME>" position:<&POS>" Ursache: Die Verarbeitung wird wegen fehlerhafter (=zu großer) Dateiposition &POS für Datei &FNAME mit Grund &REAS abgebrochen.	Bei KDCDEF-Lauf: Dateien verkleinern, z.B. mehr Pagepool-Dateien generieren (MAX PGPOOLFS=)
KCSASIO (Forts.)	Vollständiger Text: K078 "KCSASIO notice: write to "&FNAME" took "&SEC" secs" Ursache: Das Schreiben auf Datei &FNAME dauerte &SEC Sekunden.	Informationsmeldung
	Vollständiger Text: K078 "KCSASIO notice: read from "&FNAME" took "&SEC" secs" Ursache: Das Lesen von Datei &FNAME dauerte &SEC Sekunden.	Informationsmeldung
KCSCHKD	Vollständiger Text: K078 "KCSCHKD notice: back to "lost" directory "&DIR" ok" Ursache: Ein Prozess hat wegen I/O Fehler sein Dateiverzeichnis &DIR gesetzt	Informationsmeldung
KCSROPN	Vollständiger Text: K078 "KCSROPN notice: reopening for file "&FNAME"" Ursache: Die Datei &FNAME wurde wegen I/O Fehler neu geöffnet.	Informationsmeldung

xxxxxxx	Fehlerursache	Maßnahme
	Vollständiger Text: K078 "KCSROPN notice: reopening return_code for file "&FNAME" is "&DMSERR" Ursache: Beim erneuten Öffnen wurde ein Returncode zurückgegeben. &DMSERR = " " (d.h. Leerzeichen) bedeutet "OK"	Informationsmeldung
KCXOFLK	Vollständiger Text: K078 KCXOFLK LOCK-WAIT/ LOCK-IMM error "&ERRNO" for file " &FILENAME" Ursache: Ein Lockaufruf wurde mit errno abgewiesen.	Diagnosemeldung
KDCS 00	CALL „KDCS“ im START-Exit, SHUT-Exit oder INPUT-Exit	Teilprogramm korrigieren
MEM 01	Speicherengpass beim Starten der Anwendung. Vollständiger Text: K078 MEM 01 in utmwork nn Bytes not available.	Speicherbedarf überprüfen, Betriebs- system tunen
MSG 01	Die Umgebungsvariable UTM_MSG_DATE ist auf NO gesetzt	Informationsmeldung
MSG 02	Die Umgebungsvariable UTM_MSG_PID ist auf NO gesetzt	Informationsmeldung
NET 01	Die Umgebungsvariable UTM_APPLI_CONLET ist gesetzt	Informationsmeldung
OSS 02	Fehler beim Einrichten/Anmelden des OSS Shared Memories	UTM Generierung überprüfen oder kdcrem aufrufen
OSS 03	Fehler beim Laden des OSS Shared Memories in den Adressraum.	UTM Generierung überprüfen.
PIPE 01	Die Umgebungsvariable UTM_PIPE_TIME ist gesetzt	Informationsmeldung
READ 01	inkonsistente KDCFILE beim Starten der Anwendung.	Anwenderfehler

xxxxxxx	Fehlerursache	Maßnahme
SEM 01	Fehler beim Einrichten der Semaphore. Dieser kann in openUTM unter Windows-Systemen beispielsweise auftreten, wenn beim Starten der Anwendung noch ein dtp läuft.	Bei der UTM-Generierung den Wert von <i>number1</i> in MAX SEMARRAY reduzieren
SEM 02	Fehler beim Anmelden an ein Semaphor	Systemfehler
SIGNAL 01	Signal im utmwork-Prozess empfangen. Normale Signalbehandlung	Informationsmeldung, danach folgt entweder PEND ERROR oder TERM APPLICATION
SIGNAL 02	Signal SIGTRAP im utmwork-Prozess empfangen.	Informationsmeldung, Signal wird entweder ignoriert oder Anwendung wird mit TRMA IPCREM beendet.
SIGNAL 03	Nicht vorgesehenes Signal SIGALRM in utmwork Prozess empfangen	Informationsmeldung, Signal wird ignoriert.
SIGNAL 04	Ungültiges Signal in utmwork Prozess empfangen	Informationsmeldung, Signal wird ignoriert.
SIGNAL 05	Anwendungseigene Signalaroutine für SIGALRM ist nicht erlaubt.	Informationsmeldung
SIGNAL 06	Das Setzen einer anwendungseigenen Signalaroutine mit sigaction() für SIGALRM ist nicht erlaubt.	Informationsmeldung
STARTEXTIT	Startexit hat mit exit() den Workprozess beendet.	Informationsmeldung
STXIT OFF	Meldung: „STXIT OFF in utmwork: termination of utmwork process creates gcore dump“	Informationsmeldung
STXIT OFF	Meldung: „STXIT OFF in utmwork: exit() call produces termination of utm- work process with abort().“	Informationsmeldung
SYSROT 01	Die Umgebungsvariable UTM_REDIRECT_FILES ist gesetzt	Informationsmeldung
SYSROT 02	Die Umgebungsvariable UTM_MAIN_NODIRECT ist gesetzt	Informationsmeldung
TIMR 01	Die Umgebungsvariable UTM_TIMER_RETRY ist gesetzt	Informationsmeldung

xxxxxxx	Fehlerursache	Maßnahme
UNBIND	Fehler beim Entladen eines Lademoduls. Weitere Informationen gibt das Insert ' LMOD: name / version '	siehe vorherige K078
VERS 01 VERS 02	Version bzw. Bit-Modus des Programms und der KDCA-Datei passen nicht zusammen	Anwenderfehler Maßnahme: UTMPATH korrekt setzen
VSVI1	Fehler beim Ermitteln der Programm-Adressen. Weitere Informationen gibt das Insert ' for program: name'	siehe vorherige K078

K078 WARNING : in KDCRTBF! No XA-Connection generated, but startparameters given!

Es wurde ein Startparameter für den Resource Manager angegeben, es ist aber keiner generiert. Generierung oder Startparameter ändern!

K079 Accounting-Problem - Ursache: &REA2

Beim Versuch, Accounting-Information zu schreiben, meldet die Schreibroutine einen Fehler. Die Anwendung läuft weiter.

Im Insert &REA2 wird die Fehlerursache angezeigt:

REA2	Bedeutung
28	Fehler im Schreibprozess des UTM-Accounting. openUTM schaltet das Accounting aus. Die Anwendung läuft weiter.

Maßnahme: Überprüfen, ob der utmlog-Prozess noch existiert.

Nach Behebung des Fehlers kann der UTM-Administrator die Kalkulations- und/oder Abrechnungsphase mit dem Kommando KDCAPPL wieder einschalten.

K080 KDCMON ist nicht eingeschaltet

K081 Statistik: &IMSG1/&OMSG1/&CONU/&ATAC2/&LWRT/&HITR/&WTBF

Inserts	Bedeutung
&IMSG1	Anzahl Terminal Input Messages 1)
&OMSG1	Anzahl Terminal Output Messages 1)
&CONU	Anzahl connected Users
&ATAC2	Anzahl unprocessed ATAC' s
&LWRT	Anzahl Log Writes 1)
&HITR	Cache Hit Rate
&WTBF	Cache Waits for Buffer 1)

Die Meldung wird zu jeder vollen Stunde und bei normaler Beendigung der Anwendung geschrieben, mit 1) markierte Werte werden danach wieder auf 0 gesetzt.

K082 Falsche Datei &FNAM

K083 Die Datei &FNAM ist zerstört

K084 &OBJ1 &VER1 &OST1 &BMD1 und &OBJ2 &VER2 &OST2 &BMD2 sind nicht verträglich

K085 Die &FNKT Funktionen sind nicht verfügbar

K086 UTM-D Fehlerinformation: &PTRM, &PRNM, &BCAP, <RM, &USER, &SYSD, &USSD, &FMH7, &AGUS

Inserts	Bedeutung
&PTRM	Name des phys. Terminals bzw. Druckers
&PRNM	Prozessorname
&BCAP	Anwendungsname
<RM	Name des LTERM-Partners
&USER	Benutzerkennung
&SYSD	SNA sense code (system)
&USSD	SNA sense code (user)
&FMH7	Fehler-Nachricht von der entfernten Anwendung
&AGUS	Benutzerkennung des Auftraggebers

Die Inserts &SYSD und &USSD können bei homogener Kopplung folgende Werte annehmen:

&SYSD&USSD	Bedeutung
08120000	Betriebsmittelengpass (z.B. Pagepool voll)
08130000 081B0000	Contention: Eine Sessionbelegungsanforderung von der Contention-Loser-Anwendung wurde abgelehnt, da die Contention-Winner-Anwendung die Session für einen Auftrag belegt hat.

&SYSD&USSD	Bedeutung
08641003	ungültiger oder gesperrter TAC
0864C5E2	PEND ER durch UTM
0864C5D9	PEND ER vom Anwendungsteilprogramm
0864D9E2	PEND RS durch UTM
0864D9E4	PEND RS vom Anwendungsteilprogramm
0866D9E2	PEND RS durch UTM
0866D9E4	PEND RS vom Anwendungsteilprogramm
10030000	ungültiger oder gesperrter TAC

K088 UTM-D Session Start : &LSES/&RSES/&LPAP
 SR-STATE: &SRFG PET &PSQN
 SAVED : &ESQS &EBSS
 ACT. : &ESQR &ESRR &ESBR

UTM-D Sessionaufbau. Die Meldung enthält Diagnoseinformationen.

Inserts	Bedeutung
&LSES	LSES-Name
&RSES	RSES-Name
&LPAP	LPAP-Name
&SRFG	Status Bits aus dem LSES-Tabelleneintrag (SR)
&PSQN	PET-Sequenznummern zweimal 2 Bytes: – Sequence number sent saved – Sequence number sent current
&ESQS	gesicherte Sequenznummern zweimal zwei Bytes: – SLU TO PLU – PLU TO SLU
&EBSS	gesicherter Bracketzustand – "BETB" = BETween Bracket – "INBR" = IN Bracket Receive (lokale Seite ohne Senderecht) – "INBS" = IN Bracket Send (lokale Seite mit Senderecht)

Inserts	Bedeutung
&ESQR	aktuelle Sequenznummern (aus STSN request) Insgesamt 5 Bytes: <ul style="list-style-type: none"> – SEC_TO_PRI_SQN (2 Byte) – PRI_TO_SEC_SQN (2 Byte) – ACTION_CODE_SEC_TO_PRI (2 Bit) <ul style="list-style-type: none"> 00: ignore 01: set 10: test 11: set and test – ACTION_CODE_PRI_TO_SEC (2 Bit) <ul style="list-style-type: none"> 00: ignore 01: set 10: test 11: set and test – RESERVED (4 Bit)
&ESRR	aktuelle Sequenznummern RSP(STSN) Insgesamt 5 Bytes: <ul style="list-style-type: none"> – SEC_TO_PRI_SQN (2 Byte) – PRI_TO_SEC_SQN (2 Byte) – ACTION_CODE_SEC_TO_PRI (2 Bit) <ul style="list-style-type: none"> 00: ignore 01: set 10: test 11: set and test – ACTION_CODE_PRI_TO_SEC (2 Bit) <ul style="list-style-type: none"> 00: ignore 01: set 10: test 11: set and test – RESERVED (4 Bit)
&EBSR	aktueller Bracketzustand <ul style="list-style-type: none"> – "BETB" = BETween Bracket – "INBR" = IN Bracket Receive (lokale Seite ohne Senderecht) – "INBS" = IN Bracket Send (lokale Seite mit Senderecht)

- K089** Der Auftrag zum Loeschen der asynchronen Nachricht (&GNDATE/&GNTIME) an &DEST wurde angenommen.
&GNUSER / &USER / (&DLDATE/&DLTIME) / &CHAIN

&CHAIN	Bedeutung
NO	Zu der gelöschten Nachricht war keine negative Folgenachricht vorhanden.
DEL	Die negative Folgenachricht soll ebenfalls gelöscht werden.
ACT	Die negative Folgenachricht soll aktiviert, d.h. in die Nachrichtenkette ihres Empfängers eingekettet werden.

Eine evtl. vorhandene positive Folgenachricht wird bei Ausführung des Auftrags in jedem Fall gelöscht.

- K090** Der Auftrag zum Loeschen aller asynchronen Nachrichten an &DEST wurde angenommen.
&USER / (&DLDATE/&DLTIME)
- K091** Wegen Betriebsmittelengpass ist zur Zeit kein Anmelden an Anwendung &BCAP moeglich.
- K092** Bitte Passwort und wahlweise neues Passwort eingeben
- K093** Vorgangskellerung nicht moeglich.
Siehe auch KDCDEF-Steueranweisung MAX, Operand NRCONV.
- K094** Sequenz von erfolglosen Anmeldeversuchen.

Mit dieser Meldung löst openUTM einen stillen Alarm aus, um auf eine Folge von erfolglosen Anmeldeversuchen eines Benutzers oder von einem Client aus hinzuweisen. In der Anwendung kann dann z.B. über das MSGTAC-Teilprogramm auf diese Situation reagiert werden. Über die Anweisung SIGNON SILENT-ALARM=*nnn* des Dienstprogramms KDCDEF kann eingestellt werden, nach wievielen erfolglosen Versuchen openUTM die Meldung K094 erzeugen soll.

Die Meldung enthält folgende Inserts:

Inserts	Bedeutung
&PTRM	Stationsname
&PRNM	Prozessorname
&BCAP	Anwendungsname
<RM	Name des LTERM-Partners
&USER	Benutzer-Identifikation
&RCF1B	Grund für die Ablehnung
&REA4	Verursacher der Ablehnung (L=LTERM, U=USER, B=BOTH)

Das Insert &USER enthält die Benutzer-Identifikation des letzten erfolglosen Anmeldeversuchs der Sequenz. Dies kann eine für die Anwendung generierte Benutzerkennung oder aber eine beliebige Zeichenfolge sein, die UTM als Benutzer-Identifikation übergeben wurde.

Das Insert &RCF1B enthält in drei Zeichen den Grund für die Zurückweisung des letzten Anmeldeversuchs der Sequenz. Es wird von UTM so versorgt, wie die Rückgabefelder KCRSIGN1 und KCRSIGN2 des KDCS-Aufrufs SIGN ST.

Das Insert &REA4 zeigt, ob der Benutzer (U), das Client-LTERM (L) oder beide (B) Verursacher der Meldung sind. Der Zähler für die fehlerhaften Anmeldeversuche des Verursachers wird auf 0 zurückgesetzt.

K095 KDCOFF ist wirksam - Bitte Eingabe

K096 Kellerung beendet - Bitte Eingabe

K097 Die Angaben zum neuen Passwort sind nicht verwendbar - Bitte Anmelden

K098 Input-Exit &RCF1C &RCF2B - Bitte Eingabe

Die Meldung wird dann erzeugt, wenn beim Aufruf des INPUT-Exits von UTM oder vom Exit selbst Fehler entdeckt wurden.

Insert &RCF1C zeigt an, was UTM bei oder nach dem Aufruf des INPUT-Exits entdeckt:

&RCF1C	Bedeutung
IN00	Aufruf des INPUT-Exits aus Sicht von UTM o.k.
IN01	Fehler beim Aufruf des INPUT-Exits über IUTMHLL
IN02	KDCS-Aufruf im INPUT-Exit
IN03	Ausgabeparameter passen nicht zu KCICCD
IN04	unzulässiger Wert in KCICCD
IN05	DB-USER-CALL im INPUT-Exit
IN06	Signal im INPUT-Exit

Insert &RCF2B zeigt an, was der INPUT-Exit im Fall KCICCD = 'ER' in den Ausgabeparameter KCIERRCD eingetragen hat.

K099 &MSG

K101 Engpass - Bitte Eingabe wiederholen

Die Meldung wird bei folgenden Ereignissen ausgegeben:

- Wenn eine Dialog-Eingabe-Nachricht wegen einer TAC-Klassen Wartesituation im Pagepool zwischengespeichert werden soll und der Platz im Pagepool nicht mehr ausreicht.
- Wenn für ein Asynchron-Programm, eine TAC-Queue oder eine temporäre Datei eine bestimmte Eingabe-Nachricht auf Pagepool gesichert werden soll und entweder der Warnlevel 2 überschritten ist oder der Platz auf Pagepool nicht ausreicht.
- Wenn eine Eingabe-Nachricht für ein Asynchron-Programm, eine TAC-Queue oder eine Temporäre Queue bestimmt ist und die Anzahl der gesicherten Nachrichten für diesen Asynchron-TAC, diese TAC-Queue oder diese Temporäre Queue den generierten Wert QLEV schon erreicht oder überschritten hat und die Temporären Queues mit QMODE=STD generiert wurden.

K104 UTM-D &UTMDEVT (&RCVDANNO): &LSES, &LPAP, &AGUS
; alter Status: (&OCVST, &OTAST); Aktion: &ACTION; neuer Status: (&NCVST, &NTAST).

Bei Kommunikation über LU6.1 wird die Meldung bei folgenden Ereignissen ausgegeben:

- Bei Ablauf eines der folgenden Timer:
 - Session-Belegungs-Timer für einen Dialog-Auftrag (siehe Wert *time1* bei der KDCDEF-Anweisung LTAC, Parameter WAITTIME).
Das Insert &LSES enthält in diesem Fall Leerzeichen.
Ausnahme: Timer-Ablauf nach Belegung einer Contention-Loser-Session.
Sonderfall:
Ist der Session-Belegungs-Timer einer Contention-Winner-Session gleich 0 und bei Teilprogrammende keine Session frei, wird dies wie der Ablauf eines Timer behandelt.
 - Antwort-Timer (siehe Wert *time2* bei der KDCDEF-Anweisung LTAC, Parameter WAITTIME).
In diesem Fall enthalten &LSES und &AGUS unterschiedliche Namen.
 - Prepare-to-Commit-Timer (siehe KDCDEF-Anweisung UTMD, Parameter PTCTIME). &LSES und &AGUS sind in diesem Fall gleich.
- Bei Verbindungsverlust, falls die Session belegt ist
 - von einem Vorgang mit offener Transaktion oder
 - von einem Vorgang, der das Transaktionsende eingeleitet hat.

- Bei Sessionwiederanlauf, falls die Session von einem Vorgang belegt ist, der das Transaktionsende eingeleitet hat.

Bei Kommunikation über OSI TP wird die Meldung bei Ablauf eines der folgenden Timer ausgegeben. &LSES enthält immer Leerzeichen.

- Association-Belegungs-Timer (für Dialogaufträge siehe Wert *time1* bei KCDEF-Anweisung LTAC, Parameter WAITTIME, für Asynchroneaufträge interner Wert von 60 Sekunden).
Bei einem Asynchron-Auftrag enthält &ACTION den Wert ASYNCH.
- Antwort-Timer (für Dialogaufträge siehe Wert *time2* bei KCDEF-Anweisung LTAC, Parameter WAITTIME, für Asynchroneaufträge siehe Wert *time2* bei KCDEF-Anweisung UTMD, Parameter CONCTIME).
Bei einem Dialog-Auftrag wird auf die Antwort, bei einem Asynchron-Auftrag auf die Quittung von Auftragnehmer gewartet. Bei einem Asynchron-Auftrag enthält &ACTION den Wert ASYNCH.
- PEND-KP- und PGWT-KP-Timer (siehe KDCDEF-Anweisung MAX, Parameter TERMWAIT bzw. PGWTTIME).
- Ready-Timer (siehe KDCDEF-Anweisung UTMD mit Parameter PTCTIME oder KDCDEF-Anweisung MAX mit Parameter PGWTTIME)

Inserts	Bedeutung
&UTMDEVT	Ereignis, bei dem die Meldung ausgegeben wurde: RESTAR Sessionwiederanlauf T Verbindungsverlust DISCON Timeout TIMEOUT
&RCVDANNO	Zuletzt an der UTM-Börse empfangenes Announcement. Bei &UTMDEVT = TIMEOUT haben die ersten beiden Bytes in &RCVDANNO folgende Bedeutung: X'F330' Timer-Ablauf im OSI TP-Auftragnehmer-Vorgang nach dem X'F331' Senden einer Nachricht an den Auftraggeber in einer Transaktion (siehe KDCDEF-Anweisung MAX, Parameter TERMWAIT oder PGWTTIME). X'F332' Timer-Ablauf im OSI TP-Auftraggeber-Vorgang nachdem alle Auftragnehmer zum Einleiten des Transaktionsendes aufgefordert wurden. Der Timer entspricht dem größten generierten Wert von <i>time</i> in der KDCDEF- Anweisung MAX PGWTTIME= und von <i>time2</i> in der KDCDEF-Anweisung LTAC WAITTIME= der beteiligten LTACs.

Inserts	Bedeutung	
&RCVDANNO (Forts.)	X'F333'	Ablauf des Ready-Timer im OSI TP-Auftragnehmer-Vorgang (KDCDEF-Anweisung UTMD, Parameter PTCTIME oder ;KDCDEF-Anweisung, Parameter MAX PGWTTIME).
	X'F400'	Ablauf des OSI TP-Association-Belegungs-Timer für einen Dialog-Auftrag (<i>time1</i> in KDCDEF-Anweisung LTAC, Parameter WAITTIME).
	X'F520'	Ablauf des OSI TP-Association-Belegungs-Timer für einen Asynchron-Auftrag (interner Timer von 60 Sekunden).
	X'F522'	Ablauf des Timer, der den Empfang der Quittung für eine Asynchron-Nachricht überwacht, die über eine OSI TP-Association gesendet wurde (<i>time2</i> in KDCDEF-Anweisung UTMD, Parameter CONCTIME).
	X'F534'	Ablauf des OSI TP-Antwort-Timer im Auftraggeber-Vorgang für einen Dialog-Auftrag. Der Timer entspricht dem größten generierten Wert von <i>time</i> in der KDCDEF- Anweisung MAX PGWTTIME= und von <i>time2</i> in der KDCDEF-Anweisung LTAC WAITTIME= der beteiligten LTACs.
	X'F800'	Ablauf des LU6.1-Session-Belegungs-Timer für einen Dialog-Auftrag (<i>time1</i> in KDCDEF-Anweisung LTAC, Parameter WAITTIME)).
	X'F933'	Ablauf des Prepare-to-Commit-Timer im LU6.1-Auftragnehmer-Vorgang (KDCDEF-Anweisung UTMD, Parameter PTCTIME oder KDCDEF-Anweisung MAX, Parameter PGWTTIME).
	X'F934'	Ablauf des LU6.1-Antwort-Timers in einem Dialog-Auftrag (Der Timer entspricht dem größten generierten Wert von <i>time</i> in der KDCDEF- Anweisung MAX PGWTTIME= und von <i>time2</i> in der KDCDEF-Anweisung LTAC WAITTIME= der beteiligten LTACs.).
	X'F935'	Ablauf des LU6.1-Antwort-Timers in einem Dialog-Auftrag nach erneutem Aufbau der LU6.1-Session (Der Timer entspricht dem größten generierten Wert von <i>time</i> in der KDCDEF- Anweisung MAX PGWTTIME= und von <i>time2</i> in der KDCDEF-Anweisung LTAC WAITTIME= der beteiligten LTACs.).
	X'F936'	Ablauf des LU6.1-Antwort-Timer beim Vorgangwiederanlauf nach Aufbau einer Session (<i>time2</i> in KDCDEF-Anweisung LTAC, Parameter WAITTIME)
&AGUS	Name des Auftraggebers (Benutzer, Session oder Association)	

Inserts	Bedeutung
&ACTION	Maßnahme von openUTM: COMMIT Beenden der Transaktion RESET Rücksetzen der Transaktion WAIT nichts STPROG Folgeprogrammstart ASYNCH Erneute Ausführung des Asynchron-Auftrag.
&OCVST &OTAST &NCVST &NTAST	Vorgangs- und Transaktionsstatus von dem in &AGUS angegebenen Auftraggeber (vor und nach der in &ACTION angegebenen Aktion). Mögliche Werte: O O Vorgang und Transaktion sind offen. O P Der Vorgang ist offen und hat Transaktionsende eingeleitet. O C Der Vorgang ist offen, die Transaktion beendet. C C Vorgang und Transaktion wurden beendet. O R Die Transaktion wird zurückgesetzt, der Vorgang bleibt offen. Z R Die Transaktion wird rückgesetzt, der Vorgang beendet.

K105 UTM-D &SYST mismatch; &LSES, &LPAP, &AGUS

&SYST enthält das System, mit dem ein Mismatch entstanden ist:
 DB = Datenbank / LPAP = andere Anwendung

K119 OSI-TP Fehlerinformation: &OSLPAP, &USER, &TAC, &DIA1, &DIA2, &DIA3

Die Erklärungen in den folgenden Tabellen beziehen sich auf Protokollelemente (Primitive) an der XAP-TP Schnittstelle. Diese Datenelemente haben folgende Bedeutung:

Datenelement	Bedeutung
APM_ALLOCATE_CNF	Antwort (Confirmation) auf die Anforderung einer Association für einen OSI TP-Dialog. Eine negative Antwort bedeutet, dass keine Association bereitgestellt werden konnte.
TP_BEGIN-DIALOGUE_REQ	Anforderung (Request) an den Partner, einen OSI TP-Dialog zu beginnen.
TP_BEGIN_DIALOGUE_IND	Anzeige (Indication), dass der Partner einen OSI TP-Dialog beginnen möchte.
TP_BEGIN_DIALOGUE_CNF	Antwort auf die Anforderung, einen OSI TP-Dialog zu beginnen. Eine negative Antwort bedeutet, dass der Partner den Dialog abgelehnt hat.
TP_END_DIALOGUE_REQ	Anforderung an den Partner, den OSI TP-Dialog zu beenden.
TP_END_DIALOGUE_IND	Anzeige, dass der Partner den OSI TP-Dialog beenden möchte.

Datenelement	Bedeutung
TP_U_ABORT_IND	Anzeige, dass der ferne TP Service User ¹ den OSI TP-Dialog abgebrochen hat.
TP_P_ABORT_IND	Anzeige, dass der entfernte TP Service Provider ² den OSI TP-Dialog abgebrochen hat.
TP_DATA_IND	Anzeige, dass Daten vom Partner übergeben wurden.
TP_U_ERROR_IND	Anzeige, dass der Partner einen Fehler gemeldet hat.
TP_HANDSHAKE_IND	Anzeige einer Synchronisierungs (Handshake)-Anforderung durch den Partner.
APUTM_ABORT_REQ	Anforderung zum Abbruch einer Verbindung.

¹ TP Service User: UTM-Anwendung

² TP Service Provider: Systemteil, der den OSI TP-Service bereitstellt. Bei openUTM ist das der XAP-TP-Baustein.

Das Insert &DIA1 enthält die Ursache für die Ausgabe der Meldung K119.

&DIA1	Bedeutung
01	Es wurde eine negative APM_ALLOCATE_CNF empfangen.
02	Es wurde eine negative TP_BEGIN_DIALOGUE_CNF empfangen.
03	Eine TP_BEGIN_DIALOGUE_IND wurde von openUTM abgelehnt.
04	Es wurde eine TP_U_ABORT_IND bzw. eine TP_P_ABORT_IND empfangen.
05	Es wurde eine TP_U_ERROR_IND empfangen.
06	Empfang von TP_HANDSHAKE_IND für eine Dialog-Nachricht wird von openUTM nicht unterstützt.
07	Empfang von TP_END-DIALOGUE_IND vom Client für eine Dialog-Nachricht wird von openUTM nicht unterstützt, evtl. wurde bei einem asynchronen LTAC ein Dialog-RTAC generiert.
08	Bei der Unchained Transactions Functional Unit fordert UTM, dass die Transaktion mit dem Dialog beginnt.
09	Die Association wurde mit APUTM_ABORT_REQ abgelehnt.
10	Betriebsmittelengpass in openUTM

Die Inserts &DIA2 und &DIA3 enthalten je nach Wert von &DIA1 folgende Zusatzinformationen:

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
01	Verursacher der Ablehnung:	Diagnoseinformation:
	00 ACSE Service User 01 ACSE Service Provider	-1 Kein Grund angegeben 02 Application Context Name wird nicht unterstützt 03 Application Process Title (APT) des Auftraggebers ist unbekannt 04 Application Identifier (APID) des Auftraggebers ist unbekannt 05 Application Entity Qualifier (AEQ) des Auftraggebers ist unbekannt 06 Application Entity Identifier (AEID) des Auftraggebers ist unbekannt 07 Application Process Title (APT) des Auftragnehmers ist unbekannt 08 Application Identifier (APID) des Auftragnehmers ist unbekannt 09 Application Entity Qualifier (AEQ) des Auftragnehmers ist unbekannt 10 Application Entity Identifier (AEID) des Auftragnehmers ist unbekannt
	02 Presentation Service Provider	nicht relevant
01 (Forts.)	06 TP Service Provider	-1 kein GUND angegeben. 00 CCR Version 2 nicht verfügbar. 01 TP Protokoll Version nicht kompatibel 02 Keine Contention Winner Association verfügbar. Mögliche Ursache: In der Partner-Anwendung ist die Anzahl der Associations kleiner als in der lokalen UTM-Anwendung.
	07 Association Pool Manager	-1 kein GUND angegeben. 04 Association Pool Manager kann keine weitere Association aufbauen 06 Lokaler oder ferner Application Entity Title (AET) ist unbekannt. 07 Keine passende Association für angegebenen AET gefunden.

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
02	Verursacher der Ablehnung: 02 TP Service Provider 03 TP Service User	Diagnoseinformation: 00 Kein Grund angegeben 01 TPSU title nicht bekannt. 02 TPSU title permanent nicht verfügbar. 03 Mögliche Gründe sind: <ul style="list-style-type: none"> – TPSU title ist temporär nicht verfügbar – In der Anwendung des Auftragnehmer-Vorgangs ist keine freie OSI TP-Instanz-Tabelle für den ACCESS-POINT verfügbar. – Es ist keine freie Dialog-Tabelle verfügbar. – Es besteht ein anderer Betriebsmittelengpass in openUTM. Die Meldung K119 mit &DIA1=10 auf Auftragnehmer-Seite enthält nähere Informationen über die Art des Engpasses. Maßnahme: Auf der Auftragnehmer-Seite für die OSILPAP-Partner des ACCESS-POINT eine größere Anzahl Associations generieren.

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
02 (Forts.)	02 TP Service Provider 03 TP Service User	04 Kein TPSU title angegeben. 05 Functional Unit nicht unterstützt. 06 Kombination von Functional Units nicht unterstützt. 07 Association reserviert für Partner. 08 Partner-AEI unbekannt. 11 Functional Unit Shared Control wird von openUTM nicht unterstützt. 12 TPSU title nicht decodierbar. 13 Typ des TPSU title nicht unterstützt. 14 Länge des TPSU title zu groß 15 Decodierfehler für Benutzerdaten. 16 Decodierfehler für Security-Daten. 17 Abstrakte Syntax (UTMSEC) nicht bekannt bzw. nicht generiert. 18 Benutzererkennung unbekannt bzw. Partner-Anwendung ohne USER generiert oder Benutzererkennung abgelehnt, z.B. wegen eines falschen 19 Passworts. Partner nicht verfügbar, z.B. gesperrt (STATUS=OFF) oder wegen QUIET-Kommando. 20 Benutzererkennung oder Passwort zu 21 lang. Warnlevel 2 für den Pagepool wurde 22 überschritten. Mögliche Gründe sind: – Ungültiger Transaktionscode – Transaktionscode gesperrt 23 – Administrationsberechtigung fehlt. Transaktionscode nicht mit CALL=FIRST 24 generiert. Asynchron-Vorgang kann nicht gestartet werden. Der für den Asynchron-TAC generierte Queue Level (maximale Anzahl der Nachrichten in der Message 25 Queue) ist bereits erreicht. DIALOG-Vorgang kann nicht gestartet werden, weil der Transaktionscode als Asynchron-TAC generiert ist (KDCDEF- 26 Anweisung TAC TYPE=A). Kombination von Functional Unit und Restart-Funktionalität (KDCDISP) wird nicht unterstützt.

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
03	Verursacher der Ablehnung: 03 TP Service User	Diagnoseinformation: 04 Kein TPSU title angegeben 11 Functional Unit Shared Control wird von openUTM nicht unterstützt. 12 TPSU title nicht decodierbar 13 Typ des TPSU title nicht unterstützt. 14 Länge des TPSU title zu groß. 15 Decodierfehler für Benutzerdaten. 16 Decodierfehler für Security-Daten. 17 Abstrakte Syntax (UTMSEC) nicht bekannt bzw. nicht generiert. 18 Benutzerkennung unbekannt bzw. Partner-Anwendung ohne USER generiert oder Benutzerkennung abgelehnt, z.B. wegen eines falschen Passworts. Genauere Informationen können Sie der vorhergehenden Meldung K147 19 entnehmen. Partner nicht verfügbar, z.B. gesperrt (STATUS=OFF) oder wegen QUIET- 20 Kommando. 21 Benutzerkennung oder Passwort zu lang. Warnlevel 2 für den Pagepool wurde überschritten.
04	Verursacher der Ablehnung: 02 TP ServiceProvider 03 TP Service User	Diagnoseinformation: 00 Kein Grund angegeben. 01 Ein permanenter Fehler ist aufgetreten. 02 TP_BEGIN-TRANSACTION_REQ abgelehnt. 03 Ein temporärer Fehler ist aufgetreten. 04 Ein Protokollfehler ist aufgetreten. 05 Kollision zweier TP_END-DIALOGUE_REQ Primitive. 06 Kollision von TP_BEGIN-TRANSACTION-REQ und TP_END-DIALOGUE-REQ.

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
09	Verursacher der Ablehnung: 03 TP Service User	Diagnoseinformation: 21 Warnlevel 2 für den Pagepool erreicht. Die Asynchron-Nachricht konnte nicht gespeichert werden. Maßnahme: Größeren Pagepool generieren ¹ 31 Die KDCFILE ist voll: Die Nachricht konnte nicht gespeichert werden. Maßnahme: Größeren Pagepool generieren ¹ 32 Decodierfehler für UDT-Daten eines TP_DATA_IND Primitives In allen Fällen wird die Association abgebaut.
10	Betroffenes Betriebsmittel: 11 In der Anwendung des Auftraggeber-Vorgangs ist keine freie OSI TP-Instanz-Tabelle für diesen ACCESS-POINT verfügbar. Ursache ist eine Nachricht an einen OSI TP-Partner, aber alle OSI TP-Instanz-Tabellen sind belegt. Die Nachricht wird in eine Warteschlange eingereiht. Maßnahme: Für die OSI-LPAP-Partner des Access Point eine größere Anzahl Associations generieren ² .	Diagnoseinformation: nicht relevant

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
10 (Forts.)	12 In der Anwendung des Auftraggeber-Vorgangs ist keine freie OSI TP-Instanz-Tabelle für diesen OSI-LPAP-Partner verfügbar. Ursache ist eine Nachricht an den OSI TP-Partner, aber alle OSI TP Instanz-Tabellen sind belegt. Die Nachricht wird in eine Warteschlange eingereiht. Maßnahme: Für die OSI-LPAP-Partner des Access Point eine größere Anzahl Associations generieren ² .	Index OSI-LPAP
	13 In der Anwendung des Auftraggeber-Vorgangs ist die Instanz in XAPTP belegt. Die Nachricht wird in eine Warteschlange eingereiht. Maßnahme: Für die OSI-LPAP-Partner des Access Point eine größere Anzahl Associations generieren ² .	Index UTM-Instanz
	14 In der Anwendung des Auftraggeber-Vorgangs trat ein Fehler beim Lesen der zu sendenden Asynchron-Nachricht auf. Die Nachricht wird in eine Warteschlange eingereiht.	nicht relevant

&DIA1	&DIA2 / Bedeutung	&DIA3 / Bedeutung
10 (Forts.)	<p>15 Keine freie OSI TP-Node-Tabelle verfügbar. Dieser Fehler kann sowohl in einer Anwendung eines Auftraggeber-Vorgangs als auch in einer Anwendung eines Auftragnehmer-Vorgangs auftreten.</p> <p>In einer Anwendung eines Auftraggeber-Vorgangs wird die Nachricht in eine Warteschlange eingereiht.</p> <p>Maßnahme: Für die OSI-LPAP-Partner eine größere Anzahl Associations generieren².</p>	nicht relevant
	<p>16 In der Anwendung des Auftragnehmer-Vorgangs ist keine freie OSI TP-Vorgangstabelle verfügbar.</p> <p>Maßnahme: Für die OSI-LPAP-Partner eine größere Anzahl Associations generieren².</p>	nicht relevant
	<p>17 In der Anwendung des Auftragnehmer-Vorgangs ist keine freie OSI TP-Benutzer-Tabelle für den OSI-LPAP-Partner verfügbar.</p> <p>Mögliche Ursache: Für den OSI-LPAP-Partner ist noch ein Dialog-Teilprogramm aktiv, dessen Association zwischenzeitlich abgebaut wurde.</p>	Index OSI-LPAP

¹ KDCDEF-Anweisung MAX PGPOOL=

² KDCDEF-Anweisung OSI-LPAP ASSOCIATIONS=

K120 Die Gültigkeit des Passworts ist abgelaufen

K121 Das Passwort ist nur noch &NUMDAYS Tag(e) gültig.

K122 Das Passwort ist nur noch &NUMDAYS Tag(e) gültig.

K123 LTERM hat nicht die Berechtigung den Vorgang fortzusetzen.
- Bitte Anmelden

K124 Fehler: &RCXAPTP beim Start von XAP-TP in Phase: &PHAXAPTP aufgetreten

Die Meldung wird als Diagnosehilfe bei Fehlern während des Starts von XAP-TP ausgegeben.

Bedeutung der Inserts:

&PHAXAPTP Phase beim Starten von XAP-TP. Mögliche Werte:
INIT (Übergabe der Generierungswerte an XAP-TP)
START/RECOVERY (OPEN und BIND der Instanzen sowie Recovery nicht beendeter Transaktionen).

&RCXAPTP Returncode, der den Fehler genauer spezifiziert. Je nach Phase (Wert von &PHAXAPTP) haben die Returncodes unterschiedliche Bedeutungen. Sofern der Returncode nicht in der folgenden Tabelle aufgelistet ist, handelt es sich um einen internen Fehler (in diesem Falle müssen Sie eine Fehlermeldung schreiben).

&PHAXAPTP	&RCXAPTP	Bedeutung
INIT	18	Die OSS-Version ist kleiner als 4.0A20.
	20	KDCFILE und ROOT passen nicht zusammen. Mögliche Ursache: Es wurde eine neue KDCFILE mit KDCDEF erzeugt (neu: mit OSI TP, alt: ohne OSI TP); die UTM-Anwendung wurde aber nicht neu gebunden bzw. die ROOT-Source wurde nicht neu übersetzt.
	16 60 85	Es konnte nicht genug Speicherplatz angefordert werden, um die OSI TP-spezifischen Generierungswerte abzudecken.
	21	Der KCSALME Aufruf zum Anlegen des Übergabepuffers für die Transfer Syntaxen liefert einen schlechten Returncode.
	41	Die „apext_init“ Sequenz wurde bereits früher durchgeführt.
	44	Der KCSALME Aufruf zum Anlegen des Übergabepuffers für die Abstrakten Syntaxen liefert einen schlechten Returncode.
	63	Der KCSALME Aufruf zum Anlegen des Übergabepuffers für die Access Points liefert einen schlechten Returncode.

&PHAXAPTP	&RCXAPTP	Bedeutung
INIT (Forts.)	99	Eine von „apext_att“ aufgerufene Funktion hat einen schlechten Returncode geliefert. Dieser Returncode kann auch bedeuten, dass der Adressraum den OSS für sein Shared Memory in einer Folgetask belegen will, bereits mit einem anderen Shared Memory (z.B. KAA, XAPTP-Global-memory) belegt ist. Maßnahme: Versuchen weitere Tasks nachzustarten.
	102	Fehler bei attach. Maßnahme: Prüfen Sie, ob auf diesem Rechner unter diesem Namen bereits eine Anwendung angemeldet ist.
	106	XAPTP-Aufruf „apext_att“ Returncode: APEXT_ATTACH_INVEREF Bedeutung: Die beim OSS-Aufruf o_attach angegebene Waiting Point Referenz ist ungültig bzw. nicht (mehr) bekannt. Mögliche Ursache: Folgetask kann nicht gestartet werden, da die Anwendung schon beendet wird.

K125 Passwort nicht komplex genug - Administrator verstaendigen oder KDCOFF eingeben

K126 Negativer SAT-Returncode: &SATRC

K127 Interner Fehler in UTM - ROOT: &ERCD6

Das Insert &ERCD6 enthält die Ursache für die Ausgabe der Meldung K127.

&ERCD6	Bedeutung
RT04	fehlerhafter Aktionsindex in KDCRTMM

K128 UTM-D Auftrag abgelehnt: &CON/&PRNM/&BCAP/&LPAP &LSES &REA1 &RCDC &TAC

Handelt es sich um einen OSI TP UTM-D Auftrag, haben die Inserts folgende Bedeutung:

&CON: OSI-CON Name
 &PRNM: Acht Leerzeichen
 &BCAP ACCESS-POINT Name
 &LPAP OSI-LPAP Name

Das Insert &REA1 enthält die Ursache für die Ausgabe der Meldung K128.

&REA1	Bedeutung
X'01'	Ungültiger Transaktionscode – LU6.1: DPN oder PRN im FMH-5 oder FMH-6 oder die ersten 8 Zeichen der Nachricht – OSI TP: Recipient-tpsu-title in TP-BEGIN-DIALOGUE-RI &RCDC enthält einen Fehlercode KCRDC
X'02'	Transaktionscode nicht mit CALL=FIRST generiert – LU6.1: DPN oder PRN im FMH-5 oder FMH-6 oder die ersten 8 Zeichen der Nachricht – OSI TP: Recipient-tpsu-title in TP-BEGIN-DIALOGUE-RI
X'03'	Ein asynchroner Vorgang soll gestartet werden, TAC ist mit TYPE=D generiert – LU6.1: Nachricht mit Begin Bracket und End Bracket – OSI TP: Empfang eines TP-END-DIALOGUE-RI Protokollelements
X'04'	Ein Dialog-Vorgang soll gestartet werden, TAC ist mit TYPE=A generiert – LU6.1: Nachricht mit Begin Bracket und Change Direction – OSI TP: Empfang von TP-GRANT-CONTROL-RI oder TP-HANDSHAKE-AND-GRANT-CONTROL-RI
X'05'	Ein asynchroner Vorgang soll gestartet werden, aber der QLEV des Asynchron-TAC ist bereits erreicht. – LU6.1: Die Verbindung wird abgebaut. – OSI TP: Die Verbindung wird abgebaut
X'06'	Nur bei heterogener Kopplung über LU6.1: Eine asynchrone Nachricht wurde mit RQE empfangen. openUTM erwartet RQD2. Maßnahme: Bei CICS muss im START-Kommando NOCHECK und PROTECT angegeben werden
X'07'	Nur bei heterogener Kopplung über LU6.1: Eine Nachricht wurde mit EC und RQD2 empfangen, aber weder CD noch EB sind gesetzt. Maßnahme: CICS- bzw. IMS-Programm verbessern.

K129 &CMD - Startparameter zur Zeit nicht erlaubt! Anweisung wird ignoriert

K130 Task-Prioritaet &TPRIO ist fuer Task &TASK nicht erlaubt! Prioritaet nicht geaendert.

K132 &OBJ1 und &OBJ2 stammen nicht aus dem gleichen KDCDEF-Lauf

K133 Programm fuer &EXIT fehlt!

K134 Nachricht an &DEST wurde in die Dead Letter Queue gestellt. &NMSG Nachricht(en) in der Dead Letter Queue

Sobald der generierte Schwellwert erreicht ist, wird der Administrator mit dieser Meldung über Nachrichten informiert, die nach abnormaler Verarbeitung in die Dead Letter Queue gestellt wurden.

Die Inserts haben folgende Bedeutung:

Insert	Bedeutung
&DEST	Ursprüngliches Ziel der Nachricht, die zum Erreichen des Schwellwertes in der Dead Letter Queue führte.
&NMSG	Generierter Schwellwert für Anzahl Nachrichten in der Dead Letter Queue

K135 UPIC-Meldung :
&PTRM/&PRNM/&BCAP/<RM/&UPCREAS/&UPCSTAT/&UPCROT/&UPVEN
C1/&UPPENC2

Es trat ein Problem bei der Zusammenarbeit mit einem UPIC-Partner auf.

Der Wert des Inserts &UPCREAS zeigt die Ursache an:

&UPCREAS	Bedeutung	Ursache / Maßnahme
01	UPIC-Partner sendet ohne Senderecht	Fehler im Produkt UPIC oder UTM
02	UPIC-Partner sendet zu kurzes Protokoll	Fehler im Produkt UPIC
03	UPIC-Partner verwendet nicht-unterstützte Protokoll-Version	UPIC-Version passt nicht zu UTM-Version
04	UPIC-Partner sendet ungültiges Protokoll	UPIC-Fehler
05	UPIC-Partner hat Abbruch der Conversation eingeleitet	Wirkung der UPIC-Funktion Deallocate() (CMDEAL())
06	UPIC-Partner hat keinen TAC geschickt	UPIC-Fehler
07	Der vom UPIC-Partner gesendete TAC: – ist nicht generiert – LTERM/USER hat keinen Key – ist admin.TAC und der USER ist kein admin – existiert und Programm fehlt	Anwenderfehler (Side info oder SETTP)
08	UPIC-Partner hat TAC geschickt, der mit CALL=NEXT generiert ist	Anwenderfehler
09	UPIC-Partner hat TAC geschickt, der gesperrt/gelöscht ist oder zu geringe Verschlüsselung besitzt	Anwenderfehler: (Side info oder SETTP)

&UPCREAS	Bedeutung	Ursache / Maßnahme
0A	UPIC-Partner hat zu kurzen TAC geschickt	UPIC-Fehler
0B	UPIC-Partner hat TAC bei offenem Vorgang geschickt	UPIC-Fehler
0C	UPIC-Partner hat TAC bei Folgeteil geschickt	UPIC-Fehler
0D	UTM-Pagepool ist zu klein für Eingabe-Nachricht	Pagepool der Anwendung ist zu klein generiert (Anweisung MAX PGPOOL)
0E	UPIC-Partner gibt bei offener Nachricht nur Senderecht ab	UPIC-Fehler
0F	UPIC-Partner gibt bei offener Nachricht Senderecht ab ohne aktuellen Nachrichtenteilabschluss	UPIC-Fehler
10	UPIC-Partner hat TAC geschickt, der mit TYPE=A (Asynchron-TAC) generiert ist	Anwenderfehler
11	UPIC-Partner ist beim Verbindungsaufbau einem LTERM mit STATUS=OFF (gesperrt) zugeordnet	Normales Verhalten
12	UPIC-Partner ist beim Nachrichtenempfang einem LTERM mit STATUS=OFF (gesperrt) zugeordnet	Normales Verhalten Administrator hat LTERM gesperrt
13	UPIC-Partner hat USER in ungültiger Länge geschickt	UPIC-Fehler
14	UPIC-Partner hat ungültigen USER oder PASSWORD geschickt	Anwenderfehler
15	UPIC-Partner hat ungültige Daten für Vorgangswiederanlauf geschickt, z.B. KDCDISP ohne USER oder mit zusätzlichen Anwenderdaten.	Anwenderfehler
16	UPIC-Partner hat ungültige Daten für Vorgangswiederanlauf geschickt, z.B. KDCDISP für USER mit RESTART=NO	Anwenderfehler
17	Länge des empfangenen Protokolltoken ungültig	UPIC-Fehler
19	Das empfangene Protokolltoken ist ungültig.	UPIC-Fehler
1A	Die Gesamtlänge der empfangenen Daten ist inkonsistent.	UPIC-Fehler oder Transportsystem-Fehler
1B	Das Protokolltoken für Daten war nicht das letzte Protokolltoken.	UPIC-Fehler
1C	Es wurden zwei Protokolltoken für Service empfangen.	UPIC-Fehler

&UPCREAS	Bedeutung	Ursache / Maßnahme
1D	Es wurden zwei Protokolltoken für Userid empfangen.	UPIC-Fehler
1E	Es wurden zwei Protokolltoken für Passwort empfangen.	UPIC-Fehler
1F	Es wurden zwei Protokolltoken für Format empfangen.	UPIC-Fehler
20	Es wurden zwei Protokolltoken für Function Key empfangen.	UPIC-Fehler
21	Es wurden keine Protokolltoken für Daten empfangen.	UPIC-Fehler
22	Es wurden inkonsistente SIGNON-Daten empfangen.	UPIC-Fehler
23	Es wurde ein Protokolltoken für Userid empfangen, aber nicht für Service.	UPIC-Fehler
24	Ungültiger Function Key.	UPIC-Fehler
25	Protokolltoken für Format in Datenfragment empfangen.	UPIC-Fehler
26	Protokolltoken für Function Key in Datenfragment empfangen.	UPIC-Fehler
27	Ungültiger Encryption-Level	UPIC bzw. UTM Fehler
28	Länge der empfangenen User Message ist ungültig	UPIC-Fehler
29	Es ist ein Fehler bei der Protokolldiskussion aufgetreten	UPIC bzw. UTM Fehler
2A	Es wurden zwei Protokolltoken für verschlüsselte Daten empfangen	UPIC-Fehler
2B	Es wurden zwei Protokolltoken für ein verschlüsseltes Passwort empfangen	UPIC-Fehler
2C	Es wurden zwei Protokolltoken für einen RSA Schlüssel empfangen	UPIC-Fehler
2D	Es wurden zwei Protokolltoken für einen DES Schlüssel empfangen	UPIC-Fehler
2E	Es wurden zwei Protokolltoken für einen Cursor empfangen	UPIC-Fehler
2F	Es wurden zwei Protokolltoken für eine Protokoll Diskussion empfangen	UPIC-Fehler
30	Es wurde kein Protokolltoken für verschlüsselte Daten empfangen	UPIC-Fehler

&UPCREAS	Bedeutung	Ursache / Maßnahme
31	Es wurde ein Protokolltoken für ein Passwort und ein verschlüsseltes Passwort empfangen	UPIC-Fehler
32	Es ist ein Fehler bei der Passwortentschlüsselung aufgetreten	UPIC bzw. UTM Fehler
33	Es ist ein Fehler bei der Passwortentschlüsselung aufgetreten. Entschlüsselte Passwort Länge zu groß.	UPIC bzw. UTM Fehler
34	Es ist ein Fehler beim Lesen des DES Schlüssels aufgetreten	UPIC bzw. UTM Fehler
35	Der RSA Schlüssel kann nicht versendet werden	UTM-Fehler
36	Es ist ein Fehler beim Lesen des RSA Schlüssels aufgetreten	UTM-Fehler
37	Es ist ein Fehler bei der Datenverschlüsselung aufgetreten	UTM-Fehler
38	Es ist ein Fehler bei der Datenentschlüsselung aufgetreten	UPIC bzw. UTM Fehler
39	Es sind inkonsistente Message und Conversation Charakteristika aufgetreten	UTM-Fehler
3A	Die Protokoll Diskussion konnte nicht abgeschlossen werden	UTM-Fehler
3B	Es ist ein Fehler beim Schlüsselaustausch aufgetreten	UTM-Fehler
3C	Der UPIC-Partner unterstützt keine Verschlüsselung	Normales Verhalten, Generierung ändern
3D	Es wurden zwei Protokolltoken für den Transaktionsstatus empfangen	UPIC-Fehler
3E	Es wurden zwei Protokolltoken für den Client Context empfangen	UPIC-Fehler
3F	Die Länge der Usermessage im Connectionletter stimmt nicht mit der Protokollversion überein	UPIC-Fehler
40	openUTM unterstützt keine Verschlüsselung	Normales Verhalten, Generierung ändern
41	Zwei Protokolltoken für neues Passwort empfangen	UPIC-Fehler
42	Zwei Protokolltoken für verschlüsseltes neues Passwort empfangen	UPIC-Fehler
43	Protokolltoken für neues Passwort und verschlüsseltes neues Passwort empfangen	UPIC-Fehler

&UPCREAS	Bedeutung	Ursache / Maßnahme
44	Unerlaubter Operationscode beim Aufruf der internen Funktion PASSWD_ENCRYPT_PROC zum Entschlüsseln eines Passwortes aufgetreten.	UTM-Fehler
45	Zwei Protokolltoken für den Client Typ empfangen.	UPIC-Fehler
46	Es konnte kein RSA-Schlüssel gefunden werden.	Einen der RSA-Schlüssel aktivieren

Das Insert &UPCPROT ist das hexadezimal aufbereitete UPIC-Protokoll, &UPCSTAT dient der Diagnose im Fehlerfall.

Die Inserts &UPCENC1 und &UPCENC2 dienen der Diagnose von Datenverschlüsselung:

Das erste Byte von &UPVENC1 gibt den ENCRYPTION_LEVEL der Message, das zweite Byte von &UPVENC1 gibt den ENCRYPTION_LEVEL der Conversation aus.

Das erste Byte von &UPPENC2 gibt den ENCRYPTION_LEVEL der Session, das zweite Byte von &UPPENC2 gibt den ENCRYPTION_LEVEL des Partners aus.

K136 (Erste) SYSLOG-Datei ist &FNAM

openUTM gibt diese Meldung in der Startphase aus. &FNAM enthält den Namen der SYSLOG-Datei. Ist die SYSLOG als Dateigenerationsgruppe angelegt, dann enthält &FNAM den Namen der ersten Dateigeneration, die von openUTM beschrieben wird.

K137 SYSLOG umgeschaltet auf Datei &FNAM

openUTM hat erfolgreich auf eine neue SYSLOG-Dateigeneration umgeschaltet. Das Umschalten wurde durch die Administration oder durch die automatische Größenüberwachung ausgelöst. &FNAM enthält den Namen der neuen SYSLOG-Dateigeneration.

K138 SYSLOG-Datei &FNAM geschlossen

Es sind zwei Fälle zu unterscheiden:

- die SYSLOG wird als einfache Datei geführt:
Die letzte UTM-Task der Anwendung hat die SYSLOG-Datei geschlossen. &FNAM enthält den Namen der SYSLOG-Datei.
- die SYSLOG wird als SYSLOG-FGG geführt:
Eine SYSLOG-Dateigeneration wurde von der letzten UTM-Task (also vollständig) geschlossen. Sie können jetzt frei über diese Dateigeneration verfügen. Sie wird nicht mehr von openUTM benötigt. &FNAM enthält den Namen der geschlossenen Dateigeneration.

K139 Fehler beim Umschalten der SYSLOG-Datei! Es wird weiterhin die Datei &FNAM benutzt

Der Versuch, auf eine neue SYSLOG-Dateigeneration umzuschalten, ist fehlgeschlagen. openUTM arbeitet weiterhin mit der Dateigeneration &FNAM. Den Grund für den Fehler beim Umschalten können Sie evtl. dem DMS-Fehlercode in der vorausgegangen Meldung K043 entnehmen.

K140 Im Bereich &MXP1 bis &MXP2 wird keine MUX-Protokollversion unterstuetzt**K141** Die MUX-Protokollversion &MXP1 wird nicht unterstuetzt**K142** Release-Pending-Timeout fuer Session. PTERM: &PTRM MUX-PTERM: &MXPT**K143** UTM-D: STSN Sequencenumbers Response ungleich Request. Request: &STS1, &STS2 Response: &STS3, &STS4**K145** Wegen Transaktions-Recovery kann Benutzer &USER zur Zeit nicht angemeldet werden - Bitte Anmelden**K146** Erfassung BCAM Wartezeit. OPCODE= &BCMOPCD, RTCODE= &BCMRTCD, Standardheader= &STDHEAD, TSN= &TASK, BCAM-Anwendungsname= &BCAP**K147** Anmeldung fuer &USRTYPE User &USER nicht erfolgreich. &PTRM/&PRNM/&BCAP/<RM Grund: U&REA7

&USRTYPE enthält folgende Werte:

- CONNECTION beim Anmelden der Verbindungs-Benutzererkennung einer Verbindung zu einem UPIC- oder TS-Client
- CLIENT bei Anmeldung einer echten Benutzererkennung über TS-Anwendung, einem UPIC-Client oder einem OSI TP-Partner.

Die Werte in &REA7 haben folgende Bedeutung:

&REA7	Bedeutung
U1	Der angegebene USER existiert nicht.
U2	Der angegebene USER ist gesperrt.

&REA7	Bedeutung
U3	Mit diesem USER hat sich bereits jemand angemeldet.
U4	Das als „altes“ Passwort angegebene Passwort ist falsch.
U5	Angaben zum neuen Kennwort nicht verwendbar
U6	Es ist kein Kartenleser vorhanden
U7	Die Karteninformation ist falsch
U8	Die Anmeldung ist zur Zeit nicht möglich: <ul style="list-style-type: none"> – wegen Betriebsmittelengpass. – weil die Maximalzahl gleichzeitig anmeldbarer Benutzer bereits erreicht ist. – weil ein Passwort nicht geändert werden konnte, da gerade ein inverser KDCDEF läuft.
U10	Der aktuelle LTERM-Partner hat nicht die Berechtigung, den Vorgang fortzusetzen.
U11	Die Gültigkeitsdauer des Passworts wurde überschritten. Das Passwort muss durch den Administrator geändert werden.
U12	Das neue Kennwort erfüllt nicht die Anforderungen
U13	Das neue Kennwort ist zu kurz
U14	Das von KDCUPD übertragene Passwort erfüllt nicht die Anforderungen der generierten Komplexitätsstufe oder es ist zu kurz.
U15	Für den angegebenen USER ist ein Transaktionswiederanlauf erforderlich.
U16	Der offene Vorgang kann von diesem Partnertyp aus nicht fortgesetzt werden.
U17	Vom Administrator wurde SHUT WARN eingegeben; für normale Anwender ist kein Anmelden bei der Anwendung mehr möglich (ein Administrator darf sich noch anmelden).
U18	Auf der Verbindung ist der für die Fortsetzung des offenen Vorgangs nötige Verschlüsselungsmechanismus nicht verfügbar
U19	Die Gültigkeitsdauer des Passwortes ist abgelaufen. Da Grace-Sign-On generiert ist, kann die Anmeldung durch Übergabe eines neuen Passwortes wiederholt werden.
U22	Der angegebene USER existiert nicht in der Cluster-User-Datei.
U23	Mit diesem USER hat sich bereits jemand an einem anderen Knoten angemeldet.
U24	Anmeldung ist z.Z. nicht möglich, weil die Cluster-User-Datei innerhalb der generierten Zeit (Anweisung CLUSTER, Parameter FILE-LOCK-TIMER-SEC, Parameter FILE-LOCK-RETRY) nicht gesperrt werden konnte.
U25	Anmeldung an dieser Knoten-Anwendung nicht möglich, weil ein an eine andere Knoten-Anwendung gebundener Vorgang des Benutzers existiert, der nicht beendet werden darf.
U26	Anmeldung abgelehnt, da der offene Vorgang des Benutzers eine Transaktion im Zustand PTC hat, aber kein Vorgangs-Wiederanlauf angefordert wurde.

K149 Interne Diagnoseinformation &DIA5

Es ist ein Ereignis eingetreten, das mit einer Diagnoseinformation protokolliert wird. &DIA5 ist die UTM-interne Diagnoseinformation.

K151 Lauf des inversen KDCDEF beendet. Returncode: &IDEFRC &DMSE &FNAM

Das Insert &IDEFRC enthält den 16-stelligen Returncode des inversen KDCDEF. Der Returncode setzt sich aus zwei 8-stelligen, abdruckbaren Zahlen zusammen.

Die ersten 8 Zeichen haben folgende Bedeutung

Erste 8 Zeichen	Bedeutung
00000000	Der inverse KDCDEF wurde ordnungsgemäß beendet.
00000001	Bei der Dateibehandlung ist ein Fehler aufgetreten.
00000002	Ein Speicherengpass ist aufgetreten.
00000003	Es wurden keine KDCDEF-Anweisungen generiert.

Die Zeichen 9 - 16 sind nur relevant, wenn in den ersten 8 Stellen 00000001 steht. In diesem Fall enthalten die Zeichen 9 - 16 eine nähere Beschreibung des aufgetretenen Dateifehlers.

In allen anderen Fällen sind die Zeichen 9 - 16 mit 00000000 belegt.

Zeichen 9 - 16	Bedeutung
00000001	Der Dateiname für eine Ausgabedatei ist unzulässig.
00000002	Eine Ausgabedatei konnte nicht eingerichtet werden.
00000003	Eine Ausgabedatei konnte nicht geöffnet werden.
00000004	Eine Ausgabedatei konnte nicht geschrieben werden.

Das Insert &DMSE enthält den abdruckbaren DMS-Errorcode, falls bei der Dateibearbeitung ein Fehler aufgetreten ist und in den Zeichen 9 - 16 des Inserts &IDEFRC einer der Werte 00000002, 00000003 oder 00000004 steht.

Das Insert &FNAM enthält den Namen der Datei, bei der der Fehler aufgetreten ist.

K152 Heuristikmeldung: &COND &MTYPE &OSLPAP &USER <AC &AAIS &AAID

Die Inserts haben folgende Bedeutung

&COND	Bedeutung
MIX	Der Server hat eine Dateninkonsistenz gemeldet. Bei einer asynchronen Nachricht an eine UTM-Anwendung ist dieser Fall nicht möglich.

&COND	Bedeutung
HAZ	Eine Dateninkonsistenz ist möglich. Die Verbindung zum OSI TP-Auftragnehmer (Subordinate) ging verloren, nachdem ein PREPARE gesendet wurde und der OSI TP-Auftraggeber (Superior) noch keine Datensicherung durchgeführt hat. Beim Auftraggeber wird die Transaktion zurückgesetzt, eine asynchrone Nachricht erneut gesendet. In einem UTM-Auftragnehmer wird die Transaktion ebenfalls zurückgesetzt, bzw. eine asynchrone Nachricht wird verworfen (sofort nach dem Wiederanlauf), so dass die Daten in einem konsistenten Zustand sind.

&MTYPE	Bedeutung
DIAL	Es wurde eine Dialog-Nachricht gesendet.
ASYN	Es wurde eine asynchrone Nachricht gesendet.

&OSLPAP: Name der Partner-Anwendung.

&USER: Name des Benutzers, der den Auftrag erteilt hat.

<AC: bei einer asynchronen Nachricht: LTAC-Namen des Auftrags
bei einer Dialog-Nachricht: TAC der Transaktion, für die eine Inkonsistenz möglich ist.

&AAIS: Größe des Atomic Action Identifiers in Bytes.

&AAID: enthält den encodierten Atomic Action Identifier (max. 64 Bytes).

K154 Netzmeldung: **&PTRM/&PRNM/&BCAP/<RM &TCPCL &TCPRC**

Dabei haben die Inserts der Meldung folgende Bedeutung:

Insert	Bedeutung
&TCPCL	interne Socket-Funktion oder internes Socket-Event
&TCPRC	Diagnostic Word

&TCPCL wird abdruckbar ausgegeben:

&TCPCL	Bedeutung
ATTACH	Erzeugen eines Sockets
CONNECTION_REQ	Verbindungsaufbau anfordern
CONNECTION_RSP	Verbindungsaufbaubestätigung
DISCONNECT_REQ	Verbindung abbauen
DATA_REQ	Daten senden

&TCPCL	Bedeutung
DATA_IND	Daten empfangen
GET_HOST_BY_ADDR	Rechnername zu einer IP-Adressen bestimmen
GET_HOST_BY_NAME	IP-Adresse eines Rechners bestimmen

Das Diagnostic Word besteht aus 4 Byte und wird Hexadezimal angegeben. Dabei haben die Werte folgende Bedeutung:

&TCPRC	Bedeutung	Ursache / Maßnahme
00000008	ungültige Parameter	
00000014	Connection Letter zu lang	
00000028	Beim senden von Daten war die vorausgegangene Sendeaktion noch nicht beendet.	normales Verhalten
00000030	interner Fehler	s. WAIT41 bei K060 Meldung
00000038	Beim Verbindungsauf/-abbau war die vorausgegangene Sendeaktion noch nicht beendet	normales Verhalten
0400001C	Betriebsmittelengpass	UTM-Generierung überprüfen
04000020	Anwendung ist nicht angemeldet	UTM-Generierung überprüfen
04000024	Verbindung bereits abgebaut	normales Verhalten
08000024	Verbindung bereits aufgebaut	normales Verhalten
10000024	UTM Netzanbindung noch nicht aktiv	normales Verhalten
1C080000	Anwendungsende durch das Dienstprogramm KDCSHUT mit Zeitangabe	normales Verhalten
1C0C0000	Anwendungsende durch das Dienstprogramm KDCSHUT ohne Zeitangabe	normales Verhalten
24000004	Verbindung bereits abgebaut	normales Verhalten
24000018	negative Antwort auf Verbindungsaufbauwunsch	normales Verhalten
24000038	Verbindungsabbauanforderung	normales Verhalten
30000020	Fehler beim Anmelden	
FF013008	Beim Ermitteln der IP-Adresse eines Rechners wurden nur Leerzeichen übergeben	UTM-Generierung überprüfen
FF023008	Die IP-Adresse des Rechners konnte nicht ermittelt werden	UTM-Generierung überprüfen

K155 Die Passwortgültigkeit ist abgelaufen - Bitte Passwort eingeben und ändern

K156 UTM-UPIC Verschlüsselungskomponente verfügbar: &RSLT

K158 Inkonsistenter CPU-Verbrauchswert: &CPUTEXT

Bei der Berechnung der für den Benutzer aufgelaufenen CPU-Zeit ergibt sich ein inkonsistenter Wert.

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
PTRM	Name des aktuellen PTERMs.
PRNM	Name des aktuellen Prozessors.
BCAP	Name der aktuellen BCAM-Anwendung.
LTRM	Name des aktuellen LTERMs.
USER	Name des aktuellen Benutzers.
CPUTEXT	NEGATIVE: ein negativer Wert wurde beobachtet. OVERFLOW: die für den Benutzer aufgelaufene Zeit wurde zu groß.
CPUBEGIN	CPU-Verbrauchswert am Anfang des beobachteten Zeitraums.
CPUEND	CPU-Verbrauchswert am Ende des beobachteten Zeitraums.
CPUUSED	Berechneter CPU-Verbrauchswert für den beobachteten Zeitraum.
CPUCLNT	Aufgelaufener CPU-Verbrauchswert für den aktuellen Benutzer.
CPUREAS	Interner Diagnosewert.

K159 Passwort-Änderung fuer Benutzer &USER durchgeführt

Für den angegebenen Benutzer wurde das Passwort geändert. Die Meldung wird bei jeder Passwort-Änderung gesendet, unabhängig davon, wie das Passwort geändert wurde (administrativ, über SIGN CP, ...).

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
USER	Name des Benutzers, dessen Passwort geändert wurde.
ENCPW	Geändertes Passwort in verschlüsselter Form

Diese Meldung kann dazu dienen, geänderte Benutzerkennworte in eine Standby-Anwendung zu übertragen. Dazu muss im privaten Meldungsmodul das Meldungsziel MSGTAC oder USER-DEST für diese Meldung definiert werden. Das neue Benutzerkennwort kann mittels programmierter Administration KC_MODIFY_OBJECT, Objekttyp USER in verschlüsselter Form durch Setzen von pw_encrypted='Y' in der Standby-Anwendung gesetzt werden.

Hinweise:

- Es ist zu beachten, dass im MSGTAC-Teilprogramm keine UTM-D-Kommunikation erlaubt ist. Das geänderte Passwort kann somit nicht direkt aus dem MSGTAC-Teilprogramm über eine UTM-D-Verbindung an die StandBy-Anwendung gesendet werden.
- Wenn in der Produktiv-Anwendung der Anwendungslauf beendet wird, und wenn in dieser Zeit noch Passwort-Änderungen erfolgen, dann ist es möglich, dass die entsprechenden Meldungen an das MSGTAC- Programm nicht mehr empfangen werden, sondern verloren gehen.
- UTM erzeugt die neue Meldung bei jeder Art der Passwort-Änderung, also insbesondere auch dann, wenn das Passwort in der StandBy-Anwendung geändert wird. Es ist darauf zu achten, dass die StandBy-Anwendung das Passwort nicht wieder an die Produktiv-Anwendung überträgt, damit kein Ping-Pong-Effekt entsteht.
- Es ist zu beachten, dass UTM auf verschiedenen Plattformen verschiedene Algorithmen zur Verschlüsselung verwendet; somit kann die Funktion nicht plattformübergreifend eingesetzt werden.
- Die Passworte sollten in den beteiligten Anwendungen mit derselben Komplexitätsstufe generiert sein, weil UTM beim Eintragen des verschlüsselten Passworts die Komplexität nicht prüft.

K160 Die &TACNTR. Transaktion des Vorgangs &TCVG wurde durch &RBCAUSER zurueckgesetzt (&RCCC/&RCDC); (pid: &PID)

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
PTRM	Name des aktuellen PTERMs
PRNM	Name des aktuellen Prozessors
BCAP	Name der aktuellen Anwendung
LTRM	Name des aktuellen LTERMs
USER	Name des aktuellen Benutzers
TCVG	Name des Vorgangs-TACs in dem die Transaktion zurückgesetzt wurde
TAC	Name des TACs, in dem die Transaktion zurückgesetzt wurde.
TACNTR	Nummer der zurückgesetzten Transaktion innerhalb des Vorgangs

Insert	Bedeutung
RBCAUSER	<p>Verursacher des Rücksetzens. Es gibt es folgende Werte:</p> <ul style="list-style-type: none"> – User PEND RS: Das Rücksetzen wurde durch einen PEND RS im Teilprogramm ausgelöst. – System PEND RS: Das Rücksetzen wurde durch einen von UTM initiierten PEND RS ausgelöst. – User PGWT RB: Das Rücksetzen wurde durch einen PGWT RB im Teilprogramm ausgelöst. – System PGWT RB: Das Rücksetzen wurde durch einen von UTM initiierten PGWT RB ausgelöst. – User PEND ER: Das Rücksetzen wurde durch einen PEND ER/FR im Teilprogramm ausgelöst. – System PEND ER: Das Rücksetzen wurde durch einen von UTM initiierten PEND ER ausgelöst. – DBSTATUS ERROR: Fehler bei der Datenbankstatus-Abfrage während der Knoten-Recovery. – DBSTATUS OPEN: Das Rücksetzen wurde im Warmstart wegen nicht erfolgreicher Datenbankstatus-Abfrage ausgelöst. – DB CONFIRM PTC: Fehler beim XA Datenbankaufruf CONFIRM PREPARE TO COMMIT STATE während der Knoten-Recovery. – DB CATA VTV: Fehler beim Datenbankaufruf CANCEL TRANSACTION während der Knoten-Recovery. – JOURNAL: Das Rücksetzen wurde im Warmstart wegen nicht vollständig geschriebenem Journalsatz ausgelöst.
RCCC	Wert des kompatiblen Returncodes KCRCCC
RCDC	Wert des inkompatiblen Returncodes KCRCDC
PID	pid des Prozesses

K161 Synchroner Periodic Write gestartet fuer Task &PID

K162 Lange IO fuer Task &PID: &IOPG Seiten, &IOMS Millisekunden

K163 Langer Periodic Write fuer Task &PID: &IOPG Seiten, &IOMS Millisekunden

K169 Aktion: &ACTION; Knoten: &NNM1/&HST1/&STATE; ueberwachender Knoten: &NNM2/&HST2; ueberwachter Knoten: &NNM3/&HST3

Die Meldung wird beim Anmelden und Abmelden eines Knotens an einem Cluster ausgegeben, sowie beim Ausfall eines Knotens. Sie gibt Informationen zu den Überwachungsbeziehungen in der UTM-Cluster-Anwendung aus.

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
&ACTION	beschreibt die Situation, in der die Meldung ausgegeben wird. Es enthält einen der folgenden Werte: <ul style="list-style-type: none"> - ADD - REMOVE - CHANGE - FAIL
&NNM1, &NNM2, &NNM3	enthalten die Knoten-Namen.
&HST1, &HST2, &HST3	enthalten die Rechnernamen.
&STATE	enthält den Status des Knotens, der hinzugefügt bzw. entfernt wurde. Beim Hinzufügen wird der alte Status dieses Knotens ausgegeben, beim Entfernen der neue Status dieses Knotens. Das Insert &STATE kann folgende Werte enthalten: <ul style="list-style-type: none"> G: generated Knoten-Anwendung war noch nie gestartet R: registered Knoten-Anwendung ist aktiv T: terminated Knoten-Anwendung wurde normal beendet A: abterm Knoten-Anwendung wurde abnormal beendet F: failure Knoten-Anwendung als ausgefallen markiert

K170 Cluster: Neue KDCFILE erkannt; Erzeugungszeit: &DTTM, Status: &STATE

Die Meldung wird ausgegeben, wenn anhand der Erzeugungszeit eine neu generierte KDCFILE erkannt wurde, die als Basis für zukünftige Starts von Anwendungsinstanzen definiert wird.

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
&DTTM	Generierungszeit der neuen KDCFILE
&STATE	Start-Typ der lokalen Anwendung. Das Insert &STATE kann folgende Werte enthalten: G: generated Knoten-Anwendung war noch nie gestartet R: registered Knoten-Anwendung ist aktiv T: terminated Knoten-Anwendung wurde normal beendet A: abterm Knoten-Anwendung wurde abnormal beendet F: failure Knoten-Anwendung als ausgefallen markiert

K174 Inkompatible Aenderung der Konfiguration: &DIA1

Die Konfiguration der startenden Knoten-Anwendung passt nicht zur Konfiguration der laufenden UTM-Cluster-Anwendung.

Das Insert &DIA1 beschreibt die Fehlerursache:

- Werte ≤ 10 : Fehler in der Laufzeit-Konfiguration
- Werte > 10 : Fehler in der KDCDEF-Konfiguration

Werte des Inserts &DIA1 und deren Bedeutung:

&DIA1	Bedeutung
1	Mischung von System-Protokolldatei SYSLOG als einfache Datei und als Datei-generationsgruppe FGG
2	Die Verschlüsselungsfähigkeit der startenden Knoten-Anwendung stimmt nicht mit der Verschlüsselungsfähigkeit der laufenden Knoten-Anwendungen überein
3	Die Version des Anwendungsprogramms der startenden Knoten-Anwendung stimmt nicht mit der Version des Anwendungsprogramms der laufenden Knoten-Anwendungen überein
11	Mischung aus UTM-F- und UTM-S-Anwendungen (MAX APPLIMODE)
12	Mischungen aus einfacher und doppelter Dateiführung (zweiter Parameter von MAX KDCFILE, MAX USLOG)
13	Mischung aus Anwendungen mit und ohne Benutzern
14	Mischung aus Anwendungen mit und ohne erlaubter Mehrfach- Anmeldung (SIGNON MULTI-SIGNON)
16	Die Anzahl der Datenbank-Systeme wurde verändert. (DATABASE, RMXA)

&DIA1	Bedeutung
17	Die Reihenfolge der Datenbank-Anweisungen (DATABASE) wurde geändert.
18	Die Passwort-Historie wurde geändert (SIGNON PW-HISTORY)
19	Die Anzahl der LSSBs wurde verändert (MAX LSSB)
20	Die Anzahl der GSSBs wurde verändert (MAX GSSB)
21	Die Anzahl der ULS wurde verändert (ULS)
22	Ein ULS der alten KDCFILE existiert in der neuen KDCFILE nicht mehr.
23	Die Maximalzahl von Vorgängen, die ein Benutzer kellern darf, wurde verringert (MAX NRCONV)
24	Die Maximalzahl von Asynchron-Vorgängen, die gleichzeitig offen sein dürfen, wurde verringert (2. Wert bei MAX ASYNTASKS)
25	Die maximale Länge der Ausweisinformation wurde verringert (MAX CARDLTH)
26	Die maximale Länge des Kerberos-Prinzipals wurde verringert (MAX PRINCIPAL-LTH)
27	Die Größe des Pagepool wurde verkleinert (1. Parameterwert bei MAX PGPOOL)
28	Die Größe des prozessspezifischen Puffers für die Wiederanlaufdaten wurde verkleinert (2. Parameterwert bei MAX RECBUF)
29	Die Länge des Kommunikationsbereichs wurde verändert (MAX KB)
30	Die Länge des Standard Primären Arbeitsbereichs wurde verringert (MAX SPAB)
31	Die Länge des Nachrichtenbereichs wurde verringert (MAX NB)
32	Die maximale Länge von physikalischen Ausgabe-Nachrichten wurde verändert (MAX TRMSGLTH)
33	Die maximale Länge der Benutzerdaten in LPUT-Sätzen wurde verringert (MAX LPUTLTH)
34	Der Wert des Parameters ABORT-BOUND-SERVICE der Anweisung CLUSTER wurde geändert.

K175 Datei &FNAM erzeugt

K176 Prozedur &PRCN gestartet. &MSG2 RC = &RCHX

Die konfigurierte Prozedur wurde beim Erkennen eines Ausfalls einer Knoten-Anwendung gestartet.

Das Insert &PRCN enthält die gestartete Prozedur einschließlich der übergebenen Parameter.

Das Insert &MSG2 enthält zusätzliche Return-Information.

Werte des Inserts &RCHX1 und deren Bedeutung:

&RCHX1	Bedeutung
0	Die Prozedur konnte erfolgreich gestartet werden
-1	Die Prozedur konnte nicht gestartet werden. &MSG2 enthält zusätzliche Informationen.

K178 Cluster Journal Dateien: Aktion &ACTION; Stati (&STATE,&STA2); belegte Seiten (&PGS1,&PGS2); Umschaltvorgaenge &SWNR

Diagnoseinformation zu den beiden Dateien des Administrations-Journals wird ausgegeben.

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung
&ACTION	<p>Aktion des Administrations-Journals:</p> <p>CHANGE Die Datei des Administrations-Journals, in die aktuell nicht geschrieben wird, wurde von allen laufenden Knoten-Anwendungen vollständig eingearbeitet und eine Online-Kopie der Verwaltungsdaten wurde gezogen; die Einträge der Datei werden nicht mehr benötigt.</p> <p>CREATE Die Dateien des Administrations-Journals wurden (neu) erzeugt. Dies erfolgt beim Start der ersten Knoten-Anwendung, wenn die Knoten-Anwendung zuvor neu generiert wurde.</p> <p>EXTEND Die Datei des Administrations-Journals, in die aktuell geschrieben wird, wurde vergrößert, da die andere Datei des Administrations-Journals noch nicht von allen laufenden Knoten-Anwendungen eingearbeitet wurde oder die Online-Kopie der Verwaltungsdaten noch nicht abgeschlossen wurde.</p> <p>OPEN Die Dateien des Administrations-Journals wurden geöffnet. Wird beim Start jedes UTM-Prozesses ausgegeben.</p> <p>SWITCH Die Datei des Administrations-Journals, in die aktuell geschrieben wird, wurde gewechselt.</p>

Insert	Bedeutung
&STATE / &STA2	Status einer Datei des Administrations-Journals: Zu einer Zeit darf nur eine Datei Status C haben, die andere Datei darf Status O oder R haben. C (Current) In diese Datei wird aktuell geschrieben. O (Old) Diese Datei wird noch von laufenden Knoten-Anwendungen eingearbeitet oder die Online-Kopie der Verwaltungsdaten ist noch nicht abgeschlossen. R (Reusable) Die Daten dieser Datei werden nicht mehr benötigt und können nach Umschalten der Dateien des Administrations-Journals überschrieben werden.
&PGS1 / &PGS2	Gültige Seiten einer Datei des Administrations-Journals: Gibt die Anzahl der logisch gültigen Seiten einer Datei des Administrations-Journals aus. Nach dem Umschalten der Datei, in die aktuell geschrieben wird, enthält diese nur eine gültige Seite: Die erste Seite mit Kontrollinformation. Weitere Seiten mit alten ungültigen Journalsätzen werden nicht physikalisch freigegeben, sondern bleiben belegt.
&SWNR	Anzahl der Umschaltungen des Administrations-Journals (ACTION SWITCH)

- K179** Die zu importierenden Daten wurden vollständig vom Online Import importiert
- K180** Die zu importierenden Daten wurden teilweise vom Online Import importiert
- K181** Mit der Datei &FNAM wurde noch keine Anwendung gestartet, somit sind keine Daten fuer den Online Import vorhanden.
- K182** Abnormale Beendigung des Online Import aus Datei &FNAM.
- K183** Normale Beendigung des Online Import aus Datei &FNAM.
- K184** Es ist bereits ein Online Import oder ein KDCUPD-Lauf fuer die alte KDCFILE aktiv.
- K185** Beim Online Import stimmen die Generierungszeiten der KDCFILES nicht ueberein
- K186** Die Quell-Anwendung gehoert nicht zu dieser Cluster-Anwendung.
- K187** Start des Online Import aus Datei &FNAM.
- K188** Journal Erzeugungszeit ist unterschiedlich. In &OBJ1: &DTTM; in &OBJ2: &DTM2
- Für UTM-Cluster-Anwendungen wird diese Warnmeldung zur Diagnose ausgegeben, wenn in Dateien unterschiedliche Erzeugungszeiten für das Administrations-Journal gefunden wurden.

Die Meldung besitzt folgende Inserts:

Insert	Bedeutung	
&OBJ1 / &OBJ2	Dateikennzeichen.	
	jkaa	JKAA-Datei
	journal	JRN1-/JRN2-Datei
	kdcfile	KDCFILE
&DTTM / &DTM	Journal-Erzeugungszeitpunkt, der in Datei 1 bzw. Datei 2 gefunden wurde.	

K189 Anmeldung abgelehnt - Vorgangswiederanlauf in Knoten-Anwendung auf Rechner &HST1 erforderlich

K190 Diagnosehinweis &DIA1 zu der Cluster &SUFF-Datei: &INF1: &INF2

Das Insert &SUFF bezeichnet die Datei, auf die sich die Meldung bezieht.

Das Insert &DIA1 beschreibt die Fehlerursache; die Inserts &INF1 und &INF2 geben zusätzliche Informationen.

Die Spalte **Grp.** (Gruppe) in den folgenden Tabellen beschreibt, welcher Ursachen-Gruppe der Fehlercode angehört. Es gibt folgende Gruppen:

- A Ursache ist ein Anwenderfehler, z.B. ein Fehler beim
 - Generieren und Administrieren von UTM-Anwendungen
 - Betreiben von UTM-Anwendungen
 - Generieren des Systems (z.B. Aufteilung des Adressraums)
- U Ursache ist ein Fehler im UTM-Code.
- S Ursache ist ein Fehler in einer anderen Systemkomponente (Software oder Hardware).
- M Ursache ist ein Speicherengpass.
- I Die Meldung dient zur Information.

K190-Meldung für die Cluster-Konfigurationsdatei

Ist &DIA1 im Bereich 1000 bis 1099, dann ist &SUFF gleich 'CFG' und die Meldung bezieht sich auf die Cluster-Konfigurationsdatei.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-Konfigurationsdatei und deren Bedeutung, sowie Werte und Bedeutung der Inserts &INF1 und &INF2 in Abhängigkeit von &DIA1:

&DIA1	Grp.	Bedeutung	&INF1	&INF2
1001	A	APPLNAME_MISMATCH Die Anwendungsnamen in Cluster-Konfigurationsdatei und KDCFILE stimmen nicht überein.	Anwendungsname aus der Cluster-Konfigurationsdatei	Anwendungsname aus der KDCFILE
1002	A	HOSTNAME_NOT_FOUND Der eigene Rechnername wurde in der Cluster-Konfigurationsdatei nicht gefunden.	Lokaler Host-Name	Host-Name des Cluster-Knotens oder Blank
1003	A	INVALID_KDCDEF_TIME Die Erzeugungszeit der KDCFILE ist zu alt; es wurde bereits eine andere Knoten-Anwendung mit neuerer KDCFILE gestartet.	KDCDEF-Zeit aus der Cluster-Konfigurationsdatei	KDCDEF-Zeit aus der KDCFILE
1004	I	OLD_STATE_INVALID (Warnung) Der alte Status des Knotens ist "registered" und es handelt sich nicht um einen Warm-Start.	G generated: Knoten war noch nie gestartet R registered: Knoten ist aktiv T terminated: Knoten wurde normal beendet A abterm: Knoten wurde abnormal beendet F failure: Knoten als ausgefallen markiert	D: Start nach KDCDEF-Lauf C: Kaltstart W: Warmstart U: Start nach KDCUPD-Lauf
1006	A	CLUSTER_FILE_CHANGED Die Cluster-Konfigurationsdatei wurde während des Anwendungslaufs neu erzeugt.	Generierungszeit aus der Cluster-Konfigurationsdatei oder KDCFILE-Filebase aus der Cluster-Konfigurationsdatei	Generierungszeit aus der KDCFILE oder KDCFILE-Filebase aus der KDCFILE
1008	U	NODE_NOT_REGISTERED Die lokale Anwendung ist im KAA nicht als registriert eingetragen.		

&DIA1	Grp.	Bedeutung	&INF1	&INF2
1009	A	PLATFORM_MISMATCH Die Betriebssystemplattform der lokalen Anwendung stimmt nicht überein mit der Information aus der Cluster-Konfigurationsdatei.	OS-Typ oder Bit-Mode des lokalen Rechners	OS-Typ oder Bit-Mode aus der Cluster-Konfigurationsdatei
1010	AS	NODE_START_SERIALIZATION Die Sperre zur Serialisierung des Starts verschiedener Knoten-Anwendungen konnte nicht initialisiert oder nicht gesetzt werden.	INIT: Fehler beim Initialisieren der Sperre LOCK: Fehler beim Setzen der Sperre	1: Timeout 2: Locked 3: Try later
1014	A	INVALID_FILE_VERSION Die Datei-Version der Cluster-Konfigurationsdatei passt nicht zum UTM-Systemcode.	Erwartete Version der Cluster-Konfigurationsdatei	Tatsächliche Version der Cluster-Datei
1015	ASU	CLUSTER_FILE_CORRUPTED Die Beginn- und Ende-Marker der Cluster-Konfigurationsdatei konnten nicht gefunden werden; vermutlich handelt es sich bei der Datei um keine Cluster-Konfigurationsdatei.		
1016	A	NODE_KDCFILE_MISMATCH Mit dieser KDCFILE wurde schon eine Knoten-Anwendung auf einem anderen Rechner gestartet.	Rechnername aus der Cluster-Konfigurationsdatei, auf dem mit dieser KDCFILE schon gestartet wurde.	Rechnername aus der Cluster-Konfigurationsdatei, auf dem mit dieser KDCFILE nun gestartet werden soll.
1017	A	NODE_GT_KDCFILE_START Die Knoten-Anwendung wurde zuletzt nicht mit dieser KDCFILE gestartet.	Zeitpunkt des letzten Starts mit dieser KDCFILE	Zeitpunkt des letzten Starts dieser Knoten-Anwendung.
1018	A	NODE_START_GT_KDCFILE_DEF Die neu generierte KDCFILE wurde schon vor dem letzten Start der Knoten-Anwendung generiert.	Generierungszeitpunkt der KDCFILE	Zeitpunkt des letzten Starts dieser Knoten-Anwendung.
1019	A	NODE_NAME_OF_OWN_HOST Die Knoten-Recovery der Knoten-Anwendung des eigenen Rechners ist nicht erlaubt.		
1020	A	NODE_NAME_NOT_FOUND Knotenname für Knoten-Recovery existiert nicht in der Cluster-Konfigurationsdatei.	Knotenname in der Cluster-Konfigurationsdatei	Knotenname in der KDCFILE
1021	A	NODE_NAME_INCONSISTENCY Die Reihenfolge der Knotennamen in der Cluster-Konfigurationsdatei und in der KDCFILE stimmt nicht überein.		

K190-Meldung für die Cluster-User-Datei

Ist &DIA1 im Bereich 1100 bis 1199, dann ist &SUFF gleich 'USER' und die Meldung bezieht sich auf die Cluster-User-Datei. Die Meldung kann in diesem Fall auch vom Dienstprogramm KDCDEF ausgegeben werden.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-User-Datei und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1101	AS	TIMEOUT_FOR_USERFILE Beim Öffnen oder Schließen der Cluster-User-Datei kam es bei Anforderung der Datei Sperre zu einem Timeout.
1102	A	NO_USER_FILE Die erste Seite der Datei ist keine CONS Seite einer Cluster-User-Datei.
1103	A	INVALID_FILE_VERSION Die Versionsnummer in der Cluster-User-Datei passt nicht zu der Versionsnummer im UTM-Systemcode.
1104	A	APPLNAME_MISMATCH Der Anwendungsname in der Cluster-User-Datei ist ungleich dem Anwendungsname aus der KDCFILE.
1105	A	GEN_TIME_MISMATCH Die Angaben in der Cluster-User-Datei und in der KDCFILE zu der Generierungszeit der Cluster-User-Datei stimmen nicht überein.
1106	A	CLUSTER_FILEBASE_MISMATCH Die Angaben in der Cluster-User-Datei und in der KDCFILE zu der Cluster Filebase stimmen nicht überein.
1107	ASU	LAST_PAGE_CORRUPTED Die letzte Seite der Datei ist keine CONS Seite einer Cluster-User-Datei
1108	U	PAGE_COUNTERS_INCONSISTENT Die Zähler zu freien und benutzten Einträgen in der Cluster-User-Datei sind inkonsistent.
1109	U	FREE_ANCHOR_INCONSISTENT Die Felder zur Verwaltung der Freikette in der Cluster-User-Datei sind inkonsistent.
1110	U	ENTRY_NOT_FOUND Beim Adressieren eines Eintrags in der Cluster-User-Datei ist ein Fehler aufgetreten.
1111	U	FREE_ANCHOR_BROKEN Die Verkettung des ersten Eintrags in der Freikette ist inkonsistent.

&DIA1	Grp.	Bedeutung
1112	U	USER_STATE_INVALID Der Status eines User Eintrags in der Freikette ist ungültig.
1113	U	ENTRY_NOT_FREE Ein Eintrag in der Freikette ist nicht als frei gekennzeichnet.
1114	U	END_OF_CHAIN_NOT_FOUND Beim Durchsuchen der Freikette wurde das letzte Element nicht gefunden.
1115	U	FREE_CHAIN_BROKEN Die Anzahl der Elemente in der Freikette unterscheidet sich von dem Zähler in der Verwaltungsseite.
1116	AU	MAX_FILE_SIZE_REACHED Die Cluster-User-Datei kann nicht mehr erweitert werden, da sie bereits die maximale Anzahl von Einträgen enthält.
1118	A	VERSION_MISMATCH Die Cluster-User-Datei und die KDCFILE stammen von verschiedenen UTM-Versionen.
1119	A	OS_TYPE_MISMATCH Die Cluster-User-Datei und die KDCFILE unterscheiden sich bzgl. OS-Typ und oder Bit-Modus.

K190-Meldung für die Cluster-JKAA-Datei

Ist &DIA1 im Bereich 1200 bis 1399, dann ist &SUFF gleich 'JKAA' und die Meldung bezieht sich auf die Cluster-JKAA-Datei.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-JKAA-Datei und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1202	AU	FILE_IS_EMPTY Die JKAA-Datei ist leer.
1203	AU	FILE_DOES_NOT_EXIST Die JKAA-Datei existiert nicht.
1204	AU	DMS_ERROR DMS-Fehler für die JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.
1205	M	MEMORY_INSUFFICIENT Fehler beim Allokieren von Speicher für FCB und Lese-Puffer. Maßnahme: Virtuellen Speicher vergrößern.
1206	AU	FILE_OPEN_ERROR DMS-Fehler beim Öffnen der JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.

&DIA1	Grp.	Bedeutung
1207	ASU	LOCK_INIT_TIMEOUT Timeout bei Initialisierung des Datei-Locks (KCCGFLI).
1208	SU	LOCK_INIT_ERROR Fehler bei Initialisierung des Datei-Locks (KCCGFLI).
1209	ASU	FILE_LOCK_TIMEOUT Timeout beim Anfordern des Datei-Locks (KCCGFLK).
1210	SU	FILE_LOCK_ERROR Fehler beim Anfordern des Datei-Locks (KCCGFLK).
1211	AU	INVALID_FILE_TYPE Die Datei ist keine JKAA-Datei.
1212	AU	FILE_CORRUPTED Die zweite CONS-Seite hat ein ungültiges Format.
1213	ASU	FILE_UNLOCK_TIMEOUT Timeout bei Freigabe des Datei-Locks (KCCGFLK).
1214	SU	FILE_UNLOCK_ERROR Fehler bei Freigabe des Datei-Locks (KCCGFLK).
1215	ASU	DESTROY_LOCK_TIMEOUT Timeout beim Zerstören des Datei-Locks (KCCGFLK).
1216	SU	DESTROY_LOCK_ERROR Fehler beim Zerstören des Datei-Locks (KCCGFLK).
1217	SU	FILE_CLOSE_ERROR DMS-Fehler beim Schließen der JKAA-Datei.
1218	AU	READ_FILE_ERROR DMS-Fehler beim Lesen der KAA-Seiten der JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.
1219	A	VERSION_MISMATCH Die UTM-Versionen von KDCFILE und JKAA-Datei stimmen nicht überein. Insert &INF1 enthält die UTM-Version der KDCFILE. Insert &INF2 enthält die UTM-Version der JKAA-Datei.
1220	A	OS_TYPE_MISMATCH Die JKAA-Datei und die KDCFILE unterscheiden sich bzgl. OS-Typ und/oder Bit-Modus. Insert &INF1 enthält die OS-Version der KDCFILE. Insert &INF2 enthält die OS-Version der JKAA-Datei.
1221	A	APPLINAME_MISMATCH Die Anwendungsnamen von KDCFILE und JKAA-Datei stimmen nicht überein. Insert &INF1 enthält den Anwendungsnamen der KDCFILE. Insert &INF2 enthält den Anwendungsnamen der JKAA-Datei.

&DIA1	Grp.	Bedeutung
1223	A	<p>DATETIME_NOT_EQUAL</p> <p>Die Generierungszeit des KAA der JKAA-Datei stimmt nicht mit der Generierungszeit des KAA der KDCFILE überein, obwohl die startende Knoten-Anwendung dieselbe Generierungszeit hat wie die laufende UTM-Cluster-Anwendung Insert &INF1 enthält den Generierungszeitpunkt des KAA der KDCFILE. Insert &INF2 enthält den Generierungszeitpunkt des KAA der JKAA-Datei.</p>
1224	A	<p>DATETIME_NOT_LATER</p> <p>Die Generierungszeit des KAA der KDCFILE ist nicht später als die Generierungszeit des KAA der JKAA-Datei, obwohl die startende Knoten-Anwendung neu generiert wurde. Insert &INF1 enthält den Generierungszeitpunkt des KAA der KDCFILE. Insert &INF2 enthält den Generierungszeitpunkt des KAA der JKAA-Datei.</p>
1225	A	<p>KCSANT_ERROR</p> <p>Beim Aufnehmen eines zu erzeugenden KSET in den Objektbaum der lokalen Anwendung hat die Komponente KCSANT einen schlechten Returncode geliefert.</p>
1226	A	<p>OBJECT_TYPE_ERROR</p> <p>Beim Erzeugen eines Benutzers (USER) bzw. einer LU6.1-Session in der lokalen Anwendung wurde festgestellt, dass unter dem Index in der lokalen Anwendung schon eine LU6.1-Session bzw. ein Benutzer existiert.</p>
1301	A	<p>BLKSIZE_ERROR</p> <p>Die Blocksize der KDCFILE ist ungleich 4K. Maßnahme: KDCFILE mit MAX BLKSIZE=4K generieren.</p>
1302	A	<p>BUFFER_RQ_ERROR</p> <p>Fehler beim Anfordern des Schreib-/Lese-Puffers. Maßnahme: Viruellen Speicher vergrößern.</p>
1303	A	<p>FCB_RQ_ERROR</p> <p>Fehler beim Anfordern von Speicher für den FCB. Maßnahme: Viruellen Speicher vergrößern.</p>
1304	AU	<p>FILE_CMD_ERROR</p> <p>DMS-Fehler beim Einrichten der JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.</p>
1305	AU	<p>OPEN_ERROR</p> <p>DMS-Fehler beim Öffnen der JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.</p>

&DIA1	Grp.	Bedeutung
1306	U	WRITE_NSR_1_ERROR DMS-Fehler beim Schreiben der ersten NSR-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1307	U	WRITE_NSR_ERROR DMS-Fehler beim Schreiben von Folge-NSR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1308	U	WRITE_NSR_END_ERROR DMS-Fehler beim Schreiben des letzten Blocks NSR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1309	AU	PWRT_LOCK_ERROR Fehler beim Anfordern des Periodic-Write-Locks.
1310	U	READ_DSR_1_ERROR DMS-Fehler beim Lesen der ersten DSR-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1311	U	WRITE_SR_ERROR DMS-Fehler beim Schreiben von SR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1312	U	WRITE_SR_END_ERROR DMS-Fehler beim Schreiben des letzten Blocks SR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1313	U	READ_SR_1_ERROR DMS-Fehler beim Lesen der ersten SR-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1314	U	READ_SR_ERROR DMS-Fehler beim Lesen von Folge-SR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1315	U	SR_PAGE_TYPE_ERROR Falscher Seiten-Typ einer gelesenen SR-Seite. Insert &INF1 enthält den Typ der gelesenen Seite. Insert &INF2 enthält die Halfpage-Number.
1316	U	READ_DSR_ERROR DMS-Fehler beim Lesen von Folge-DSR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.

&DIA1	Grp.	Bedeutung
1317	U	DSR_PAGE_TYPE_ERROR Falscher Seiten-Typ einer gelesenen DSR-Seite. Insert &INF1 enthält den Typ der gelesenen Seite. Insert &INF2 enthält die Halfpage-Number.
1318	U	WRITE_DSR_ERROR DMS-Fehler beim Schreiben von DSR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1319	U	WRITE_DSR_END_ERROR DMS-Fehler beim Schreiben des letzten Blocks DSR-Seiten. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1320	U	READ_CONS_ERROR DMS-Fehler beim Lesen der ersten CONS-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1321	AU	WRITE_CONS_1_ERROR DMS-Fehler beim Schreiben der ersten CONS-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1322	AU	WRITE_CONS_2_ERROR DMS-Fehler beim Schreiben der letzten CONS-Seite. Insert &INF1 enthält den DMS-Fehlercode. Insert &INF2 enthält die Halfpage-Number.
1323	U	CLOSE_ERROR DMS-Fehler beim Schließen der JKAA-Datei. Insert &INF1 enthält den DMS-Fehlercode.
1324	SU	GFLI_ERROR Fehler bei Initialisierung des Datei-Locks KCCGFLI. Insert &INF1 enthält den Returncode von KCCGFLI.
1325	ASU	GFLI_TIMEOUT Timeout bei Initialisierung des Datei-Locks KCCGFLI. Insert &INF1 enthält den Returncode von KCCGFLI.
1326	SU	GFLK_LOCK_ERROR Fehler beim Anfordern des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.
1327	AU	GFLK_LOCK_TIMEOUT Timeout beim Anfordern des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.

&DIA1	Grp.	Bedeutung
1328	SU	GFLK_UNLOCK_ERROR Fehler bei Freigabe des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.
1329	ASU	GFLK_UNLOCK_TIMEOUT Timeout bei Freigabe des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.
1330	SU	GFLK_DESTROY_ERROR Fehler beim Zerstören des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.
1331	ASU	GFLK_DESTROY_TIMEOUT Timeout beim Zerstören des Datei-Locks KCCGFLK. Insert &INF1 enthält den Returncode von KCCGFLK.
1332	U	JFCT_START_ERROR Fehler beim Schreiben des Copy-Status (Started) in die Journal-Datei.
1333	U	JFCT_END_ERROR Fehler beim Schreiben des Copy-Status (Completed) in die Journal-Datei.

K190-Meldung für die Dateien des Administrations-Journals

Ist &DIA1 im Bereich 1400 bis 1499, dann ist &SUFF gleich 'JRN' und die Meldung bezieht sich auf die Dateien des Administrations-Journals.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Dateien des Administrations-Journals und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1401	AU	Eine oder beide Journal-Dateien existieren nicht. Siehe vorherige K043-Meldung(en) Maßnahme: Anwendungen neu generieren.
1402	AU	Fehler beim Öffnen einer Journal-Datei. Siehe vorherige K043-Meldung.
1403	AU	Fehler beim Erzeugen einer Journal-Datei. Siehe vorherige K043-Meldung.
1404	SU	Fehler beim Einrichten der Sperre für die Journaldateien.
1405	SU	Fehler beim Schließen einer Journal-Datei. Siehe vorherige K043-Meldung.
1406	ASU	Fehler beim Freigeben reservierter Seiten einer Journaldatei. Siehe vorherige K043-Meldung.

&DIA1	Grp.	Bedeutung
1407	A	Fehlerhafte Journaldatei(en): Falsches Kennzeichen. &INF1/&INF2: Kennzeichen in Datei mit Suffix JRN1/JRN2. Maßnahme: Dateien löschen, Anwendungen neu generieren.
1408	A	Fehlerhafte Journaldatei(en): Falsches Datei-Kennzeichen. &INF1/&INF2: Kennzeichen in Datei mit Suffix JRN1/JRN2.
1409	A	Fehlerhafte Journaldatei(en): Falscher UTM Anwendungsname. &INF1/&INF2: Anwendungsname in JRN1-Datei / KDCFILE.
1410	A	Fehlerhafte Journaldatei(en): Falscher UTM Anwendungsname. &INF1/&INF2: Anwendungsname in JRN2-Datei / KDCFILE.
1411	A	Fehlerhafte Journaldatei(en): Falsche Cluster-Filebase. &INF1/&INF2: Cluster-Filebase in JRN1-Datei / KDCFILE.
1412	A	Fehlerhafte Journaldatei(en): Falsche Cluster-Filebase. &INF1/&INF2: Cluster-Filebase in JRN2-Datei / KDCFILE.
1413	A	Fehlerhafte Journaldatei(en): Erzeugungszeit verschieden. &INF1/&INF2: Erzeugungszeit Datei mit Suffix JRN1/JRN2.
1414	AU	Fehlerhafte Journaldatei(en): Datei-Status ungültig. &INF1/&INF2: Datei-Status in Datei mit Suffix JRN1/JRN2.
1415	SU	Fehler beim Zerstören der Sperre für die Journaldateien.
1416	AU	Fehlerhafte Journaldatei(en): Falsche Dateisequenznummern. &INF1/&INF2: Dateisequenznummer in JRN1/JRN2-Datei.
1417	A	Fehlerhafte Journaldatei(en): Falsche Dateiversion. &INF1/&INF2: Version der JRN1-Datei / erwartete Version.
1418	A	Fehlerhafte Journaldatei(en): Falsche Dateiversion. &INF1/&INF2: Version der JRN2-Datei / erwartete Version.

K190-Meldung für die Verwaltungsdatei des Cluster-Pagepools

Ist &DIA1 im Bereich 1500 bis 1599, dann ist &SUFF gleich 'CPMD' und die Meldung bezieht sich auf die Verwaltungsdatei des Cluster-Pagepools.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Verwaltungsdatei des Cluster-Pagepools und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1501	A	Falsche Dateiversion der Verwaltungsdatei. &INF1/&INF2: Version der CPMD-Datei / erwartete Version Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen.
1502	A	Ein Cluster-Update mit Übertragung in die CPMD-Datei wurde abnormal beendet. Maßnahme: Cluster-Konfigurationsdateien neu erzeugen.

&DIA1	Grp.	Bedeutung
1503	A	Die Größe des Cluster-Pagepools wurde verkleinert. &INF1/&INF2: Größe in CPMD-Datei / Größe in KDCFILE. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen.
1504	A	Die Anzahl der Dateien des Cluster-Pagepools wurde geändert. &INF1/&INF2: Anzahl in CPMD-Datei / Anzahl in KDCFILE. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen.
1505	A	Fehler beim Versuch, die Dateien des Cluster-Pagepool zu vergrößern. &INF1: 4K Seitennummer, die nicht geschrieben werden konnte. &INF2: Generierte Größe des Cluster-Pagepools in 4K Seiten. Maßnahme: Genügend Plattenspeicherplatz für UTM-Cluster Dateien zur Verfügung stellen.
1506	AU	Ungültige Größe der Freiseiten-Kontingente des Cluster-Pagepools. &INF1: Kontingentgröße in Anzahl UTM-Seiten. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen oder Problem-meldung schreiben.
1507	AU	Ungültige Anzahl von Seiten der Freiseiten-Kontingente des Cluster-Pagepools. &INF1: Anzahl der Seiten aller Freiseiten-Kontingente. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen oder Problem-meldung schreiben.
1508	AU	Fehlerhafte Verwaltung der Freiseiten-Kontingente des Cluster-Pagepools. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen oder Problem-meldung schreiben.
1509	AU	Ungültiges Kennzeichen zur Datei-Vergrößerung des Cluster-Pagepools. &INF1: Kennzeichen zur Datei-Vergrößerung. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen oder Problem-meldung schreiben.
1510	AU	Ungültiges Kennzeichen zum Lauf eines Cluster-Updates mit Übertragung in die CPMD-Datei. &INF1: Kennzeichen zum Lauf des Cluster-Updates. Maßnahme: Alle UTM-Cluster-Dateien neu erzeugen oder Problem-meldung schreiben.
1511	A	MAX APPLIMODE der Knoten-Anwendung wurde geändert. Für einen Wechsel zwischen Secure und Fast müssen alle UTM-Cluster-Dateien neu erzeugt werden. &INF1/&INF2 : APPLIMODE in CPMD-Datei / in KDCFILE
1512	A	MAX VGMSIZE der Knoten-Anwendung wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : VGMSIZE in CPMD-Datei / in KDCFILE

&DIA1	Grp.	Bedeutung
1513	A	MAX KB der Knoten-Anwendung wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : KB in CPMD-Datei / in KDCFILE
1514	A	MAX NB der Knoten-Anwendung wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : NB in CPMD-Datei / in KDCFILE
1515	A	MAX LSSBS der Knoten-Anwendung wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : LSSBS in CPMD-Datei / in KDCFILE
1516	A	MAX TRMSGLTH der Knotenanwendung wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : TRMSGLTH in CPMD-Datei/KDCFILE
1517	A	Die Anzahl der generierten Datenbanken wurde geändert. Alle UTM-Cluster-Dateien müssen neu erzeugt werden. &INF1/&INF2 : Anzahl in CPMD-Datei/KDCFILE
1518	A	Unzulässige Mischung mit und ohne Benutzerkennungen. Alle UTM-Cluster-Dateien müssen neu erzeugt werden.
1520	A	Änderung von CLUSTER ABORT-BOUND-SERVICE. Alle UTM-Cluster-Dateien müssen neu erzeugt werden.
1521	A	Änderung der generierten Datenbanksysteme. Alle UTM-Cluster-Dateien müssen neu erzeugt werden.

K190-Meldung für die Cluster-ULS-Datei

Ist &DIA1 im Bereich 1600 bis 1699, dann ist &SUFF gleich 'ULS' und die Meldung bezieht sich auf die Cluster-ULS-Datei, in der in UTM-Cluster-Anwendungen die Verwaltungsdaten zu ULS-Bereichen gehalten werden.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-ULS-Datei und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1601	AU	Die Datei ist keine Cluster-ULS-Datei oder die Datei ist korrupt.
1602	A	Die Cluster-ULS-Datei hat eine ungültige Dateiversion.
1603	A	Die Anwendungsnamen aus Cluster-ULS-Datei und KAA stimmen nicht überein.
1604	AU	Die letzte Seite der Cluster-ULS-Datei ist zerstört.
1605	U	Generierungsfehler: Die Anzahl der ULS-Blöcke aus der Cluster-ULS-Datei stimmt nicht überein mit der Anzahl der ULS-Blöcke in der KDCFILE.

&DIA1	Grp.	Bedeutung
1606	U	Generierungsfehler: Ein ULS aus der Cluster-ULS-Datei fehlt in der KDCFILE.
1607	ASU	Beim Anfordern einer Sperre auf die Cluster-ULS-Datei kam es zu einem Timeout.

K190-Meldung für die Cluster-Lock-Datei

Ist &DIA1 im Bereich 1700 bis 1799, dann ist &SUFF gleich 'LOCK' und die Meldung bezieht sich auf die Cluster-Lock-Datei für die Sperren von globalen Betriebsmitteln in UTM-Cluster-Anwendungen (GSSB und ULS).

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-Lock-Datei und deren Bedeutung:

&DIA1	Grp.	Bedeutung
1700	AU	Beim Öffnen der bestehenden Cluster-Lock-Datei ist ein Fehler aufgetreten. Siehe vorherige K043-Meldung.
1701	AU	Beim Anlegen der Cluster-Lock-Datei ist ein Fehler aufgetreten. Siehe vorherige K043-Meldung.
1702	SU	Fehler beim Schreiben der Cluster-Lock-Datei. Siehe vorherige K043-Meldung.
1703	SU	Fehler beim Schließen der Cluster-Lock-Datei. Siehe vorherige K043-Meldung.
1704	SU	Fehler beim Öffnen der Cluster-Lock-Datei. Siehe vorherige K043-Meldung.
1705	SU	Fehler beim Ersetzen der Cluster-Lock-Datei. Siehe vorherige K043-Meldung.
1706	I	Informationsmeldung - Cluster-Lock-Datei wurde erweitert. Es wird die Anzahl der neuen Seiten ausgegeben..
1707	SU	Fehler beim Schließen der Cluster-Lock-Datei. Siehe vorherige K043-Meldung.

K190-Meldung für die Cluster-GSSB-Datei

Ist &DIA1 im Bereich 1800 bis 1899, dann ist &SUFF gleich 'GSSB' und die Meldung bezieht sich auf die Cluster-GSSB-Datei, in der in UTM-Cluster-Anwendungen die Verwaltungsdaten für GSSBs gehalten werden.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 für die Cluster-GSSB-Datei und deren Bedeutung, sowie Werte und Bedeutung der Inserts &INF1 und &INF2 in Abhängigkeit von &DIA1:

&DIA1	Grp.	Bedeutung	&INF1	&INF2
1800	ASU	GSSBFILE_LOCK_ERROR Beim Anfordern oder Aufheben einer Sperre auf die Cluster-GSSB-Datei kam es zu einem Fehler oder Timeout.	LOCK: Fehler beim Anfordern der Sperre UNLOCK: Fehler beim Freigeben der Sperre	TIMEOUT: Timeout (Returncode vom KCSGFLK) TRY LATER: Try later (Returncode vom KCSGFLK) LOCKED: Datei ist bereits vom eigenen Task gesperrt NOT LOCKED: Datei ist nicht gesperrt
1801	A	INVALID_FILE_VERSION Die Cluster-GSSB-Datei hat eine ungültige Dateiversion.	Dateiversion in der GSSB-Datei	Erwartete Dateiversion
1802	AU	HEADER_PAGE_CORRUPTED Die Verwaltungsseite der Cluster-GSSB-Datei ist zerstört.	Fehlerhafter Feldinhalt in der GSSB-Datei	Erwarteter Feldinhalt
1803	AU	FREE_PAGES_ANCHOR_CORRUPTED Die Freiseitenverkettung der Cluster-GSSB-Datei ist korrupt.		
1804	AU	NO_GSSBFILE Die Datei ist keine Cluster-GSSB-Datei oder die Datei ist korrupt.	Dateityp auf der CONS-Seite der GSSB-Datei	Erwarteter Dateityp
1805	AU	PAGE_COUNTER_INCONSISTENT Die Seitenzähler der Cluster-GSSB-Datei sind korrupt.		
1806	AU	LAST_PAGE_INCONSISTENCY Die Dateigröße auf CONS- und Verwaltungsseite der Cluster-GSSB-Datei stimmen nicht überein.		
1808	SU	CONS_PAGE_READ_ERROR Fehler beim Lesen der ersten CONS-Seite der Cluster-GSSB-Datei.		
1809	SU	CONS_PAGE_WRITE_ERROR Fehler beim Schreiben einer CONS-Seite der Cluster-GSSB-Datei.		

&DIA1	Grp.	Bedeutung	&INF1	&INF2
1810	SU	READ_ERROR_GSSBFILE Fehler beim Lesen einer Seite der Cluster-GSSB-Datei.		
1811	SU	WRITE_ERROR_GSSBFILE Fehler beim Schreiben einer Seite der Cluster-GSSB-Datei.		
1812	U	RELEASE_BUFFER_ERROR Fehler beim Freigeben einer Seite der Cluster-GSSB-Datei.		
1814	U	FREE_ENTRY_CHAIN_CORRUPTED Die Verkettung der freien GSSB-Einträge in der Cluster-GSSB-Datei ist korrupt.		
1815	U	HASH_CHAIN_CORRUPTED Die Hash-Verkettung der GSSB-Einträge in der Cluster-GSSB-Datei ist korrupt.		
1816	U	FREE_PAGE_CHAIN_CORRUPTED Die Freiseitenverkettung in der Cluster-GSSB-Datei ist korrupt.		
1817	U	GSSB_CNTR_ERROR Der Zähler über die belegten GSSB-Einträge in der Cluster-GSSB-Datei ist korrupt.		
1818	AU	MAX_GSSB_ERROR Der Wert von MAX GSSB in der KDCFILE stimmt nicht mit dem Wert in der Cluster-GSSB-Datei überein.	MAX_GSSB in der GSSB-Datei	MAX_GSSB in der KDCFILE

K190-Meldung für UTM-Cluster-Dateien

Ist &DIA1 im Bereich 1900 bis 1949, so bezieht sich die Meldung auf die Anforderung einer Sperre für eine UTM-Cluster-Datei und dient nur zur Diagnose.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 und deren Bedeutung.

&DIA1	Grp.	Bedeutung
1900	I	Die Lock-Anforderung konnte in der generierten Zeit (Operand CLUSTER FILE-LOCK-TIMER-SEC) nicht durchgeführt werden. Die Lock-Anforderung wird in Abhängigkeit von dem im Operanden CLUSTER FILE-LOCK-RETRY generierten Wert wiederholt. &INF1: interner Fehlercode der Lock-Anforderung.

&DIA1	Grp.	Bedeutung
1901	I	Diese Meldung kommt bei Freigabe des Datei-Locks, wenn die Sperre von diesem Prozess mindestens so lange wie die halbe generierte Zeit des im Operanden FILE-LOCK-TIMER-SEC in der CLUSTER-Anweisung generierten Wertes gehalten wurde. &INF1: Status der angeforderten Sperre. &INF2: Lock-Haltdauer des Prozesses in Sekunden.

&SUFF gibt den Suffix der UTM-Cluster-Datei an, für die die Sperre angefordert wurde. Außerdem werden unter Unix- und Linux-Systemen werden noch folgende Suffixe verwendet, die stellvertretend für reale KDCFILE-Dateien stehen:

IKDC für die importierte KDCFILE
OKDC für die eigene KDCFILE
EKDC für die KDCFILE eines anderen Knotens

K190-Meldung für Ausfall einer Knoten-Anwendung

Ist &DIA1 im Bereich 2001 bis 2099, so ist &SUFF gleich 'KDCA' und die Meldung bezieht sich auf die KDCFILE einer Knoten-Anwendung. Diese Meldung wird ausgegeben, wenn der Ausfall eines Knotens festgestellt wird.

Die folgende Tabelle gibt eine Übersicht über die Werte des Inserts &DIA1 und deren Bedeutung:

&DIA1	Grp.	Bedeutung
2001	I	Die Anwendung ist nicht aktiv.
2002	S	Beim Anfordern von Speicher kam es zu einem Fehler.
2003	ASU	Vom Dateisystem wurde ein Fehler gemeldet; siehe auch K043-Meldung
2004	SU	Beim Initialisieren des Locks kam es zu einem Fehler.
2005	U	Der Knoten konnte in der UTM-Cluster-Datei nicht gefunden werden

&INF1 enthält die Knotennummer des Knotens, dessen Ausfall erkannt wurde.

&INF2 enthält den Dateinamen der KDCFILE des Knotens, dessen Ausfall erkannt wurde.

K191 Dump wird ohne &SUFF-Datei geschrieben, da nicht genügend Speicher (&RQM Bytes) allokiert werden konnte.

K192 Knoten-Recovery fuer Knoten &NNM1 auf Rechner &HST1, RESET-PTC=&RSPTC

Diese Meldung wird zu Beginn einer Knoten-Recovery auf *stdout* und *stderr* ausgegeben.

K193 PTC gefunden : ID=&PTCID, USER=&USER, LPAP=&LPAP, LSES=&LSES, USER-Typ=&USTYPPTC

Diese Meldung wird bei einer Knoten-Recovery für jede Transaktion im Status PTC einmal ausgegeben.

K194 Nach Knoten-Recovery: Anzahl noch gesperrter GSSB: &GBLNBR, Anzahl noch gesperrter ULS: &ULLNBR

Diese Meldung wird am Ende der Knoten-Recovery ausgegeben.

Hält die ausgefallene Knoten-Anwendung, für die eine Knoten-Recovery durchgeführt wurde, noch Sperren auf GSSBs, kann dies die laufende UTM-Cluster-Anwendung beeinträchtigen.

K201 XA(&TSNPID) Resource Manager Anbindung ueber X/Open &XASPEC

&XASPEC bezeichnet die Version des XA-Anschlusses. Das Insert kann nur den Wert XA-CAE-Spec annehmen.

K202 XA(&TSNPID) RM-Instanz &INSTNUM,&TEXT32, &RMSTAT

Die Meldung informiert über den Status der XA-Verbindung.

Die Inserts haben folgende Bedeutung:

&INSTNUM &INSTNUM Laufende Nummer der Resource Manager-Instanz, beginnend mit 1. Die Nummerierung entspricht der Reihenfolge der Startparameter.

&TEXT32 Name des Resource Managers.

&RMSTAT Status der Verbindung. Folgende Werte sind möglich:

&RMSTAT	Bedeutung
opened	Der Resource Manager wurde geöffnet
reopened	Der Resource Manager wurde nach einem Wiederanlauf geöffnet
closed	Der Resource Manager wurde geschlossen

K203 XA(&TSNPID) RM &TEXT32,&INSTNUM; Wiederhergestellte Transaktionen:&RTAANZ

Die Inserts haben folgende Bedeutung:

&TEXT32 Name des Resource Managers.

&INSTNUM Laufende Nummer der Resource Manager-Instanz, beginnend mit 1. Die Nummerierung entspricht der Reihenfolge der RMXA-Startparameter.

&RTAANZ Anzahl der wiederhergestellten Transaktionen.

K204 XA(&TSNPID) Precommit erfordert generelles Ruecksetzen; Ursache:&XATXTA=&INTTAID

Precommit ist fehlgeschlagen, die globale Transaktion wird zurückgesetzt.

Die Inserts haben folgende Bedeutung:

&XATXT Grund für das Fehlschlagen des Precommit. Folgende Werte sind möglich:

&XATXT	Bedeutung
XA_RBROLLBACK	Rollback aus nicht näher spezifiziertem Grund
XA_RBCOMMFAIL	Rollback wegen eines internen Kommunikationsfehlers im Resource Manager
XA_RBDEADLOCK	Rollback wegen Deadlocks
XA_RBINTEGRITY	Rollback wegen Ressourcen-Inkonsistenz
XA_RBOTHER	Rollback aus nicht näher spezifiziertem Grund
XA_RBPROTO	Rollback wegen eines internen Protokollfehlers beim Resource Manager
XA_RBTIMEOUT	Rollback wegen Zeitüberschreitung der Transaktionsdauer
XA_RBTRANSIENT	Rollback wegen eines vorübergehenden Fehlers

&INTTAID Beschreibung der UTM-Transaktion, die das globale Rücksetzen der anderen Transaktionen ausgelöst hat.

Byte 0 enthält die Instanznummer des Resource Managers.

Wenn die XID (= XA-Transaktionsbezeichner), in einer OSI TP-Auftragnehmer-Transaktion angelegt wurde, folgt hier der AAID (Atomic Action Identifier).

Anderenfalls enthalten die nachfolgenden Bytes:

Byte	Bedeutung
1 - 8	Hostname
9 - 16	Anwendungsname
17	Typ der BS2000-Hardware-Plattform (z.B. 1 ≙ Windows, 4 ≙ SUN)
18	Bit-Modus (1 ≙ 32-Bit-Plattform, 2 ≙ 64-Bit-Plattform)
19	Byte-Format (2 ≙ Bigendian)
20	Nummer der openUTM-Session
21 - 22	Nummer der Transaktion innerhalb des Vorgangs
23 - 24	reserviertes Feld
25 - 28	Vorgangsnummer
29 - 32	Vorgangsindex

K205 XA(&TSNPID) Transaktion vorge setzt; Ursache: &XATXT
TA=&INTTAID

Die Inserts haben folgende Bedeutung:

&XATXT Grund für das Vorsetzen der Transaktion (Commitment). Mögliche Gründe: „Recovery“ oder „Int.Event“.

&INTTAID Beschreibung der UTM-Transaktion, die vorge setzt wurde (Inhalt siehe [K204](#)).

K206 XA(&TSNPID) Transaktion zurueckge setzt; Ursache: &XATXT
TA=&INTTAID

Die Inserts haben folgende Bedeutung:

&XATXT Grund für den Rollback. Mögliche Gründe: „Recovery“ oder „Int.Event“.

&INTTAID Beschreibung der UTM-Transaktion, die vorge setzt wurde (Inhalt siehe [K204](#)).

K207 XA(&TSNPID) Transaktion ist dem RM &INSTNUM nicht bekannt
TA=&INTTAID

Die Inserts haben folgende Bedeutung:

&INSTNUM Instanz-Nummer des Resource Managers, siehe [K203](#).

&INTTAID Beschreibung der UTM-Transaktion, die dem RM nicht bekannt ist
(Inhalt siehe [K204](#)).

K210 XA(&TSNPID) Returncode: &XATXT; Open RM: &TEXT32,&INSTNUM

K211 XA(&TSNPID) Returncode: &XATXT; Close RM: &TEXT32,&INSTNUM

Die Inserts der Meldungen K210, K211 und K216 haben folgende Bedeutung:

&TEXT32 Name des Resource Managers.

&INSTNUM Instanz-Nummer des Resource Managers, siehe [K203](#).

&XATXT Schlüsselwort für den Returncode der aufgerufenen XA-Funktion.
Mögliche Werte sind:

&XATXT	Bedeutung
XA_OK	Normale Ausführung
XA_NOMIGRATE	Die Transaktion wurde suspendiert, aber der Kontext kann nicht migriert und die Transaktion nur im selben Prozess fortgesetzt werden
XA_HEURHAZ	Aufgrund möglicher RM-interner Fehler kann die Transaktion heuristisch abgeschlossen worden sein
XA_HEURRB	Aufgrund einer heuristischen Entscheidung des Resource Managers wurde die Transaktion intern zurückgesetzt
XA_HEURMIX	Aufgrund einer heuristischen Entscheidung des Resource Managers wurde die Transaktion intern teilweise vor- und teilweise zurückgesetzt
XA_RETRY	Der Aufruf der Routine muss wiederholt werden
XA_RDONLY	Die Transaktion war „read_only“ und wurde vorgesetzt
XA_RBROLLBACK	Rollback aus nicht näher spezifiziertem Grund
XA_RBCOMMFAIL	Rollback wegen Kommunikationsfehlers
XA_RBDEADLOCK	Rollback wegen Deadlocks
XA_RBINTEGRITY	Rollback wegen Ressourcen-Inkonsistenz
XA_RBOTHER	Rollback aus nicht näher spezifiziertem Grund
XA_RBPROTO	Rollback wegen eines RM-internen Protokollfehlers
XA_RBTIMEOUT	Rollback wegen Zeitüberschreitung der Transaktionsdauer
XA_RBTRANSIENT	Rollback wegen vorübergehendem Fehlers

&XATXT	Bedeutung
XAER_ASYNC	Eine asynchrone Operation steht noch aus
XAER_RMERR	Im Resource Manager ist ein nicht wiederherstellbarer Fehler aufgetreten. Mögliche Ursache: Der Resource Manager wurde nicht oder falsch initialisiert
XAER_NOTA	Der Transaktions-Identifikator ist ungültig bzw. dem Resource Manager nicht bekannt
XAER_INVALID	Ungültige Funktionsargumente wurden angegeben. Möglicherweise enthalten der Open- oder Close- string falsche Parameter.
XAER_PROTO	Die Routine wurde RM-intern in unpassendem Kontext aufgerufen
XAER_RMFAIL	Der Resource Manager ist nicht mehr verfügbar
XAER_DUPID	Der Transaktions-Identifikator existiert bereits im Resource Manager. openUTM beendet den Vorgang. Zur Beseitigung dieser „XID-Leiche“ bei XAER_DUPID aus dem Resource Manager sollte der Datenbank-Administrator diese Transaktion entfernen. Mögliche Maßnahme: Datenbank herunterfahren und neu starten.
XAER_OUTSIDE	Der Resource Manager arbeitet außerhalb der Transaktion.

K212 XA(&TSNPID) xa_start(&XAFLAG) - Returncode: &XATXT
TA=&INTTAID

K213 XA(&TSNPID) xa_end(&XAFLAG) - Returncode: &XATXT
TA=&INTTAID

K214 XA(&TSNPID) xa_commit() - Returncode: &XATXT
TA=&INTTAID

K215 XA(&TSNPID) xa_rollback() - Returncode: &XATXT
TA=&INTTAID

Die Inserts der Meldungen K212 bis K215 haben folgende Bedeutung:

&XATXT	Schlüsselwort für den Returncode der aufgerufenen XA-Funktion. Mögliche Werte siehe K211 .
&XAFLAG	Schlüsselwort, das die Flags beim Aufruf von xa_start() bzw. xa_end() beschreibt.
&INTTAID	Beschreibung der UTM-Transaktion, bei der der Returncode aufgetreten ist (Inhalt siehe K204).

- K216** XA(&TSNPID) Returncode: &XATXT; Recover PTC-Liste, RM:
&TEXT32,&INSTNUM
Bedeutung und Inhalt der Inserts siehe [K211](#).
- K217** XA(&TSNPID) xa_prepare() - Returncode: &XATXT
TA=&INTTAID
Zur Bedeutung der Inserts siehe [K211](#) (&XATXT) und [K215](#) (&INTTAID).
- K218** XA(&TSNPID) xa_forget() - Returncode: &XATXT
TA=&INTTAID
Zur Bedeutung der Inserts siehe [K211](#) (&XATXT) und [K215](#) (&INTTAID).
- K220** XA(&TSNPID) Fehler: xa_switch Definition fuer spezifizierten RM nicht gefunden:
&TEXT32
Ein Resource Manager (RM), für den RMXA-Startparameter angegeben wurden, wurde bei der KDCDEF-Generierung nicht definiert.
&TEXT32 enthält den Namen des Resource Managers.
- K221** XA(&TSNPID) Fehler: Startparameter fuer definierten RM nicht gefunden:
&TEXT32
Für einen in der KDCDEF-Generierung enthaltenen Resource Manager (RM) sind keine Startparameter angegeben.
&TEXT32 enthält den Namen des Resource Managers .
- K222** XA(&TSNPID) Fehler: Gebundener RM ist nicht &XASPECkompatibel: &TEXT32
Der Resource Manager (RM) benutzt nicht die XA-Schnittstelle, die bei der KDCDEF-Generierung angegeben wurde.
&TEXT32 enthält den Namen des Resource Managers, &XASPEC die bei der Generierung angegebene XA-Schnittstelle.
- K223** XA(&TSNPID) Syntaxfehler in Startparameter:
Die folgende Zeile enthält die fehlerhafte Zeile aus der Startparameterdatei.

- K224** XA(&TSNPID) &XACALL - Returncode &XATXT der RM-Instanz &INSTNUM, &TEXT32 entspricht nicht der XA(CAE)-Spezifikation
- Der Resource Manager (RM) liefert einen unerwarteten Returncode.
- Die Inserts haben folgende Bedeutung:
- &XACALL Name des Aufrufs an der XA-Schnittstelle, z. B. xa_start().
 - &XATXT Schlüsselwort für den Returncode der aufgerufenen XA-Funktion (siehe [K211](#)).
 - &INSTNUM Instanz-Nummer des Resource Managers, siehe [K203](#).
 - &TEXT32 Name des Resource Managers.
- K225** XA(&TSNPID) rekursiver Aufruf: &XADBC1 - Fehler/Signal im DB/XA-Anschluss bei &XADBC2
- Die Inserts haben folgende Bedeutung:
- &XADBC1 Name der aufgerufenen Funktion im UTM/XA-Anschlussmodul.
 - &XADBC2 Name der Funktion im UTM/XA-Anschlussmodul, in der ein Fehler festgestellt wurde.
- K230** XA(&TSNPID) Int.Fehler: &TEXT32
- &TEXT32 spezifiziert einen internen Fehler. Dieser kann z.B. auf mangelnden Speicherplatz für malloc() hindeuten.
- K231** XA(&TSNPID) Int.Fehler: PETA wird nicht unterstuetzt
- Das vorläufige Transaktionsende (Preliminary End of Transaction) wird nicht unterstützt.
- K232** XA(&TSNPID) Int.Fehler: DBSTAT sekundaerer Operationscode inkonsistent
- Der sekundäre Operationscode des primären Operationscodes DBSTAT ist nicht konsistent.
- K233** Heur. Entsch. in Task / Prozess &TSNPID, RM=&INSTNUM in &XACALL(&DBCALL), xa-rc=&XATXT, LTHGTRID=<HGTRID, GTRID=>RID
- Die Inserts haben folgende Bedeutung:
- &INSTNUM Instanz-Nummer des Resource Managers (RM), siehe [K203](#).
 - &XACALL Name des Aufrufs an der XA-Schnittstelle, z.B. xa_start().
 - &DBCALL Name der aufgerufenen Funktion der UTM-DB-Schnittstelle
 - &XATXT Schlüsselwort für den Returncode der aufgerufenen XA-Funktion (siehe [K211](#)).

- >RID Globale Transaktions-ID (gtrid) der UTM-Transaktion entsprechend der XA/CAE-Spezifikation (Inhalt siehe [K204](#)).
- <HGTRID Länge der globalen Transaktions-ID.
- K235** Die Namensauflösung fuer &PRNM dauerte &TCPMS Millisekunden (Socket-Aufruf: &TCPCL, Returncode: &TCPRC, IP-Adresse: &IPADDR).
- K251** Version V&IMPVER of filebase &FBASUPD does not match current version V&DEFVER.
- K252** &UPDERR Fehler.
- Ist &UPDERR auf den Wert PARAM gesetzt, deutet dies auf einen Syntaxfehler hin. KDCUPD bricht ab und setzt den Schalter 3 auf ON.
Maßnahme: Richtiges Kommando eingeben!
Ansonsten ist ein Fehler bei Speicherplatzanforderung, beim Einlesen etc. aufgetreten
- Maßnahme: PM schreiben mit Unterlagen
- K255** DMS-Fehler &DMSE fuer Datei &FNAM.
- Fehler bei der Dateibearbeitung.
Maßnahme: entsprechend DMS-Fehlercode, siehe [Seite 383](#).
- K256** Datei &FNAM entspricht nicht KDCFILE - Konventionen.
- Die Datei ist nicht vom Typ KDCFILE (KDC-, Pool-, Conf-Datei)
- K257** Der Anwendungslauf wurde nicht ordnungsgemaess beendet.
- K258** Mit der Datei &FNAM ist schon gestartet worden.
- K260** Unbekannte Version &DEFVER von Filebase &FBASUPD.
- K261** Inkonsistente Datei &FNAM.
- Die Datei ist in sich nicht konsistent
Maßnahme: Datei wiederherstellen oder neu generieren
- K262** Falsche Datei &FNAM.
- Die Datei wurde durch eine andere KDCFILE-Datei überschrieben, näheres siehe stderr-Protokoll. Maßnahme: Richtige Datei bereitstellen oder neu generieren
- K263** Mit der Datei &FNAM ist noch nicht gestartet worden.
- K269** &OBJ1 &OST1 &BMD1 und &OBJ3 &OST2 &BMD2 sind nicht vertraeglich

K273 Fehler: &TRMA in Modul &UPDMODUL.

Beschreibung der Fehlerursachen bei Meldung K273 mit Maßnahmen:

&TRMA	Ursache	Maßnahme
BFMMER	Fehler beim Initialisieren des Buffer-Managements (Cluster)	ggf. virtuellen Adressraum erhöhen oder PM
CFGERR	Fehler beim Bearbeiten der Cluster-Konfigurationsdatei	vorherige Diagnose-Meldungen oder falls keine: inkonsistente Reihenfolge Knotennamen
CONSER	Interner Fehler	PM + Unterlagen
DMSERR	Ein DMS-Fehler ist aufgetreten	siehe Meldung K255
DxxxRD	DMS-Fehler xxx beim Lesen der ersten Seite	je nach DMS-Fehlercode, siehe auch Seite 383
DxxxWR	DMS-Fehler xxx beim Schreiben der ersten Seite	
xxxx01	DMS-Fehler xxxx beim Lesen der ersten KAA-Seite	
GMDTER	Fehler beim Prüfen der Cluster-GSSB-Datei	siehe vorherige Diagnose-Meldungen
INCONS	Fehler bei Konsistenzprüfung	siehe Meldung K261/262
LOOKFL LOOKFU	Fehler bei Suche nach Verursacher (LTERM, USER)	PM + Unterlagen
LOOKLT	Fehler bei Suche nach LTERM	PM + Unterlagen
LOOKTC	Fehler bei Suche nach TAC	PM + Unterlagen
LOOKUS	Fehler bei Suche nach USER	PM + Unterlagen
OPWRON	Interner Fehler	PM + Unterlagen
PPMMER	Verwaltungsdatei des Cluster Pagepools fehlerhaft.	siehe vorherige Diagnose-Meldungen
REQM01	Fehler bei Request Memory für 1. Seite	evtl. System neu generieren
REQM02	Speichermangel beim Anlegen von Shared Memory für UTM-Cache	Betriebssystem-Generierung überprüfen oder Parameter CACHESIZE kleiner machen
REQKTA	Fehler bei Request Memory KTA	siehe REQM01
SHM002	Ursache: Die UTM-Anwendung läuft gerade. Ein KDCUPD-Lauf ist zur Zeit nicht möglich.	Maßnahme: Anwendung beenden.

&TRMA	Ursache	Maßnahme
UMDTER	Fehler beim Prüfen der Cluster-ULS-Datei	siehe vorherige Diagnose-Meldungen
UPDSTA	Datei wurde bereits mit KDCUPD bearbeitet	keine
USFERR	Fehler beim Öffnen der Cluster-User-Datei	siehe vorherige Diagnose-Meldungen
...GSBGB	Fehler bei GSSB-Behandlung	PM + Unterlagen
...TLSTL	Fehler bei TLS-Behandlung	PM + Unterlagen
...ULSUL	Fehler bei ULS-Behandlung	PM + Unterlagen
..DIAL	Fehler beim Übertragen von geketteten Vorgängen	PM + Unterlagen
...ASYAS	Fehler bei Behandlung von FPUTs oder DPUTs an LTERM	PM + Unterlagen
...IMSIM	Fehler bei Behandlung von FPUTs oder DPUTs an TAC	PM + Unterlagen
...UPIUP	Fehler bei Behandlung einer Dialognachricht für UPIC	PM schreiben und Unterlagen erstellen
...SOCSO	Fehler bei Behandlung einer Dialognachricht für SOCKET	PM schreiben und Unterlagen erstellen
...UMSUM	Fehler bei Behandlung einer Nachricht einer User-Queue	PM schreiben und Unterlagen erstellen
...QMSQM	Fehler bei Behandlung einer Nachricht einer Queue	PM schreiben und Unterlagen erstellen
...LSB	Fehler bei LSSB-Behandlung	PM + Unterlagen
xxxx99	Interner Fehler in KDCUPD	PM + Unterlagen

Weitere Fehlercodes &TRMA können die Term-Application-Codes von openUTM sein. In diesem Fall immer eine PM schreiben und Unterlagen erstellen.

K274 Programm Ende mit Dump.

K277 Mindestens eine Knoten-Anwendung wurde nicht ordnungsgemaess beendet.
Näheres siehe stderr-Protokoll.

K278 Mindestens eine Knoten-Anwendung wurde schon einmal gestartet.

K279 Fehler bei Cluster-Bearbeitung, siehe (moegliche) vorherige Diagnosemeldung

- K300** &UPDPRO Prozent des &PGPOOL belegt.
- K303** &UKCOP-Daten uebertragen. KCRN = &UKCRN, Typ = &UPDTYP, KCLM = &UKCLM.
- K304** Vorgangs-Daten uebertragen fuer USER &USER. Vorgangs-Typ = &TACTYPE, Stapelhoehe = &UKCHSTA
- K305** Belegte Seiten des &PGPOOL fuer &UPDTYP: &PGS1 insgesamt, &PGS2 fuer &UKCRN.
- K306** Belegte Seiten des &PGPOOL fuer &UPDTYP: &PGS1.
- K310** * &UPDTYP &UKCRN nicht gefunden.
- K311** * &UPDTYP &UKCRN nicht gefunden. Daten nicht uebertragen.
- K314** * Warnung! LPUT-Daten gehen verloren.
- K317** * &UKCOP-Daten nicht uebertragen. KCRN= &UKCRN, Typ = &UPDTYP, KCLM= &UKCLM, KCRCCC= &RCCC, KRCDC= &RCDC . Sender ist LTERM = <RM, USER = &USER.
- K318** * &UPDTYP &UKCRN : Sender der Asynchron-Nachricht nicht gefunden. LTERM = <RM, USER = &USER.
- K320** * Vorgangs-Daten fuer USER &USER nicht uebertragen. Vorgangs-Typ = &TACTYPE, Grund: &UERRCODE, &UERRINFO, &RCDC

&UERRCODE zeigt den Grund an, aus dem die Vorgangsdaten für den USER &USER nicht übertragen wurden. &UERRINFO enthält Zusatzinformation.

&UERRCODE	&UERRINFO	Bedeutung
KBLTH	<bytes>	Der Generierungs Parameter MAX KB in der neuen KDCFILE ist zu klein. Der Vorgang benötigt „bytes“ Bytes.
LSSBS	<anzahl>	Anzahl <anzahl> LSSB' s in Vorgang des Benutzers ist größer als der Generierungswert (MAX LSSBS=).
LTERM	<lterm>	Das Sender-LTERM ist nicht definiert.
MISSMATCH		Der Benutzer ist in der neuen KDCFILE mit RESTART=NO generiert oder sein Typ hat sich geändert (Verbindungs-Benutzerkennung ja/nein).
NRCONV	<anzahl>	Anzahl <anzahl> der gekellerten Vorgänge des Benutzers ist größer als der Generierungswert (MAX NRCONV =).
PAGEPOOL		Der Pagepool ist zu klein (MAX PAGEPOOL =).

&UERRCODE	&UERRINFO	Bedeutung
RESSHORT		UTM-Tabelleneinträge zur Verwaltung von Vorgangsdaten reichen nicht aus. In der neuen KDCFILE wurden mehr Benutzerkennungen, Sessions und/oder Tasks generiert als in der alten oder der Wert von ASYNTASKS ist kleiner als in der alten KDCFILE.
TAC	<name>	Transaktionscode <name> ist ungültig.
TRMSGLTH		Nachricht für ein Folgeprogramm ist zu groß (MAX TRMSGLTH =).

5.2 Meldungen des XAP-TP-Providers

Die Meldungen des XAP-TP-Providers beginnen mit dem Buchstaben 'P'. Die Werte der Inserts sind entweder im Anschluss an die Meldung erläutert oder - für mehrfach vorkommende Inserts - in [Abschnitt „Allgemeine Inserts der XAP-TP-Meldungen“ auf Seite 333](#) aufgeführt.

P001 Fehler beim OSS Aufruf (&XPFUNC): &ACPNT, &XPRET, &XPERR, &XP1INFO, &XP2INFO

Diese Meldung wird ausgegeben, wenn der Aufruf einer OSS-Funktion (&XPFUNC) einen Fehler liefert. Handelt es sich um einen vom Transportsystem gemeldeten Fehler, wird zusätzlich die Meldung P012 ausgegeben.

Die Inserts haben folgende Bedeutung:

&XPFUNC	Name der OSS-Funktion
&ACPNT	Name des lokalen ACCESS-POINT
&XPRET	siehe Tabelle auf Seite 333
&XPERR	siehe Tabelle auf Seite 333
&XP1INFO	OSS-Zusatzinformation
&XP2INFO	OSS-Zusatzinformation

P002 Fehler beim Associationaufbau (&XPFUNC): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

Diese Meldung wird ausgegeben, wenn der Aufruf einer OSS-Funktion (&XPFUNC), die zum Aufbau einer Association benötigt wird, einen Fehler liefert. Handelt es sich um einen vom Transportsystem gemeldeten Fehler, wird zusätzlich die Meldung P012 ausgegeben. Ist es kein vom Transportsystem gemeldeter Fehler, wird die Anwendung mit „Termapplication“ beendet.

Die Inserts haben folgende Bedeutung:

&XPFUNC	Name der OSS-Funktion
&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPRET	siehe Tabelle auf Seite 333
&XPERR	siehe Tabelle auf Seite 333
&XP1INFO	OSS-Zusatzinformation
&XP2INFO	OSS-Zusatzinformation

P003 Association abgelehnt (a_assin()): &ACPNT, Grund: &XPRJCT, Laenge: &XPLTH

Diese Meldung wird ausgegeben, wenn der Aufbau einer Association von außen abgelehnt wird.

Die Inserts haben folgende Bedeutung:

&ACPNT Name des lokalen ACCESS-POINT

&XPLTH fehlerhafte Länge

&XPRJCT siehe Tabelle auf [Seite 336](#)

P004 Association abgelehnt (a_assin()): &ACPNT, &OSLPAP, Grund: &XPRJCT

Diese Meldung wird ausgegeben, wenn der Aufbau einer Association von außen abgelehnt wird.

Die Inserts haben folgende Bedeutung:

&ACPNT Name des lokalen ACCESS-POINT

&OSLPAP Name des Partners in der lokalen Anwendung

&XPRJCT siehe Tabelle auf [Seite 336](#)

Mögliche Ursachen bei &XPRJCT = 34 oder &XPRJCT = 35
(NO_MORE_CONTENTION_LOSER / WINNER_ASSOCIATIONS, [Seite 336](#)):

- Associations wurden abgebaut, ohne dass die lokale UTM-Anwendung darüber informiert wurde. Die Associations sind aus Sicht der lokalen UTM-Anwendung noch aufgebaut. Die OSI TP-Partner-Anwendung versucht diese Associations wieder aufzubauen. Diese können aber erst wieder aufgebaut werden, wenn entweder die betroffenen Associations durch Ablauf des Idle-Timers oder alle Associations zum betroffenen Partner administrativ abgebaut werden.
- In der OSI TP-Partner-Anwendung sind mehr Associations generiert als in der lokalen UTM-Anwendung.
- Die Anzahl der Contention Loser / Winner in der lokalen und in der Partner-Anwendung passt nicht zusammen.

P005 Association abgelehnt (a_assin()): &ACPNT, Grund: Partner unbekannt
 N-SEL: &XPNSEL, T-SEL: &XPTSEL
 S-SEL: (&XPLSSEL,&XPCSSEL,&XPHSSEL)
 P-SEL: (&XPLPSEL,&XPCPSEL,&XPHPSEL)

Diese Meldung wird ausgegeben, wenn der Aufbau einer Association von außen abgelehnt wird, weil der entfernte Partner in der lokalen Anwendung nicht bekannt ist.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&XPNSEL	Network Selektor des entfernten Partners
&XPTSEL	Transport Selektor des entfernten Partners
&XPLSSEL	Länge des Session Selectors des entfernten Partners
&XPCSSEL	Session Selector (abdruckbar) des entfernten Partners
&XPHSSEL	Session Selector (hexadezimal) des entfernten Partners
&XPLPSEL	Länge des Presentation Selectors des entfernten Partners
&XPCPSEL	Presentation Selector (abdruckbar) des entfernten Partners
&XPXPSEL	Presentation Selector (hexadezimal) des entfernten Partners

P006 Association abgelehnt (a_assin()): &ACPNT, &OSLPAP, Grund: Falscher Application Context Name (&XP0OBID, &XP1OBID, &XP2OBID, &XP3OBID, &XP4OBID, &XP5OBID, &XP6OBID, &XP7OBID, &XP8OBID, &XP9OBID)

Diese Meldung wird ausgegeben, wenn der Aufbau einer Association von außen abgelehnt wird. Der Application Context Name für den entfernten Partner stimmt nicht mit dem in der lokalen Anwendung für diesen Partner generierten Application Context Namen überein.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XP0OBID - &XP9OBID	sind die (maximal) zehn Elemente des Objektbezeichners, die den Application Context Namen des entfernten Partners bilden. Für nicht belegte Elemente wird -1 ausgegeben.

P007 Fehler beim Associationaufbau (`a_assrs()`): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

Diese Meldung wird ausgegeben, wenn der Aufruf der OSS-Funktion `a_assrs()`, mit der der Associationaufbau von außen beantwortet wird, einen Fehler liefert. Handelt es sich um einen vom Transportsystem gemeldeten Fehler, wird zusätzlich die Meldung P012 ausgegeben.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPRET	siehe Tabelle auf Seite 333
&XPERR	siehe Tabelle auf Seite 333
&XP1INFO	OSS-Zusatzinformation
&XP2INFO	OSS-Zusatzinformation

P008 Association aufgebaut: &ACPNT, &OSLPAP

Diese Meldung wird ausgegeben, wenn eine Association aufgebaut wurde.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung

P009 Association abgelehnt (`a_asscf()`): &ACPNT, &OSLPAP, Grund: &XPRJCT, Laenge: &XPLTH

Diese Meldung wird ausgegeben, wenn der aktive Aufbau einer Association abgelehnt wird, weil die Bestätigung des Partners nicht akzeptiert werden kann.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPRJCT	siehe Tabelle auf Seite 336
&XPLTH	eventuell eine fehlerhafte Länge

P010 Association abgelehnt (a_asscf()): &ACPNT, &OSLPAP, Grund: Partner unbekannt
 N-SEL: &XPNSEL, T-SEL: &XPTSEL
 S-SEL: (&XPLSSEL,&XPCSSEL,&XPHSSEL)
 P-SEL: (&XPLPSEL,&XPCPSEL,&XPHPSEL)

Diese Meldung wird ausgegeben, wenn der aktive Aufbau einer Association abgelehnt wird, weil der entfernte Partner bei der Bestätigung des Associationsaufbaus mit einer Adresse (&XPADDR) antwortet, die in der lokalen Anwendung nicht bekannt ist.

Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPNSEL	Network Selektor des entfernten Partners
&XPTSEL	Transport Selektor des entfernten Partners
&XPLSSEL	Länge des Session Selectors des entfernten Partners
&XPCSSEL	Session Selector (abdruckbar) des entfernten Partners
&XPHSSEL	Session Selector (hexadezimal) des entfernten Partners
&XPLPSEL	Länge des Presentation Selectors des entfernten Partners
&XPCPSEL	Presentation Selector (abdruckbar) des entfernten Partners
&XPXPSEL	Presentation Selector (hexadezimal) des entfernten Partners

P011 Association abgelehnt (a_asscf()): &ACPNT, &OSLPAP, Grund: Falscher Application Context Name (&XP0OBID, &XP1OBID, &XP2OBID, &XP3OBID, &XP4OBID, &XP5OBID, &XP6OBID, &XP7OBID, &XP8OBID, &XP9OBID)

Diese Meldung wird ausgegeben, wenn der aktive Aufbau einer Association abgelehnt wird, weil der entfernte Partner bei der Bestätigung des Associationsaufbaus mit einem Application Context Namen antwortet, der nicht mit dem in der lokalen Anwendung für diesen Partner konfigurierten Application Context Namen übereinstimmt. Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XP0OBID - &XP9OBID	sind die (maximal) zehn Elemente des Objektbezeichners, die den Application Context Namen des entfernten Partners bilden. Für nicht belegte Elemente wird -1 ausgegeben.

P012 CMX Diagnoseinformation: &XPCTYPE, &XPCCLS, &XPCVAL

Diese Meldung wird ausgegeben, wenn eine vorhergehende Meldung auf Grund eines vom Transportsystem gemeldeten Fehlers ausgegeben wurde. Es wird der Diagnosecode des Transportsystems aufbereitet. In den folgenden Tabellen werden einige Werte für &XPCTYPE, &XPCCLS und &XPCVAL beschrieben. Eine vollständige Auflistung enthält die zu CMX gehörende Include-Datei `cmx.h`.

XPCTYPE	Bedeutung (CMX-Fehlertyp)
0	T_CMXTYPE: CMX-Fehler, der von der CMX-Bibliothek erkannt wurde
2	T_DSTEMPERR: Temporärer TNS-Fehler
3	T_DSCALL_ERR: TNS-Aufruffehler
4	T_DSPERM_ERR: Permanenter TNS-Fehler
5	T_DSWARNING: TNS-Warnung
>15	CMX-Fehler auf Grund von Fehlercodes aus dem Transportsystem

XPCCLS	Bedeutung (CMX-Fehlerklasse, gültig für &XPCTYPE < 15)
0	T_CMXCLASS: CMX-Klasse
2	T_DSNOT_SPEC: TNS-Klasse nicht spezifiziert
3	T_DSPAR_ERR: TNS-Parameterfehler
4	T_DSILL_VERS: Ungültige TNS-Version
5	T_DSSYS_ERR: TNS-Systemfehler
6	T_DSINT_ERR: Interner TNS-Fehler
7	T_DSMESSAGE: TNS-Hinweis

XPCVAL	Bedeutung (CMX-Fehlerwert)
0	T_NOERROR: Kein Fehler
5	T_EIO: Momentaner Engpass bzw. Fehler im Transportsystem
14	T_EFAULT: IO_Area nicht allokiert
100	T_UNSPECIFIED: Nicht näher spezifizierter Fehler, i. a. Fehler bei einem Systemaufruf
101	T_WSEQUENCE: Unzulässige Aufrufreihenfolge
103	T_WPARAMETER: Ungültiger Parameter
104	T_WAPPLICATION: Die Task ist zur Anmeldung der Anwendung nicht berechtigt oder die Anwendung ist von dieser Task bereits eröffnet.
105	T_WAPP_LIMIT: Der Grenzwert für gleichzeitig aktive Anwendungen ist erreicht.

XPCVAL	Bedeutung (CMX-Fehlerwert)
106	T_WCONN_LIMIT: Der Grenzwert für gleichzeitig aktive Verbindungen ist erreicht.
107	T_WTREF: Ungültige Transportreferenz oder Transportverbindung bereits abgebaut.
111	T_NOCCP: Das Transportsystem unterstützt die gewünschte Anwendung oder Verbindung nicht.
114	T_CCP_END: Das Transportsystem wurde beendet oder die Anwendung wurde vom Administrator geschlossen.
255	T_WLIBVERSION: Kein Anschluss an das CMX-Subsystem möglich.
-100	T_INVREF: Die evid ist ungültig. CMX kann den Aufruf keinem Wartepunkt zuordnen.

P013 Association abgelehnt (a_asscf()): &ACPNT, &OSLPAP, Grund: &XPCRES, &XPSRC, &XPNDIA
CCR V2 = &XP1BOOL, Version Incompatibility = &XP2BOOL
ContWin Assignment rejected = &XP3BOOL
Bid mandatory rejected = &XP4BOOL, No reason = &XP5BOOL

Diese Meldung wird ausgegeben, wenn der aktive Aufbau einer Association vom entfernten Partner abgelehnt wird.

Die Inserts haben folgende Bedeutung:

&ACPNT Name des lokalen ACCESS-POINT

&OSLPAP Name des Partners in der lokalen Anwendung

&XPCRES gibt an, ob die Ablehnung vorübergehend oder dauerhaft ist:
0 = permanent reject (dauerhafte Ablehnung)
1 = transient reject (vorübergehende Ablehnung)

&XPCSRC gibt an, wer den Aufbau abgelehnt hat:
0 = ACSE Service User
1 = ACSE Service Provider
2 = Presentation Service Provider

&XPNDIA siehe Tabelle auf [Seite 339](#)

&XP1BOOL - &XP5BOOL

können die Werte TRUE und FALSE annehmen. Die mit TRUE belegten Werte geben an, welche Gründe der Partner für die Ablehnung des Associationsaufbauwunsches mitgeschickt hat:

&XP1BOOL: Die CCR Version 2 ist nicht verfügbar
 &XP2BOOL: Die TP-Protokollversionen sind nicht kompatibel
 &XP3BOOL: Die Contention-Winner-Zuordnung wird abgelehnt
 &XP4BOOL: Die Festlegung „Bidding ist Pflicht“ oder „Bidding ist nicht Pflicht“ wird abgelehnt
 &XP5BOOL: Es wird kein Grund angegeben

Mögliche Ursachen dafür, dass die Contention-Winner-Zuordnung abgelehnt wird (&XP3BOOL=TRUE):

- In der OSI TP-Partner-Anwendung sind weniger Associations generiert als in der lokalen UTM-Anwendung.
- Die Anzahl der Contention Loser/Winner in der lokalen und in der Partner-Anwendung passt nicht zusammen.
- Associations wurden abgebaut, ohne dass die OSI TP-Partner-Anwendung darüber informiert wurde. Die OSI TP-Partner-Anwendung lehnt den Aufbau der Associations ab, weil diese aus Sicht der OSI TP-Partner-Anwendung noch aufgebaut sind.

In der OSI TP-Partner-Anwendung wird in dieser Situation die Meldung P004 mit &XPRJCT = 34 oder &XPRJCT = 35 erzeugt (NO_MORE_CONTENTION_LOSER / WINNER_ASSOCIATIONS, [Seite 336](#)).

P014 Fehler beim Associationabbau (&XPFUNC): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

Diese Meldung wird ausgegeben, wenn der Aufruf einer OSS-Funktion, die zum Aufbau einer Association benötigt wird, einen Fehler liefert. Handelt es sich um einen vom Transportsystem gemeldeten Fehler, wird zusätzlich die Meldung P012 ausgegeben. Ist es kein vom Transportsystem gemeldeter Fehler, wird die Anwendung mit „Termapplication“ beendet. Die Inserts haben folgende Bedeutung:

&XPFUNC	Name der OSS-Funktion
&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPRET	siehe Tabelle auf Seite 333
&XPERR	siehe Tabelle auf Seite 333
&XP1INFO	OSS-Zusatzinformation
&XP2INFO	OSS-Zusatzinformation, zur Zeit immer Null.

P015 Association abgebaut (&XPFUNC): &ACPNT, &OSLPAP, &XPLNK, &XPSRC, &XPNDIA, &XPINI, &XP1INFO, &XP2INFO

Diese Meldung wird ausgegeben, wenn eine Association abgebaut wird.

Die Inserts haben folgende Bedeutung:

&XPFUNC	Name der OSS-Funktion
&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPLNK	repräsentiert den internen Zustand der Association 0 = Association nicht verknüpft 1 = Association mit Channel verknüpft 2 = Association mit Instanz verknüpft
&XPCSRC	Verursacher des Abbaus 0 = ACSE Service User 1 = ACSE Service Provider 2 = Presentation Service Provider
&XPNDIA	siehe Tabelle auf Seite 339
&XP1INFO	OSS-Zusatzinformation
&XP2INFO	OSS-Zusatzinformation
&XPINI	siehe folgende Tabelle

XPINI	Bedeutung
0	Die Association wurde von innen abgebaut.
401	O_LOC_TRAN Der Verursacher ist das lokale Transportsystem. &XP1INFO enthält den CMX-Returncode. Dieser wird in der Folgemeldung P012 detailliert ausgegeben.
402	O_REM_TRAN Der Verursacher ist das ferne Transportsystem. &XP1INFO enthält den Grund des CMX-Events t_disin. Die Werte sind in <code>cmx.h</code> definiert. Nachfolgend einige ausgewählte Werte von &XP1INFO und ihre Bedeutung. Dabei werden folgende Abkürzungen verwendet: <ul style="list-style-type: none"> – CCP (Communication Control Program) für das Anwendungsprogramm, das die Kommunikation steuert – TSAP (Transport Service Access Point) für den Zugriffspunkt zum Transport-Service, – NSAP (Network Service Access Point) für den Zugriffspunkt zum Netzwerk-Service . – PDU (Protocol Data Unit) für Datenelemente einer Protokoll-Schicht. &XP1INFO kann z. B. folgende Werte annehmen:

XPINI	Bedeutung
402 (Forts.)	<p>0 (T_USER) Der Abbau erfolgte durch den Kommunikationspartner, u. U. auch durch einen Benutzerfehler auf der Partnerseite.</p> <p>1 (T_TIMEOUT) Wegen Inaktivität der Verbindung gemäß Parameter t_timeout wurde die Verbindung lokal durch CMX abgebaut.</p> <p>2 (T_RADMIN) Wegen Außerbetriebnahme des CCP durch die Administration wurde die Verbindung lokal durch CMX abgebaut.</p> <p>3 (T_CCPEND) Wegen CCP-Ausfall wurde die Verbindung lokal durch CMX abgebaut.</p> <p>256 (T_RUNKOWN) Der Partner oder das CCP hat die Verbindung abgebaut. Ein Grund für den Abbau wurde nicht angegeben.</p> <p>257 (T_RSAP_CONGEST) Wegen eines TSAP-spezifischen Engpasses hat das Partner-CCP die Verbindung abgebaut.</p> <p>258 (T_RSAP_NOTATT) Das Partner-CCP hat die Verbindung abgebaut, weil der adressierte TSAP dort nicht angemeldet ist.</p> <p>259 (T_RUNSAP) Das Partner-CCP hat die Verbindung abgebaut, weil der adressierte TSAP dort nicht bekannt ist.</p> <p>261 (T_RPERMLOST) Abbau durch die Netzadministration oder Administration des Partner-CCP</p> <p>262 (T_RSYSERR) Fehler im Netz</p> <p>385 (T_RCONGEST) Das Partner-CCP hat die Verbindung wegen Betriebsmittelengpass abgebaut.</p> <p>386 (T_RCONNFAIL) Das Partner-CCP hat die Verbindung wegen Misslingens des Verbindungsaufbaus abgebaut.</p> <p>387 (T_RDUPREF) Weil für ein NSAP-Paar eine zweite Verbindungsreferenz vergeben wurde (Systemfehler), wurde die Verbindung vom Partner-CCP abgebaut.</p> <p>388 (T_RMISREF) Das Partner-CCP hat die Verbindung wegen einer nicht zuzuordnenden Verbindungsreferenz (Systemfehler) abgebaut.</p>

XPINI	Bedeutung
402 (Forts.)	<p>389 (T_PROTERR) Das Partner-CCP hat die Verbindung wegen eines Protokollfehlers (Systemfehler) abgebaut.</p> <p>391 (T_PREFLOW) Das Partner-CCP hat die Verbindung wegen Verbindungsreferenz-Überlauf abgebaut.</p> <p>392 (T_RNOCONN) Das Partner-CCP hat den Aufbau der Netzverbindung abgelehnt.</p> <p>394 (T_RINLNG) Das Partner-CCP hat die Verbindung wegen falscher Header- oder Parameterlänge (Systemfehler) abgebaut.</p> <p>448 (T_RLCONGEST) Das lokale CCP hat die Verbindung wegen Betriebsmittelengpass abgebaut.</p> <p>449 (T_RLNOQOS) Das lokale CCP hat die Verbindung abgebaut, weil „Quality of Service“ nicht mehr unterstützt wird.</p> <p>451 (T_RILLPWD) Ungültiges Verbindungspasswort.</p> <p>452 (RNETACC) Netzzugang wurde verweigert.</p> <p>464 (T_RLPROTERR) Das lokale CCP hat die Verbindung wegen eines Transportprotokollfehlers (Systemfehler) abgebaut.</p> <p>465 (T_RLINTIDU) Das lokale CCP hat die Verbindung abgebaut, weil es eine zu lange Schnittstellen-Dateneinheit erhalten hat (Systemfehler).</p> <p>466 (T_RLNORMFLOW) Das lokale CCP hat die Verbindung wegen Verletzung der Flusskontrollregeln für Normaldaten (Systemfehler) abgebaut.</p> <p>467 (T_RLEXFLOW) Das lokale CCP hat die Verbindung wegen Verletzung der Flusskontrollregeln für Vorrangdaten (Systemfehler) abgebaut.</p> <p>468 (T_RLINSAPID) Das lokale CCP hat die Verbindung abgebaut, weil es eine ungültige TSAP-Identifikation erhalten hat (Systemfehler).</p> <p>469 (T_RLINCEPID) Das lokale CCP hat die Verbindung abgebaut, weil es eine ungültige TCEP-Identifikation (Transport Connection End Point, Endpunkt einer Transportverbindung) erhalten hat (Systemfehler).</p>

XPINI	Bedeutung
402 (Forts.)	<p>470 (T_RLINPAR) Das lokale CCP hat die Verbindung wegen eines unzulässigen Parameterwerts (z. B. Benutzerdaten zu lang oder Vorrangdaten nicht zugelassen) abgebaut.</p> <p>480 (T_RLNOPERM) Die Administration des lokalen CCP hat den Verbindungsaufbau verhindert.</p> <p>481 (T_RLPERMLOST) Die Administration des lokalen CCP hat die Verbindung abgebaut.</p> <p>482 (T_RLNOCNN) Weil keine Netzverbindung verfügbar ist, konnte das lokale CCP den Verbindungsaufbau nicht durchführen.</p> <p>483 (T_RLCONNLOST) Das lokale CCP hat die Verbindung wegen Verlust der Netzverbindung abgebaut. Häufigste Ursache: Generierungsfehler auf CCP- und PDN-Seite, z. B. unstimmmige Link-Adressen. Als Problemursache kommen außerdem in Frage: Partner ist nicht vorhanden, Modem ist defekt oder falsch eingestellt, DÜ-Anschluss ist nicht gesteckt, DFÜ-Board defekt.</p> <p>484 (T_RLNORESP) Das lokale CCP kann die Verbindung nicht aufbauen, weil der Partner nicht auf die Verbindungsanforderung (CONRQ) antwortet.</p> <p>485 (T_RLIDLETRAF) Das lokale CCP hat die Verbindung wegen Verbindungsverlust (Idle Traffic Timeout) abgebaut.</p> <p>486 (T_RLRESYNC) Das lokale CCP hat die Verbindung abgebaut, weil die Resynchronisierung (mehr als zehn Wiederholungen) erfolglos war.</p> <p>487 (T_RLEXLOST) Das lokale CCP hat die Verbindung abgebaut, weil der Vorrangdatenkanal defekt ist (mehr als drei Wiederholungen).</p>
403	<p>O_LOC_SESS Der Verursacher ist der lokale Session Provider. &XP1INFO kann folgende Werte annehmen:</p> <p>4 (S_PROTERROR) Protokollfehler: fehlerhafter Aufbau der Session PDU oder fehlerhafter SPDU Parameter</p> <p>16 (S_PICSREST) Verstoß gegen implementierungsbedingte Beschränkungen.</p>

XPINI	Bedeutung
404	<p>O_REM_SESS Der Verursacher ist der entfernte Session Provider. &XP1INFO kann folgende Werte annehmen:</p> <p>1 (S_TCDISCON) Abbau der Transportverbindung (transport disconnect)</p> <p>4 (S_PROTERROR) Protokollfehler</p> <p>8 (S_UNDEFINED) Undefiniert</p> <p>16 (S_PICSREST) Verstoß gegen implementierungsbedingte Beschränkungen</p>
405	<p>O_LOC_PRES Der Verursacher ist der lokale Presentation Provider. &XP1INFO kann folgende Werte annehmen:</p> <p>0 (P_ARRNO) reason not specified</p> <ul style="list-style-type: none"> – Ein intern angeforderter Dekodierpuffer kann wegen Speichermangel nicht bereitgestellt werden. – Überlauf des internen Datenpuffers beim Reassemblieren von fragmentierten Nachrichten. – Ein unbekanntes Session-Ereignis wurde angezeigt. – Systemengpass oder Systemfehler. <p>1 (P_ARNRPDU) unrecognized PPDU</p> <ul style="list-style-type: none"> – Es sind keine Session-Userdaten vorhanden oder es kann deren Presentation-Teil nicht dekodiert werden (Systemfehler). <p>4 (P_ARNRPAR) unrecognized PPDU parameter</p> <ul style="list-style-type: none"> – Fehler beim Dekodieren der ACSE-, Presentation- oder einer Benutzer-syntax. <p>5 (P_ARNEPAR) unexpected PPDU parameter</p> <ul style="list-style-type: none"> – PPDU-Parameter nicht im normal mode. <p>6 (P_ARNIPAR) invalid PPDU parameter</p> <ul style="list-style-type: none"> – Ungültiger context identifier beim Dekodieren. – Ungültiger PPDU-Parameter, z. B. falsche Länge. Dieser „abort“ kann vom UTM-Benutzer durch die Angabe von ungültigen Presentation-/Session-Selektoren ausgelöst werden.

XPINI	Bedeutung
406	<p>O_REM_PRES Der Verursacher ist der entfernte Presentation Provider. &XP1INFO kann folgende Werte annehmen:</p> <p>-1 (O_NOVALUE) optional parameter is not present</p> <p>0 (P_ARNNO) reason not specified</p> <p>1 (P_ARNRPDU) unrecognized PPDU</p> <p>2 (P_ARNEPDU) unexpected PPDU</p> <p>3 (P_ARNESSP) unexpected session service primitive</p> <p>4 (P_ARNRPAR) unrecognized PPDU parameter</p> <p>5 (P_ARNEPAR) unexpected PPDU parameter</p> <p>6 (P_ARNIPAR) invalid PPDU parameter value</p>
407	<p>O_LOC_ACSE Der Verursacher ist der lokale ACSE Provider &XP1INFO hat immer den folgenden Wert:</p> <p>1 (A_ABSASP) ACSE-service-provider initiated the abort Es wird die Instanz angegeben, die aus ACSE-Sicht den „abort“ ausgelöst hat ("abort source").</p>
408	<p>O_REM_ACSE Der Verursacher ist der ferne ACSE-Service-Provider. &XP1INFO kann folgende Werte annehmen:</p> <p>0 (A_ABSASU) ACSE-service-user initiated the abort</p> <p>1 (A_ABSASP) ACSE-service-provider initiated the abort</p>

P016 Association abgebaut (a_relin()): &ACPNT, &OSLPAP, &XPLNK, &XPNDIA

Diese Meldung wird ausgegeben, wenn eine Association abgebaut wird, weil eine „release indication“ empfangen wurde. Die Inserts haben folgende Bedeutung:

&ACPNT	Name des lokalen ACCESS-POINT
&OSLPAP	Name des Partners in der lokalen Anwendung
&XPLNK	repräsentiert den internen Zustand der Association 0 = Association nicht verknüpft 1 = Association mit Channel verknüpft 2 = Association mit Instanz verknüpft
&XPNDIA	siehe Tabelle auf Seite 339

P017 OSS Dekodierfehler: &XPPDU, &XP1DIA, &XP2DIA, &XP3DIA

Diese Meldung wird ausgegeben, falls OSS beim Dekodieren einer TP-, CCR- oder Benutzerdaten-PDU einen Fehler erkennt. Das Insert &XPPDU kennzeichnet den Typ der betroffenen PDU.

Die Inserts haben folgende Bedeutung:

XPPDU	Bedeutung
0	Unbekannter PDU-Typ (PDU_UNKNOWN)
1	TP_BEGIN_DIALOGUE_RI
2	TP_BEGIN_DIALOGUE_RC
3	TP_BID_RI
4	TP_BID_RC
5	TP_END_DIALOGUE_RI
6	TP_END_DIALOGUE_RC
7	TP_U_ERROR_RI
8	TP_U_ERROR_RC
9	TP_ABORT_RI
10	TP_GRANT_CONTROL_RI
11	TP_REQUEST_CONTROL_RI
12	TP_HANDSHAKE_RI
13	TP_HANDSHAKE_RC
14	TP_HSK_AND_GRT_CTRL_RI
15	TP_HSK_AND_GRT_CTRL_RC
16	TP_DEFER_RI
17	TP_PREPARE_RI

XPPDU	Bedeutung
18	TP_HEURISTIC_REPORT_RI
19	TP_TOKEN_GIVE_RI
20	TP_TOKEN_PLEASE_RI
21	TP_RECOVER_RI
22	TP_INITIALIZE_RI
23	TP_INITIALIZE_RC
24	CCR_INITIALIZE_RI
25	CCR_INITIALIZE_RC
26	CCR_BEGIN_RI
27	CCR_BEGIN_RC
28	CCR_PREPARE_RI
29	CCR_READY_RI
30	CCR_COMMIT_RI
31	CCR_COMMIT_RC
32	CCR_ROLLBACK_RI
33	CCR_ROLLBACK_RC
34	CCR_RECOVER_RI
35	CCR_RECOVER_RC
50	PDU_ANY
51	PDU_UASE_RI

XP1DIA / XP2DIA	Bedeutung
1	not supported parameter was received and skipped
2	received data truncated
4	required transfer syntax name missing in user data or not specified in AVX list, error codes in &XP2DIA
6	no transfer syntax name in user data though presentation negotiation was not completed
7	transfer syntax name encoded in user data not found in AVX list
10	invalid value in data structure
11	invalid object identifier in data structure
12	invalid length or count in data structure
13	invalid index in data structure (EXTERNAL, CHOICE)
14	invalid value of ax_typtag in corresponding syntax table

&XP3DIA zugehöriger Index in der Syntaxtabelle

P018 FSM Protokollfehler: &ACPNT, &OSLPAP, &XPPTYP, &XPFSMN

Diese Meldung wird ausgegeben, wenn die Finite-State-Machine einen Fehler meldet.

Die Inserts haben folgende Bedeutung:

&ACPNT Name des lokalen ACCESS-POINT

&OSLPAP Name des Partners in der lokalen Anwendung

&XPPTYP Typ des Service-Protokollelements

&XPFSMN Name der Finite-State-Machine

P019 APDU enthaelt ungueltigen Wert: &ACPNT, &OSLPAP, &XPAPDU, &XP3INFO

Diese Meldung wird ausgegeben, wenn eine ungültige APDU empfangen wurde.

Die Inserts haben folgende Bedeutung:

&ACPNT Name des lokalen ACCESS-POINT

&OSLPAP Name des Partners in der lokalen Anwendung

&XPAPDU Typ der APDU

&XP3INFO Zusatzinformation zum Fehler

P020 OTRACE implizit ausgeschaltet. Grund: &XPTRFAIL

Diese Meldung wird ausgegeben, wenn der Versuch, einen Trace-Record zu schreiben, fehlgeschlagen ist. Als Folge des Fehlers wurde der OSS-Trace implizit ausgeschaltet. Nachdem der Fehler behoben worden ist, kann über die Administration der OSS-Trace wieder eingeschaltet werden.

Die Inserts haben folgende Bedeutung:

XPTRFAIL	Bedeutung
1	Die OSS-Funktion o_wutr() lieferte den Returncode O_ERROR. In der vorangegangenen P001-Meldung finden Sie nähere Informationen zu dem Fehler.
2	Die OSS-Funktion o_wutr() lieferte den Returncode O_INVEREF.
3	Die OSS-Funktion o_wutr() lieferte einen unbekanntem Returncode.

P021 Unerwartetes Event &XPEVT fuer Association aufgetreten, Event wird ignoriert:
&ACPNT, &OSLPAP, &XPOSAS, &XPASST

Diese Meldung wird ausgegeben, wenn ein Ereignis auftritt, das nicht zum momentanen Zustand der Association passt. XAPTP berücksichtigt dieses Ereignis nicht.

Die Inserts der Meldung haben folgende Bedeutung:

&XPEVT	Typ des aufgetretenen Ereignisses.
&ACPNT	Name des lokalen Access Point (KDCDEF-Anweisung ACCESS-POINT).
&OSLPAP	Name des OSI-LPAP-Partners in der lokalen Anwendung.
&XPOSAS	Index der betroffenen Association.
&XPASST	Zustand der betroffenen Association.

5.2.1 Allgemeine Inserts der XAP-TP-Meldungen

XPRET	Bedeutung
2	Not first process of application
-1	Function call not successful due to permanent error.
-2	Function call not successful due to transient error. Retry the call later
-3	Function call not successful, data flow stopped Continue after event GO
-4	Session call: Expedited function call stopped due to expedited data flow control shortage Continue after event S_XGO/S_GO Presentation call: Function call not successful, apref invalid Local function call: Invalid connection reference ACSE call: Function call not successful, apref resp. aref invalid
-5	Invalid waiting point reference
-6	Invalid application reference
-7	Waiting period to obtain a lock on a shared association expired

XPERR	Bedeutung
1	No memory available (temporary)
100	Call sequence error
101	Application not attached
102	Sending of data not allowed; wait for GO event
103	Internal error
104	Shared association is not locked
200	Missing ACSE/presentation reference
201	Invalid ACSE/presentation reference
202	Presentation call: missing AVX list (o_attach) ACSE call: missing application reference
203	Presentation call: invalid AVX list ACSE call: invalid application reference

XPERR	Bedeutung
204	Presentation call: invalid abstract syntax name in AVX ACSE call: missing ACSE parameters
205	Presentation call: invalid decoding mode in AVX ACSE call: missing presentation parameters
206	Presentation call: invalid user data length ACSE call: missing session parameters
207	Presentation call: invalid context id in p_udl ACSE call: missing application context name
208	Presentation call: invalid next parameter in p_udl ACSE call: invalid application context name
209	Presentation call: invalid pdv parameter in p_udl ACSE call: invalid calling AP Title
210	Presentation call: invalid chaining parameter ACSE call: invalid calling AE Qualifier
211	Presentation call: missing token parameter ACSE call: invalid called AP Title
212	Presentation call: invalid token parameter ACSE call: invalid called AE Qualifier
213	Presentation call: missing rtype parameter ACSE call: invalid responding AP Title
214	Presentation call: invalid rtype parameter ACSE call: invalid responding AE Qualifier
215	Presentation call: missing type parameter ACSE call: missing called p_address
216	Presentation call: invalid type parameter ACSE call: invalid called p_address
217	Presentation call: invalid syncp parameter ACSE call: missing calling p_address
218	Presentation call: missing syncp parameter ACSE call: missing responding p_address
219	Presentation call: invalid ctxlst parameter ACSE call: no mode parameter
220	Presentation call: invalid number of abstract syntaxes passed to OSS ACSE call: invalid mode parameter
221	Presentation call: invalid transfer syntax name ACSE call: missing result
222	Presentation call: invalid number of transfer syntaxes ACSE call: invalid result

XPERR	Bedeutung
223	Presentation call: invalid number of abstract syntaxes ACSE call: missing result source
224	Presentation call: same abstract syntax occurred already in transparent or non-transparent mode ACSE call: invalid result source
225	Presentation call: invalid data separation parameter ACSE call: invalid diagnostic
226	ACSE call: missing reason
227	ACSE call: invalid reason
228	ACSE call: missing provider reason
229	ACSE call: invalid provider reason
230	ACSE call: missing abort source
231	ACSE call: invalid p-requirements
232	ACSE call: invalid s-requirements
233	ACSE call: invalid syntax identifier
234	ACSE call: invalid p-context identifier
235	ACSE call: invalid p-context definition list
236	ACSE call: invalid p-context definition result list
237	ACSE call: invalid result in p-context definition result list
238	ACSE call: invalid default p-context result
239	ACSE call: invalid default p-context name
240	ACSE call: invalid user data length
241	ACSE call: invalid quality of service
242	ACSE call: invalid sync point serial number
243	ACSE call: invalid tokens
244	ACSE call: invalid SS-user reference
245	ACSE call: invalid SS-common reference
246	ACSE call: invalid SS-additional reference
250	Presentation call: ASN encoding error ACSE call: ASN encoding error
251	Presentation call: ASN decoding error ACSE call: ASN decoding error
252	Presentation call: ASN: invalid value in data struct ACSE call: ASN: invalid value in data struct

XPERR	Bedeutung
253	Presentation call: ASN: invalid object id in data struct ACSE call: ASN: invalid object id in data struct
254	Presentation call: ASN: invalid length in data struct ACSE call: ASN: invalid length in data struct
255	Presentation call: ASN: invalid index in data struct ACSE call: ASN: invalid index in data struct
256	Presentation call: ASN: invalid tag in syntax table ACSE call: ASN: invalid tag in syntax table
300	Presentation call: invalid protocol state ACSE call: invalid protocol state Local function call: error on system call
301	Presentation call: protocol error ACSE call: protocol error Local function call: error on transport system call
302	Local function call: error on local function call
305	Local function call: error on session call
306	Local function call: error on presentation call
307	Local function call: error on ACSE call

XPRJCT	Bedeutung
0	NO_REJECT
1	APPLICATION_CONTEXT_NAME_TOO_LONG Der vom Partner empfangene Objektbezeichner, der den Application Context Name bildet, enthält mehr Elemente, als von openUTM unterstützt.
2	CALLING_APT_TOO_LONG Bei der Association Indication wird für den Application Process Title eine Länge angegeben, die openUTM nicht unterstützt.
3	CALLING_AEQ_TOO_LONG Bei der Association Indication wird für den Application Entity Qualifier eine Länge angegeben, die openUTM nicht unterstützt.
4	CALLED_APT_TOO_LONG Der gerufene Application Process Title ist länger als von openUTM unterstützt.
5	CALLED_AEQ_TOO_LONG Der gerufene Application Entity Qualifier ist länger als von openUTM unterstützt.
6	CONTEXT_DEFINITION_LIST_TOO_LONG Bei der Association Indication werden mehr abstrakte Syntaxen mitgegeben, als openUTM unterstützt.

XPRJCT	Bedeutung
7	CONTEXT_RESULT_LIST_TOO_LONG Die beim Associationsaufbau (Association indication bzw. confirmation) mitgegebene Liste der unterstützten abstrakten Syntaxen enthält mehr Elemente, als von openUTM unterstützt.
9	ADDRESS_NO_PSAP_INFO Die bei Association indication bzw. confirmation mitgegebene Adresse enthält keine Informationen zum PSAP.
10	ADDRESS_NO_INFO_VERS_0_PSAP Die bei Association indication bzw. confirmation mitgegebene Adresse enthält für die PSAP-Information eine falsche Version.
11	ADDRESS_INVALID_P_SEL_LENGTH Die bei Association indication bzw. confirmation mitgegebene Adresse enthält für den Presentation-Selektor eine ungültige Länge.
12	ADDRESS_NO_SSAPINFO Die bei Association indication bzw. confirmation mitgegebene Adresse enthält keine Information zum SSAP.
13	ADDRESS_NO_INFOVERS_0_SSAP Die bei Association indication bzw. confirmation mitgegebene Adresse enthält für die SSAP-Information eine falsche Version.
14	ADDRESS_INVALID_S_SEL_LENGTH Die bei Association indication bzw. confirmation mitgegebene Adresse enthält keinen gültigen Teil für den Session-Selektor.
15	ADDRESS_NO_PARTNER_MODE Die bei Association indication bzw. confirmation mitgegebene Adresse enthält keinen gültigen Teil für den Network- und Transport-Selektor.
16	ADDRESS_TNSX_ERROR Die bei Association indication bzw. confirmation mitgegebene Adresse wird vom TNS abgewiesen.
17	UNKNOWN_PARTNER Die bei Association indication bzw. confirmation mitgegebene Adresse ist in der lokalen Anwendung nicht bekannt.
18	WRONG_APPLICATION_CONTEXT_NAME Der bei Association indication bzw. confirmation mitgegebene Application Context Name entspricht nicht dem in der lokalen Anwendung generierten Application Context Namen.
19	ABSTRACT_SYNTAX_MISSING Bei Association indication bzw. confirmation werden weniger abstrakte Syntaxen unterstützt, als in der lokalen Anwendung generiert wurden.
20	OSITP_SYNTAX_MISSING Bei Association indication bzw. confirmation wird die abstrakte Syntax für OSI TP nicht unterstützt.

XPRJCT	Bedeutung
21	NO_TP_INITIALIZE Bei Association indication bzw. confirmation wird keine TP-INITIALIZE-RI/RC PDU mitgeliefert.
22	OSITP_NO_VERSION_1 Der Partner unterstützt die Version 1 des OSI TP-Protokolls nicht.
23	OSITP_RCH_WRONG_LENGTH Der bei der TP-INITIALIZE-indication bzw. TP-INITIALIZE-confirmation mitgegebene Recovery Context Handle hat eine Länge, die von openUTM nicht unterstützt wird.
24	NO_CCR_INITIALIZE Die CCR-INITIALIZE-RI PDU fehlt.
25	CCR_NOT_VERSION_2 Der Partner unterstützt die Version 2 des CCR-Protokolls nicht.
26	SESSION_NO_FDX Die Session-Funktionalität „Full Duplex“ ist nicht gesetzt.
27	SESSION_NO_DATA_SEPARATION Die Session-Funktionalität „Data Separation“ ist nicht gesetzt, obwohl CCR im Kontext ist.
28	SESSION_NO_TYPED_DATA Die Session-Funktionalität „Typed Data“ ist nicht gesetzt, obwohl CCR im Kontext ist.
29	SESSION_NO_MINOR_SYNCHRONIZE Die Session-Funktionalität „Minor Synchronize“ ist nicht gesetzt, obwohl CCR im Kontext ist.
30	SESSION_NO_RESYNCHRONIZE Die Session-Funktionalität „Resynchronize“ ist nicht gesetzt, obwohl CCR im Kontext ist.
31	TOKEN_CONTENTION_WINNER_AND_NO_TOKEN Die lokale Anwendung ist Contention Winner, aber nicht im Besitz des „Tokens“ (nur wenn CCR im Kontext ist).
32	TOKEN_CONTENTION_LOSER_AND_TOKEN Die lokale Anwendung ist Contention Loser, aber im Besitz des „Tokens“ (nur wenn CCR im Kontext ist).
33	INITIAL_SYNC_POINT_SERIAL_NUMBER_NOT_SET Die Initial Syncpoint Serial Number ist nicht gesetzt, obwohl CCR im Kontext ist.
34	NO_MORE_CONTENTION_LOSER_ASSOCIATIONS Der Associationsaufbau von außen wird abgelehnt, weil in der lokalen Anwendung alle Contention Loser Associationen bereits aufgebaut sind.
35	NO_MORE_CONTENTION_WINNER_ASSOCIATIONS Der Associationsaufbau von außen wird abgelehnt, weil in der lokalen Anwendung alle Contention Winner Associationen bereits aufgebaut sind.

XPRJCT	Bedeutung
36	CCR_BUT_NO_PARTNER_AET Der Partner hat keinen Application Entity Title angegeben, obwohl CCR im Kontext ist.
37	CCR_BUT_NO_OWN_AET In der lokalen Anwendung ist kein Application Entity Title angegeben, obwohl CCR im Kontext ist.
38	RESPONDING_APT_TOO_LONG Der in der Association confirmation angegebene Application Process Title ist länger, als von openUTM unterstützt.
39	RESPONDING_AEQ_TOO_LONG Der in der Association confirmation angegebene Application Entity Qualifier ist länger, als von openUTM unterstützt.
40	ASS_ESTABLISHMENT_TIMEOUT Der von der lokalen Anwendung begonnene Associationsaufbau kann nicht in der vorgegebenen Zeit vollendet werden.
41	PARTNER_IS_IN_QUIET_STATE Der Associationsaufbau wird abgelehnt, weil der Partner in der lokalen Anwendung auf Quiet gesetzt ist.
42	NO_SPACE_FOR_RCH Der PutElement Aufruf zum Abspeichern des Recovery Context Handle liefert einen schlechten Returnwert.
43	REMOTE_AET_2_BIG Der Application Entity Title des Partners ist länger, als von openUTM unterstützt.
44	REMOTE_AET_CHANGED Beim Aufbau von parallelen Associationen zu einem Partner wird vom Partner nicht derselbe Application Entity Title, wie für die erste aufgebaute Association zu diesem Partner, geliefert.
45	NO_SPACE_FOR_REMOTE_AET Der PutElement Aufruf zum Abspeichern des Application Entity Title des Partners liefert einen schlechten Returnwert.
46	PARTNER_HAS_STATUS_OFF Der Aufbau der Association wird abgelehnt, weil der Partner in der lokalen UTM-Anwendung gesperrt ist (es ist STATUS=OFF gesetzt).

XPNDIA	Bedeutung
0	NO_REASON_GIVEN
1	NO_COMMON_ACSE_VERSION Der Partner lehnt den Associationsaufbauwunsch ab, weil es keine gemeinsame ACSE Version gibt.

XPNDIA	Bedeutung
2	APPL_CONTEXT_NAM_NOT_SUPPORTED Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Application Context Name nicht unterstützt.
3	CALLING_AP_TITLE_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil beim Partner der Absender nicht richtig generiert ist (z.B. falscher N-SEL). oder (nur bei heterogener Kopplung): Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Calling Application Process Title nicht kennt.
4	CALLING_AE_QUALI_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Calling Application Entity Qualifier nicht kennt.
5	CALLING_AP_INVOC_ID_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Calling Application Process Invocation Identifier nicht kennt.
6	CALLING_AE_INVOC_ID_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Calling Application Entity Invocation Identifier nicht kennt.
7	CALLED_AP_TITLE_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Called Application Process Title nicht kennt.
8	CALLED_AE_QUALI_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Called Application Entity Qualifier nicht kennt.
9	CALLED_AP_INVOC_ID_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Called Application Process Invocation Identifier nicht kennt.
10	CALLED_AE_INVOC_ID_NOT_RECON Der Partner lehnt den Associationsaufbauwunsch ab, weil er den Called Application Entity Invocation Identifier nicht kennt.
11	PERMANENT_FAILURE Der Partner baut die Association ab, weil ein permanenter Fehler aufgetreten ist.
12	BEGIN_TRANSACTION_REJECT Der Partner baut die Association ab, weil er den Beginn einer Transaktion ablehnt.
13	TRANSIENT_FAILURE Der Partner baut die Association ab, weil ein vorübergehender Fehler aufgetreten ist.
14	PROTOCOL_ERROR Der Partner baut die Association ab, weil ein Protokollfehler aufgetreten ist.

XPNDIA	Bedeutung
15	UNRECOGNIZED_PDU Die Association wird von außen mit P-ABORT abgebaut, weil die Presentation Schicht eine unbekannte Presentation PDU erhalten hat.
16	UNEXPECTED_PDU Die Association wird von außen mit P-ABORT abgebaut, weil die Presentation Schicht eine unerwartete Presentation PDU erhalten hat.
17	UNEXPECTED_SESSION_SERVICE_PRIMITIVE Die Association wird von außen mit P-ABORT abgebaut, weil die Session Schicht eine unerwartete Session Service Primitive erhalten hat.
18	UNRECOGNIZED_PDU_PARAMETER Die Association wird von außen mit P-ABORT abgebaut, weil die Presentation Schicht einen unbekannten PPDU Parameter erhalten hat.
19	UNEXPECTED_PPDU_PARAMETER Die Association wird von außen mit P-ABORT abgebaut, weil die Presentation Schicht einen unerwarteten PPDU Parameter erhalten hat.
20	INVALID_PPDU_PARAMETER_VALUE Die Association wird von außen mit P-ABORT abgebaut, weil die Presentation Schicht einen ungültigen PPDU Parameterwert erhalten hat.
21	RELEASE_NORMAL Die Association wird vom Partner mit release abgebaut. Der Partner gibt als Grund release normal an.
22	RELEASE_URGENT Die Association wird vom Partner mit release abgebaut. Der Partner gibt als Grund release urgent an.
23	RELEASE_USER_DEFINED Die Association wird vom Partner mit release abgebaut. Der Partner gibt als Grund user defined an
24	IDLE_TIMEOUT_ABORT Die Association wird von der lokalen Anwendung abgebaut, weil die Association in der mit IDLETIME generierten Zeit nicht benutzt worden ist.
25	ABORT_BY_ASS_INDICATION Die Association wurde von der lokalen UTM-Anwendung abgebaut, weil ein Association-Aufbauwunsch empfangen wurde und keine passende nicht aufgebaute Association verfügbar war.

5.3 Meldungen des Generierungstools KDCDEF

- K400** KDCDEF &DEFTYP &VERS wurde gestartet.
- K401** Bitte Steueranweisungen eingeben:
- K402** Syntaxfehler: &SYNERR.
- K403** Unzulaessiger Wert des Operanden &PARAM15.
- K404** DMS-Fehler &ERROR fuer Datei &FNAM.
Die DMS-Fehlercodes sind auf [Seite 383](#) beschrieben.
- K405** Fehler &ERROR.
- K406** Unzulaessiger Wert des Operanden &PARAM15. &PARVAL8 wurde uebernommen.
- K407** User "&PARVAL8" konnte nicht in die Cluster-User-Datei eingetragen werden.
- K408** &STMTNAME-Name &UTMNAME darf nicht mit "KDC" beginnen.
- K409** LTERM-Parameter ist bei PTERM-Anweisung mit PTYPE=APPLI, SOCKET, UPIC-L oder UPIC-R Pflicht.
- K410** Warnung: Das LTERM ist einem PTERM mit PTYPE=APPLI, SOCKET, UPIC-L oder UPIC-R zugeordnet und ein ungueltiger USER oder USAGE=O ist angegeben.
- K411** Der Pflichtoperand &PARAM15 wurde in keiner &PARAM15 Anweisung angegeben.
- K412** DESTADM &UTMNAME ist weder als Asynchron-TAC noch als LTERM definiert. Es wurde kein DESTADM uebernommen.
- K413** &STMTNAME &OPERAN32 &STRING64 mehrfach definiert.
- K414** LSES-Anweisung fuer LPAP &UTMNAME fehlt.
- K415** Unerlaubte Angabe im MAX-Operanden DPUTLIMIT1 oder DPUTLIMIT2. Standardwert wurde angenommen.
- K416** &PARAM30 ist mehrfach definiert.
- K417** &PARAM10 aber Typ ist nicht "&PARAM30".
- K418** CID-Parameter ist Pflicht fuer ein PTERM das von einer Druckersteuerstation kontrolliert wird.
- K419** Ein Druckerbuendel wurde definiert aber der Typ ist nicht "OUTPUT".
- K420** Ein LTERM mit USAGE=D darf keinem PTERM mit USAGE=O zugeordnet sein.

- K421** &PARAM30 und &PARAM50 schliessen sich gegenseitig aus.
- K422** Der Pflichtoperand &PARAM10 wurde nicht angegeben.
- K423** Warnung: Anzahl der Pagepoolseiten (PGPOOL) nicht grosser als Anzahl der Seiten des LPUT-Puffers (LPUTBUF).
- K424** Ungueltiger Schluesselwert (KEY): &KEYVALUE. Zulaessig sind: $0 \leq \text{KEY} \leq \text{MAX KEYVALUE}$.
- K425** &STMTNAME-Name &PARAM32 nicht definiert.
- K426** Steueranweisung &STMTNAME fehlt.
- K427** Event-Exit &EXITNAME bereits definiert.
- K428** Ungueltiger CARD-Parameter: &CARDINFO oder POSITION = 0.
- K429** Maximale Anzahl von Datenbank-Typen ueberschritten.
- K430** Zu viele &STMTNAME-Steueranweisungen.
- K431** User "&PARVAL8" aus der Cluster-User-Datei fehlt in der KDCFILE.
- K432** Es wurde kein &STMTNAME mit &PARAM32 gefunden.
- K433** Name der Filebase zu lang. Der Generierungslauf wird abgebrochen.
- K434** Minimalgrosse fuer Wiederanlaufpuffer (Operand RECBUF) ist 1024: RECBUF = 1024 angenommen.
- K435** Der Event-Exit &PARAM8 muss einem ASSEMBLER-Programm zugeordnet werden.
- K436** MAX-Anweisung fehlt. Der Generierungslauf wird abgebrochen.
- K437** Fehler in KDCDEF; Grund: &TERMREAS.
TERMREAS bezeichnet die Ursache, die zur abnormalen Beendigung des Dienstprogrammes f"uhrt.
Zur Kl"arung des Fehlers dient auch die Beschreibung der UTM Meldung K060.
Ma"nahme: Fehlerbehebung laut Ma"nahme zu K060.
In den anderen F"allen liegt ein Fehler in KDCDEF vor. Unterlagen sicherstellen und PM schreiben.
- K438** Mehr LSES-Anweisungen als CON-Anweisungen fuer LPAP &UTMNAME angegeben.
- K439** Im Parameter KDCFILE wurde ein Dateiname mit Catalog-ID angegeben.
- K440** Operand &PARAM8 in &STMTNAME-Anweisung mit &PARAM10 nicht erlaubt.
- K441** SHARE-Tabelle &UTMNAME ist leer.

- K442** Operand &PARAM8 nur sinnvoll wenn &STMTNAME -Anweisung einem &PARVAL8-Teilprogramm zugeordnet.
- K443** ULS-Steueranweisungen nur sinnvoll, wenn die Anwendung mit USER-Anweisungen generiert wurde.
- K444** OPTION-Steueranweisung nur wirksam, wenn von stdin gelesen wird.
- K445** &PARAM30 wird nicht mehr unterstuetzt.
- K446** Der Operand wurde ignoriert.
- K447** Die Steueranweisung wurde ignoriert.
- K448** Warnungen bei der KDCFILE-Generierung; KAA-Groesse: &KAASIZE K
- K449** Es ist mindestens ein Fehler aufgetreten. Der Generierungslauf wurde abgebrochen.
- K450** KDCFILE erzeugt; KAA-Groesse: &KAASIZE K
- K451** Datei &FNAM erzeugt.
- K452** Zuviele &PARAM10-Namen. Der Generierungslauf wird abgebrochen.
- K453** Von APPLNAME abweichender BCAMAPPL-NAME ist nur bei einer &STMTNAME-Anweisung mit PTYPE=APPLI, SOCKET oder UPIC-R erlaubt.
- K454** Laenge des LTERM-Parameters &UTMNAME mit NUMBER-Parameter unvertraeglich.
- K455** Ein dem Parameter STACK zugeordneter TAC darf nicht mit TYPE=A oder CALL=NEXT generiert werden.
- K456** Einem PTERM darf als LTERM kein Gruppen-LTERM zugewiesen werden.
- K457** Warnung: &PARAM10 groesser als &PARAM10A
- K458** LTERM &PARAM8 ist LTERM &PARAM8A als CTERM zugeordnet. Die Angabe "USAGE = OUTPUT" wirkt nicht.
- K459** Bei GROUP darf kein LTERM angegeben werden das selbst ein Gruppen-LTERM ist.
- K460** Datei &FNAM wurde bearbeitet.
- K461** &STMTTP30 &STMTNM30 nicht von &UTMVARunterstuetzt.
- K462** In Verbindung mit der lokalen Partneranwendung &PARAM8 darf nur eine OSICON-Verbindungsgruppe aktiv sein.
- K463** &PARAM30 auf &PARAM30gesetzt.
- K464** Laenge von Session-Namen + Laenge von Verbindungen > 8.

- K465** &STMTNAME-Steueranweisung obligatorisch, wenn &PARAM30 generiert ist.
- K466** Parameter &PARAM30 ist zwingend, wenn &PARAM30 generiert ist.
- K467** Fehler: User "&PARAM8" ist mehrfach in der Cluster-User-Datei enthalten
- K468** Passwort-Parameter passt nicht zur Sicherheitsstufe.
- K469** Parameter &PARAM32 nur erlaubt bei APPLIMODE = SECURE.
- K470** Kein Benutzer kann KDCSHUT aufrufen.
- K471** &OPERAND&PARAM10 bereits gesetzt.
- K472** &PARAM32 zu lang.
- K473** Parameter &PARAM10 mehr als einmal fuer &PARAM30 benutzt.
- K474** Parameter &PARAM30 unzuessaessig, wenn &PARAM30 generiert ist.
- K475** Parameter &PARAM30 unnoetig, wenn &PARAM30 generiert ist.
- K476** In &STMTNAME-Anweisungen wurde &OPERAND mehr als einmal fuer PRONAM=&PARAM8 und BCAMAPPL=&PARAM8 angegeben.
- K477** Parameter &PARAM30 unzuessaessig, wenn ein &PARAM30 angegeben ist.
- K478** Anweisung &STMTNAME unzuessaessig, wenn &PARAM30 angegeben wurde.
- K479** &STMTNAME &PARAM50 nicht verwendet.
- K480** Zuviele &STMTNAME Steueranweisungen fuer diesen &PARAM30.
- K481** Das primary LTERM einer Gruppe darf nicht gleichzeitig Slave eines Verbindungs-buendels sein.
- K482** EXIT und TAC muessen im gleichen LOAD-MODULE sein, wenn die EXIT-Routine sich auf ein LOAD-MODULE mit LOAD-MODE=ONCALL bezieht.
- K483** &PARAM32 darf sich nicht auf ein &STMTNAME mit &PARAM32 beziehen.
- K484** &PARAM32 mit &PARAM32 nicht generiert.
- K485** Die angegebene Anweisung ist zu lang.
- K486** Es wurden unterschiedliche Bibliotheken fuer denselben Datenbanktyp angegeben.
- K487** Das Master LTERM eines Verbindungs-buendels darf nicht einem PTERM zugewiesen werden.
- K488** Alle Slave LTERMs eines Verbindungs-buendels muessen PTERMs mit identischem PTYPE (APPLI oder SOCKET) zugewiesen werden.
- K489** Das Master LTERM eines Verbindungs-buendels darf nicht seinerseits Slave LTERM sein.

- K490** Ein Master LTERM muss mit QAMSG=YES und RESTART=YES generiert werden. Die Parameter wurden korrigiert.
- K491** Die Angabe &OPERAND&PARAM32&PARAM20 wird in der Folgeversion nicht mehr unterstützt.
- K492** Hinweis: die Angabe von mehr als einer ACCESS-POINT Anweisung kann zu Problemen führen.
- K493** Das Erzeugen der TNS-Eingabedatei ist misslungen.
- K494** Der Operand &PARAM30 muss angegeben werden, wenn die Steueranweisung &STMTNAME mehr als einmal benutzt wird.
- K495** Der Standardmeldungsmodul wurde in keiner MESSAGE- Steueranweisung angegeben, oder er wurde durch keine MAX-Steueranweisung definiert.
- K496** Die Datei &FNAM ist zerstört oder keine KDCFILE.
- K497** KDCDEF &VER1 / &OST1 / &BMD1 kann KDCFILE &VER2 / &OST2 / &BMD2 nicht lesen.
- K498** Es sind keine &STMTNAME Steueranweisungen zu erzeugen.
- K499** &PARAM30 in &STMTNAME Steueranweisung und &PARAM30 passen nicht zusammen.
- K500** Es wurde(n) keine Datei(en) erzeugt.
- K501** Ungültiges Zeichen "&CHAR1" in der Zeichenkette: &STRING64
- K502** Anzahl freier Einträge für &PARAM10:&PARAM11
- K503** Der referenzierte ACCESS-POINT &PARAM8 muss einen APPLICATION-ENTITY-QUALIFIER enthalten, da das referenzierte OSI-LPAP &PARAM8 einen APPLICATION-CONTEXT mit der ABSTRACT-SYNTAX CCR enthält.
- K504** Das OSI-CON &PARAM8 referenziert denselben OSI-LPAP aber einen anderen ACCESS-POINT.
- K505** Der Operand &PARAM32 muss angegeben werden, wenn der referenzierte APPLICATION-CONTEXT die ABSTRACT-SYNTAX CCR enthält.
- K506** Es kann kein APPLICATION-CONTEXT angegeben werden, der die ABSTRACT-SYNTAX CCR enthält, da in der UTMD-Anweisung kein APPLICATION-PROCESS-TITLE spezifiziert wurde.
- K507** Es wurden zuviele &PARAM32 generiert.
- K508** Fehler beim Erzeugen der RSA-Keys.

- K509** Das OSI-CON &PARAM8 referenziert denselben LOCAL-ACCESS-POINT und ein OSI-LPAP mit identischen APPLICATION-ENTITY-QUALIFIER und APPLICATION-PROCESS-TITLE.
- K510** Der Operand &PARAM32 benoetigt mindestens &PARAM8 Parameter.
- K511** Es sind hoechstens &PARAM11 gleichzeitige gekellerte Vorgaenge moeglich.
- K512** Es sind hoechstens &PARAM11 gleichzeitige Anmelde- und/oder gekellerte Vorgaenge moeglich.
- K513** KDCDEF erzeugt jetzt neue RSA-Keys. Abhaengig von der Schluessellaenge und der Systemkonfiguration kann dieser Prozess einen laengeren Zeitraum in Anspruch nehmen.
Bitte warten ...
- K514** Erzeugung der RSA-Keys abgeschlossen
- K515** Warnung: Die Anwendung benoetigt zum korrekten Ablauf RSA-Keys. Es wurden jedoch keine RSA-Keys generiert.

Wenn Objekte mit Encryption-Level generiert sind, dann benoetigt die Anwendung fuer den korrekten Ablauf RSA-Schlüssel. Sind in der Anwendung keine RSA-Schlüssel verfügbar, dann kann die Anwendung nur mit Einschränkungen betrieben werden: TACs mit Encryption-Level können nicht aufgerufen werden und zu PTERMs oder TPOOLS mit Encryption-Level kann keine Verbindung aufgebaut werden.

Drei Maßnahmen sind möglich:

- Vor dem Start der Anwendung können die RSA-Schlüssel per KDCUPD aus einer alten KDCFILE in die neue KDCFILE übertragen werden,
- oder der KDCDEF-Lauf kann mit OPTION GEN-RSA-KEYS=YES wiederholt werden,
- oder die Anwendung kann gestartet werden und die benötigten RSA-Schlüssel können administrativ erzeugt und aktiviert werden (z.B. durch WinAdmin/WebAdmin).

K516 Fehler &DIAG1 bei Bearbeitung der Cluster User-Datei &STRING64

Das Insert &DIAG1 beschreibt die Fehlerursache; das Insert &STRING64 enthält den Dateinamen.

Das Insert &DIAG1 kann folgende Werte annehmen:

Wert	Bedeutung
1	OPEN_ERROR Fehler beim Öffnen der Datei.
2	CLOSE_ERROR Fehler beim Schließen der Datei.
3	FILE_ERROR Die Datei ist inkonsistent.
4	LOCK_ERROR Fehler bei Anforderung des Datei-Locks.
5	UNLOCK_ERROR Fehler bei Freigabe des Datei-Locks.
6	USER_NOT_FOUND KCCUFUE gibt beim Update eines User-Eintrags einen schlechten Returncode zurück; der Name des Users ist im zweiten Insert enthalten
7	UNEXPECTED_RETURNCODE Von einer gerufenen Funktion wurde ein nicht erwarteter Returncode zurückgegeben; der Wert des Returncodes ist im zweiten Insert enthalten
8	KDCDEF_RUNNING Die Cluster-User-Datei ist bereits von einem KDCDEF-Lauf gesperrt
9	FILE_IS_EMPTY Die Cluster-User-Datei ist katalogisiert, aber leer
10	Fehler beim Aufruf getLocalHostName Der Fehlercode wird im zweiten Insert bereitgestellt.
11	NO_XCS_GROUP Der Rechner gehört zu keinem XCS-Verbund; die Funktion "distributed lock" steht damit nicht zur Verfügung.

K517 &PARAM30 nicht unterstuetzt fuer &PARAM30

K518 Die Datei &FNAM existiert bereits. Der Generierungslauf wird abgebrochen.

K521 MAX PRIVILEGED-LTERM=&UTMNAME ist nicht als Dialog-LTERM definiert.

K522 Hinweis: Es ist kein privilegiertes LTERM generiert (siehe MAX PRIVILEGED-LTERM=)

5.4 Meldungen des UTM-Tools KDCPSYSL

- K600** Programm &PRGNMSG &VERS gestartet.
- K602** Programm KDCPSYSL normal beendet - SYSLOG-Datei wurde aufbereitet.
- K604** Programm KDCPSYSL abnormal beendet - SYSLOG-Datei wurde nicht aufbereitet.
- K605** Fehler &ERROR.
- K607** DMS-Fehler &DMSE auf der Datei &LINK
Die DMS-Fehler sind auf [Seite 383](#) beschrieben.
- K611** Programmfehler in &PRGNMSG; Grund: &TRMA
- K621** NLS Katalog &NLSCAT fuer >&NLSLANG< nicht vorhanden
- K622** NLS-Initialisierung fuer >&NLSLANG< fehlgeschlagen

5.5 Meldungen der UTM-Tools KDCMMOD / KDCMTXT

- K650** Programm &PRGNMSG wurde abnormal beendet.
- K651** Programm Fehler (&TRMA)
- K652** DMS Fehler &DMSE fuer Datei &FNAM.
Die DMS-Fehler sind auf [Seite 383](#) beschrieben.
- K653** Zeile &LINENR: Pflichtoperand fehlt.
- K654** Zeile &LINENR: Syntax Fehler.
- K655** Zeile &LINENR: Insert &INSMMSG ist verboten fuer Meldung &IDMSG. **K656** Zeile &LINENR: Konstante &CONMSG wurde nicht definiert.
- K657** Ende von SYSDTA wurde erreicht. END-Steueranweisungen wurden generiert.
- K658** * * * * * Steueranweisung wurde ignoriert. * * * * *
- K659** Datei &FNAM ist keine erlaubte Meldungsdefinitionsdatei.
- K660** Programm &PRGNMSG &VERS wurde gestartet.
- K661** Programm &PRGNMSG wurde normal beendet.
- K662** Zeile &LINENR: Funktionseinheit &FUMSG wurde nicht definiert.
- K663** Zeile &LINENR: Sprache &LANGMSG ist nicht definiert fuer Funktionseinheit &FUMSG.
- K664** Zeile &LINENR: Meldung &IDMSG ist verboten fuer Funktionseinheit &FUMSG.
- K665** Zeile &LINENR: GEN-Steueranweisung bereits eingegeben.
- K666** Zeile &LINENR: Meldungsziel &DESTMSG ist fuer Meldung &IDMSG verboten.
- K667** Zeile &LINENR: Meldungsziel &DESTMSG wird fuer Meldung &IDMSG gefordert.
- K668** Zeile &LINENR: GEN-Steueranweisung fehlt.
- K669** Zeile &LINENR: Funktionseinheit &FUMSG darf nicht modifiziert werden.
- K670** Quelldatei &FNAM fuer Meldungsmodul wurde erzeugt.
- K671** Quelldatei fuer Meldungsmodul wurde nicht erzeugt.
- K672** Zeile &LINENR: Konstante &CONMSG wurde bereits definiert.
- K673** Falsche Version &VERS der Meldungsdefinitionsdatei &FNAM.
- K681** NLS-Quelldatei &FNAM wurde erzeugt.
- K682** NLS-Quelldatei &FNAM wurde nicht erzeugt.

- K686** Zeile &LINENR: Text fuer Meldung &IDMSG laenger als &MSGMAXL.
Der Text der Meldung ist einschließlich der Inserts länger als 512 Zeichen. Die Dienstprogramme KDCMMOD und KDCMTXT können den Text nicht verarbeiten. KDCMMOD erzeugt keine Source für einen neuen Meldungsmodul. KDCMTXT ergänzt die Meldungsdefinitionsdatei nicht.
- K687** Zeile &LINENR: Warnung - Text fuer Meldung &IDMSG (SYSLINE) laenger als &MSGMAXL.
Der Text der Meldung (&IDMSG) mit dem Ziel SYSLINE ist einschließlich Inserts länger als 40 Zeichen. Die Dienstprogramme KDCMMOD und KDCMTXT akzeptieren den Meldungstext. Bei der späteren Ausgabe der Meldung in die Systemzeile gibt UTM nur die ersten 40 Zeichen des Meldungstextes aus.
- K688** Zeile &LINENR: Meldungsnummern > 999 sind verboten.
Es wurde eine Meldungsnummer größer 999 angegeben. Die Dienstprogramme KDCMMOD und KDCMTXT lehnen dies ab. KDCMMOD erzeugt keine Source für einen neuen Meldungsmodul. KDCMTXT ergänzt die Meldungsdefinitionsdatei nicht.
- K690** Text fuer Meldung &IDMSG in Sprache &LANGMSG und Funktionseinheit &FUMSG wurde nicht definiert.
- K691** Meldungsdefinitionsdatei &FNAM wurde erzeugt.
- K692** Meldungsdefinitionsdatei wurde nicht erzeugt.
- K693** Meldungsdefinitionsdatei &FNAM wurde veraendert.
- K694** Meldungsdefinitionsdatei wurde nicht veraendert.
- K695** Zeile &LINENR: Insert &INSMSG wurde nicht definiert.
- K696** Zeile &LINENR: FU/Konstante/Insert &CONMSG wurde bereits definiert.
- K697** Zeile &LINENR: Falscher Wert fuer Quell-Laenge wurde durch Default-Wert ersetzt.
- K698** Zeile &LINENR: Bereich bereits fuer andere Funktionseinheit genutzt.

5.6 Meldungen des UTM-Tools KDCDUMP

Die Meldungen von KDCDUMP sind ausschließlich auf Englisch verfügbar.

- K700** KDCDUMP &VERS started.
- K701** Difference between the number of existing UTM tables and the number of UTM tables which are known by KDCDUMP.
UTM area(s): &UTMA1&UTMA2&UTMA3&UTMA4&UTMA5&UTMA6&UTMA7
- K702** Abnormal end
- K703** Term application reason &TRMA
- K704** Table entry index not valid.
- K705** UTM dump &FNAM or part of it read in memory.
- K706** This type of preparation is not allowed for table &TABNAM.
- K707** No further process is available.
- K708** The editor given by the editor shell variable does not exist.
- K709** Only one entry of a slot table can be prepared.
- K710** Error on command
- K711** No UTM dump in memory.
- K712** Requested address X'&SADDR' not in present UTM dump or selected domain.
- K713** Requested address X'&SADDR' is present in table &TABNAM, but not within one table entry.
- K714** Error on link &LNAM: DMS return code &DMSE.
- K716** Open error on file &FNAM. DMS return code : &DMSE.
- K717** Request memory error
- K718** File &FNAM does not contain a UTM dump.
- K719** KDCDUMP &VER1 cannot prepare a UTM dump &VER3.
- K720** Only the directory of the UTM dump exists.
- K721** &UTMA1 is not present in UTM-dump or in selected domain.
- K722** &NUM1 bourse cycle with the criterias CREF=&CREF and ANNO=&ANNO1 is found.
- K723** Write error on file &FNAM. DMS return code: &DMSE.
- K724** Command not allowed at present time.

- K725** Table index is too low or too high.
- K726** Address is too low or too high.
- K727** FSTAT: Error on file &FNAM. DMS return code: &DMSE.
- K728** Normal end
- K729** Output file &FNAM is written.
- K730** Read error on file &FNAM. DMS return code: &DMSE.
- K731** The name &DEFTYPE is not a valid UTM type.
- K732** UTM dump does not contain UTM type &DEFTYPE.
- K733** Table name &TABNAM does not exist.
- K734** The UTM dump &FNAM could not be written completely. Last written UTM area is &UTMA1.

Diese Meldung besagt, dass der UTM-Dumperzeuger den UTM-Dump nicht vollständig schreiben konnte. Es stand u.U. zu wenig Platz auf der Kennung zur Verfügung. Die UTM-Bereiche werden in folgender Reihenfolge abgezogen:

CAA, SLOT, CACHE, MGP, XAPTP-GLOBAL, KTA, XAPTP-LOCAL, STACK, ROOT, User File, Journal File 1, Journal File 2, Buffer Segments, GSSB File, Lock File, CFG File, ULS File

Bei einem PEND ER-Dump wird nur ROOT abgezogen.

Die einzelnen Tabellen zu den UTM-Bereichen lassen sich durch die KDCDUMP-Anweisung HELP TABLE-NAMES ([Seite 73](#)) ausgeben, und zwar in der Reihenfolge des Speicherabzugs.

Die XAPTP-Teile bilden hier zusammen einen Block.

- K735** Table &TABNAM does not exist in UTM dump or in selected domain.
- K736** Index out of range - index is set to lowest or highest allowed table index.
- K737** End-index lower than start-index - end-index is given the value of the start-index.
- K740** Abbreviation of table name ambiguous with regard to &TABNAM &TABNAM1 &TABNAM2 &TABNAM3 &TABNAM4 &TABNAM5.
- K741** Information not available. Table name: &TABNAM
- Die Meldung besagt, dass ein UTM-Bereich oder eine Tabelle vom UTM-Dump-Erzeuger nicht abgezogen werden konnte. Der Grund ist, dass der UTM-Bereich oder die UTM-Tabelle nicht vollständig zugreifbar waren.
- K742** The KDCDUMP version on the hardware &HW1 with &OS1 as operating system cannot read a UTM dump written on the hardware &HW2 with the operating system &OS2.

- K743** For the table &TABNAM no displacement is possible.
- K744** Offset lower than 0 or higher than the no. of bytes for one table entry.
- K745** No symbolic preparation for this table &TABNAM possible.
- K746** Error on symbolic preparation.
- K747** Command has no result.
- K753** The name &FIRES is not a UTM resource of UTM type &DEFTYPE.
- K754** No further entry with this UTM type in this dictionary table.
- K755** Input error: No further command can be read.
- K756** False value for DB operand. Only &DBMAX database(s) are generated.
- K757** The version of &UTMA1 (&VER1) is not compatible with the version of KDCDUMP (&VER2).
- K758** No summary was written.
- K759** Input string is longer than the permitted length (256 char).
- K760** No entry with the name &FIRES found.
- K761** No further entry with the name &FIRES found.
- K770** Error detect while checking compression of file &FNAM. Error code: &ERRC.
- K771** Error detect while creating temporary file for &FNAM. Error code: &ERRC.
- K772** Error detect while decompressing file &FNAM. Error code: &ERRC.
- K773** Requested information can not be found in the selected domain.
- K774** Requested domain for command SFIND not read from dump file.
Empfehlung: Domain einlesen mit FILE=<dumpfile>, DOMAIN=<domain> und SFIND-Befehl wiederholen.
- K780** No dump file was closed.
- K781** Compressed dump file could not be uncompressed.
- K782** No hit found.
- K783** Value of HITS is neither ALL nor between 0 and 32767.
- K784** Size of address greater than 32 bit is forbidden.

5.7 Meldungstexte des UTM-Tools KDCUPD

Das UTM-Tool KDCUPD gibt sowohl Meldungen des Transaktionsmonitors als auch eigene Meldungen aus. Bei allen Meldungen von KDCUPD, bei denen der Text mit einem „*“ (Stern) beginnt, konnten die Daten nicht übertragen werden.

- K800** KDCUPD &BMD1 &VERS gestartet.
- K801** Bitte Steueranweisungen eingeben
- K802** Steueranweisung KDCFILE/CHECK: Parameter &UPDCMD ist nicht angegeben.
- K803** Basisnamen fuer alte und neue &PARAM17 muessen unterschiedlich sein.
- K804** Uebertragung von UTM &UPDVERS nach UTM &UPDVERS nicht unterstuetzt.
- K805** Konsistenz-Pruefung in Ordnung fuer Filebase &FBASUPD.
- K806** Alle gewuenschten Daten wurden uebertragen.
- K807** Gewuenschte Daten wurden teilweise uebertragen.
- K808** Keine Daten aus &FNAM zu transferieren.
Die bei Parameter OLD angegebene Datei stammt nur aus einer KDCDEF-Generierung. Ein KDCUPD-Lauf ist daher nicht sinnvoll!
- K809** Steueranweisung KDCFILE/CHECK: Basisname &FBASUPD ist zu lang.
- K810** Kombination der Kommandos CHECK und KDCFILE ist nicht erlaubt.
- K812** Abnormale Beendigung von KDCUPD.
- K813** Normale Beendigung von KDCUPD.
- K814** CLUSTER-Transfer unterschiedlicher Versionen nicht unterstuetzt.
- K831** KDXUPDS: Fehler beim Lesen von KDCFILE.
- K832** KDXUPDX: Ungueltige Anzahl von Argumenten.
- K833** &PRGUPDX: SHARED-MEMORY kann nicht erzeugt werden. Fehler: &ERRNO.
- K834** &PRGUPDX: Auf SHARED-MEMORY kann nicht zugegriffen werden. Fehler: &ERRNO.
- K835** Prozess &UPDMODUL kann nicht erzeugt werden. Fehler: &ERRNO.
- K836** Prozess &UPDMODUL kann nicht aufgerufen werden. Fehler: &ERRNO.
- K837** KDCUPD: Ungueltiger Kind-Prozess gestorben.
- K838** KDCUPD: Kind-Prozess &UPDMODUL gestorben.
- K839** KDCUPD: Eltern-Prozess gestorben.

K840 &PRGUPDX: Error &ERRNO bei Semaphor-Operation &SEMOP.

K841 KDCUPD: Transfer von 32-Bit zu 64-Bit Architektur.

Zusätzlich werden noch Informations-Meldungen mit dem Präfix "KDCUPD:" ausgegeben, an denen Details zu den gestarteten Hilfs-Prozessen abgelesen werden können.

Beispiel

Informations-Meldungen auf einem Unix-System, wenn openUTM unter dem Verzeichnis "/opt/lib/utm63a00" installiert ist:

```
KDCUPD: full Path of READ process: "/opt/lib/utm63a00/64/./32/ex/kdcrV63A"
KDCUPD: full Path of WRITE process: "/opt/lib/utm63a00/64/ex/kdcwV63A"
```

K851 &UPDTYP-Daten uebertragen. KCRN = &UKCRN,&PARAM4 = &UKCLA.

K852 &PARAM3-Daten uebertragen. KCRN = &UKCRN,&PARAM4 = &UKCLT, KCLA = &UKCLA.

K853 Datenbank-Generierung. Nummer: &DBCOUNT FILEOLD: &DBOLD FILENEW: &DBNEW

K854 Warnung: Die Komplexitaetsstufe fuer USER &USER hat sich geaendert. Passwort kann ungueltig werden.

Die Komplexitätsstufe des Passworts für den User &USER ist in der neuen KDCFILE höher als in der alten KDCFILE, siehe USER-Anweisung in KDCDEF-Generierung. Wenn das übertragene Passwort diese Bedingung nicht erfüllt, kann sich der Benutzer nicht anmelden.

Maßnahme: Der Administrator muss dann ein neues Passwort vergeben.

K855 * &UPDTYP-Daten nicht uebertragen. KCRN = &UKCRN, &PARAM4 = &UKCLA, KCRCCC = &RCCC, KCRCDC = &RCDC.

K856 * &PARAM3-Daten nicht uebertragen. KCRN = &UKCRN, &PARAM4 = &UKCLT, KCLA = &UKCLA, KCRCCC = &RCCC, KCRCDC = &RCDC.

K857 * Load Module &PROG nicht gefunden. Aktuelle Version &PVER nicht uebertragen.

K858 Aktuelle Version &PVER von Load Module &PROG uebertragen.

5.8 U-Meldungen

5.8.1 Meldungen des Dialog-Terminalprozesses

- U101** Argumente fuer utmdtp-Prozess sind ungueltig
- U102** Eingabe-Umlenkung fuer utmdtp-Prozess ist nicht erlaubt
- U103** Bitte warten ...
- U104** Bitte Daten eingeben ...
- U106** Bitte Anwendungsnamen eingeben:
- U107** Benutzername:
- U108** Passwort:
- U109** Betriebsmittel-Engpass im UTM-System
- U110** Geraet ist schon mit UTM-Anwendung &APPL verbunden
- U111** UTM-Anwendung &APPL ist nicht gestartet
- U112** UTM-Anwendung &APPL hat sich beendet
- U113** UTM-Anwendung &APPL wird gerade gestartet oder hat sich beendet
- U114** Verbindung von UTM-Anwendung &APPL zurueckgewiesen
- U115** Anmeldung mit Login-Name von UTM-Anwendung &APPL abgewiesen
- U116** Verbindung von UTM-Anwendung &APPL abgebaut
- U117** Keine Antwort von UTM-Anwendung &APPL in Wartezeit erhalten
- U118** Eingabe mit "END"-Taste beendet
- U119** Eingabe mit "DEL"-Taste beendet

U120 utmdtp-Prozess beendet sich wegen Fehlernummer &UERRNO

Das Insert &UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung	Ursache / Maßnahme
1	interner Fehler utmdtp.	PM
71	negative Länge für Linemode-Ausgabe	PM
72	Fehler bei write() für Linemode-Ausgabe.	PM
80	Umgebungsvariablen nicht korrekt gesetzt: – \$LANG \$NLSPATH nicht korrekt. – \$LINES < 25 oder \$COLUMN < 80	– Variablen korrigieren – Auf 25 bzw. 80 setzen – Bei graphischen Oberflächen em97801 mit TERM=97801 verwenden. Andere Ursache: PM
81	Fehler bei n_cpi.	PM
82	Umgebungsvariable \$LINES<25 oder \$COLUMNS<80	auf 25 / 80 setzen
83	Fehler bei n_chr.	PM
84	Fehler bei n_wir.	PM
85	Fehler bei n_wrt.	PM
86	Fehler bei n_ger.	PM
88	Fehler bei n_csm.	PM
89	Fehler bei n_dpi.	PM
90	Fehler bei n_cls.	PM
91	Fehler bei n_rac.	PM
92	Fehler bei n_mom.	PM
95	Fehler bei f_open Mögliche Ursachen: wie bei &UERRNO=80	wie bei &UERRNO=80
96	Fehler bei f_read	PM
99	Fehler bei f_switch	PM

Tritt diese Meldung auf, wird zur Diagnose mit abort() ein core-Dump erzeugt, der als notwendige Unterlage zur PM auf Datenträger mitgeliefert werden soll.

U121 USER mit Login-Name bereits angeschlossen

U123 Neues Passwort:

U124 Wiederholung neues Passwort:

U125 utmdtp Prozess wird durch kdcrem beendet

5.8.2 Meldungen des Printerprozesses

U151 utmprint: PTERM &PTRM beendet sich wegen Fehlernummer &UERRNO

Das Insert &UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung	Maßnahme
1	interner Fehler utmprint.	PM
50	Verbindungsaufbauwunsch des utmprint wurde abgewiesen.	PM
60	Tiam-Protokoll ungültig.	PM
71	negative Länge für Linemode-Ausgabe.	PM
72	Fehler bei Linemode-Ausgabe.	PM
131	Shell-Skript „utmlp“ weder unter \$PATH noch unter <i>utmpfad</i> /shsc gefunden oder keine Ausführungsberechtigung	UTM-Installationsfehler Skript bereitstellen

Tritt diese Meldung auf, wird zur Diagnose mit abort() ein core-Dump erzeugt, der als notwendige Unterlage zur Problemmeldung auf Datenträger mitgeliefert werden soll.

U154 utmprint: PTERM &PTRM ist keine konfigurierte Druckergruppe fuer lpr

U155 utmprint: Argumente fuer utmprint-Prozess sind ungueltig

U156 utmprint: PTERM &PTRM Fehler bei Ausgabe errno: &ERRNO

U157 utmprint: PTERM &PTRM Fehler bei Ausgabe mit utmlp exitcode: &EXITC

5.8.3 Meldungen des utmlog-Prozesses

- U171** utmlog: Anzahl der Argumente ungueltig
- U172** utmlog-Prozess wird beendet. Fehler beim Zugriff auf UTM-Logpipe. CMD = &CMD
errno = &ERRNO
- U173** utmlog-Prozess wird beendet. Fehler beim Lesen UTM-Logpipe (Satzkopf): errno =
&ERRNO, Rueckgabewert = &RETVALUE
- U174** utmlog-Prozess wird beendet. Fehler beim Lesen UTM-Logpipe (Satz): errno =
&ERRNO, Rueckgabewert = &RETVALUE, Sollwert = &NBRBYTES
- U175** utmlog: Fehlerhafte Identifikation in UTM-Logging-Satz (&LOGREC)
- U176** utmlog-Prozess wird beendet. Fehler beim Aufruf von &CMD fuer Datei = &FNAM,
errno = &ERRNO
- U177** utmlog-Prozess wird normal beendet (&FNAM)

5.8.4 Allgemeine U-Meldungen

U181 Programm &OBJ1 &VERS gestartet (pid: &PID, &STRTIME)

U182 &OBJ3 &VERS und &OBJ2 &VERS sind nicht vertraglich

U184 &OBJ1 DMS Fehler &DMSE fuer Datei &FNAM aufgetreten

Die möglichen Fehlercodes, die im Insert &DMSE ausgegeben werden, sind auf [Seite 383](#) beschrieben.

U185 kdcdef waehrend Anwendungslauf nicht erlaubt

U186 &OBJ1 KCSTRMA mit Grund &TRMA aufgerufen (&STRTIME)

U187 &OBJ1 Verwendeter applifile: &OBJ3

U188 &OBJ1 UTMPATH ist nicht gesetzt

U189 &OBJ1 (&PTRM, &PRNM): IPC Engpass &IPCOBJ &IPCREAS

Die Inserts &IPCOBJ und &IPCREAS haben folgende Bedeutung:

&IPCOBJ	&IPCREAS	Bedeutung	Maßnahme
NET	NOT ATT	UTM Netzprozess noch nicht gestartet	normales Verhalten bei Anwendungsstart
NET	PROC DEAD	UTM Netzprozess hat sich beendet	siehe Meldung vom Netzprozess
TSAP	tsapname	UTM Netzprozess konnte sich nicht mit dem BCAMAPPL bzw. ACCESS-POINT (tsapname) beim Transportssystem anmelden	normales Verhalten bei Anwendungsstart
EXTP EXTP EXTP	WORK USED NET USED EXTP USED	Prozessverwaltungstabelle vollständig belegt	UTM-Generierung bzgl. Semaphore überprüfen
SEMA	USED	alle Semaphore belegt	UTM-Generierung bzgl. Semaphore überprüfen
LETT	IPC FULL	Datenbereich ist bereits zu voll um noch Daten für diese Verbindung aufzunehmen	UTM-Generierung bzw. Umgebungsvariable UTM_IPC_LETTER überprüfen. (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen“)

&IPCOBJ	&IPCREAS	Bedeutung	Maßnahme
LETT	EXTP FULL	Datenbereich für Empfangs-Nachrichten ist bereits vollständig belegt	UTM-Generierung bzw. Umgebungsvariable UTM_IPC_LETTER überprüfen. (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen“)
LETT	USED	Datenbereich ist belegt	UTM-Generierung bzw. Umgebungsvariable UTM_IPC_LETTER überprüfen. (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen“)
LETT LETT	MAX ILETT MAX OLETT	maximaler Datenbereich pro Verbindung belegt	UTM-Generierung überprüfen bzw. mittels der Umgebungsvariablen UTM_IPC_EXTP_LETTER vor dem nächsten Start den Datenbereich der Verbindung im Shared Memory vergrößern (siehe openUTM-Handbuch „Einsatz von openUTM-Anwendungen“) (Voreinstellung: 64 KB, erlaubte Werte: Vielfache von 4KB).
ANNO	USED	alle IPC ANNOs belegt	PM
ELEM	USED	alle IPC ELEMENTs belegt	PM

U190 &OBJ1 SHM Fehler (Schlüssel: &SHMKEY, Laenge: &SHMLTH): &UERRNO

Das Insert &UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung	Ursache
1	IPC Shared Memory kann in der angeforderten Größe nicht eingerichtet werden.	Anwenderfehler Maßnahme: Ggf. muss die Maschine getuned werden. Hinweise hierzu entnehmen Sie bitte der Freigabemitteilung.
2	IPC Shared Memory kann nicht eingerichtet werden, da bereits vorhanden.	Anwenderfehler
3	Fehler beim Einrichten des IPC Shared Memory.	Systemfehler

&UERRNO	Bedeutung	Ursache
4	IPC Shared Memory kann in der angeforderten Größe nicht in den Prozessadressraum geladen werden.	Anwenderfehler
5	Fehler beim Laden des IPC Shared Memory in den Prozessadressraum.	Systemfehler

U191 NLS-Initialisierung fuer >&NLSLANG< fehlgeschlagen

U192 &OBJ1Systemaufruf &SCALL fehlgeschlagen; rc: &RETVALUE errno: &UERRNO

U193 &OBJ1Interner Fehler '&UABNREAS' verursacht eine abnormale Beendigung des Anwendungslaufs

5.8.5 Meldungen des Timerprozesses

U201 utmtimer: Anzahl der Argumente ungueltig

U202 utmtimer: ungueltiger Anwendungsname &PARAM10

U203 utmtimer: Fuer die Anwendung &APPL ist bereits ein utmtimer-Prozess vorhanden

U204 utmtimer: Anwendung &APPL ist nicht vorhanden in &OBJ3

U205 utmtimer: Fehler &UERRNO waehrend utmtimer-Ablauf

Das Insert UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung	Maßnahme
1	Parameterfehler beim Anmelden, utmtimer inkonsistent gebunden	PM
2	Pufferengpass beim Anmelden	PM
3	Timeout beim Anmelden an das IPC Shared Memory mögliche Ursache: Dauer des Startexits 8 Minuten	Dauer des Anwendungsstarts überprüfen
9	Unerwarteter Returncode beim Anmelden	PM
10	Speicherengpass beim Vergrößern der Timerliste	
11	Parameterfehler beim Empfangen, utmtimer inkonsistent gebunden	PM
12	Workprozess beim Empfangen nicht vorhanden, da Anwendungsende	
19	Unerwarteter Returncode beim Empfangen	PM
21	Parameterfehler beim Senden, utmtimer inkonsistent gebunden	PM
22	Workprozess beim Senden nicht vorhanden, da Anwendungsende	
29	Unerwarteter Returncode beim Senden	PM
31	Parameterfehler beim Abmelden, utmtimer inkonsistent gebunden	PM
32	Kommunikations Shared Memory ist gesperrt beim Abmelden. Kann in gewissen Situationen nach Fehler 12 oder 22 auftreten.	
40	Die Signal Handler Routine für das Signal SIGALRM konnte nicht gesetzt werden.	PM + Unterlagen

U206 utmtimer: Empfangene Nachricht hat falschen Typ

U207 utmtimer: Vergroesserung der Timerliste von &UDIA1 auf &UDIA2 Elemente

5.8.6 Meldungen des utmmain-Prozesses

- U221** &OBJ1 UTM-Anwendung &APPL &VERS beendet (&STRTIME)
- U222** &OBJ1 Anzahl der Argumente ungueltig
- U223** &OBJ1 Laut internem Status laeuft die UTM-Anwendung &APPL, applifile: &OBJ3
Es existiert eine laufende Anwendung dieses Namens.
Das kann folgende Ursachen haben:
- Der gewählte Anwendungsname ist nicht systemweit eindeutig,
Maßnahme: beim UTM-Administrator nachfragen
 - Die eigene UTM-Anwendung lief im TEST Betrieb. Anschließend wurde nicht durch Aufruf des Dienstprogrammes kdcrem der Grundzustand hergestellt.
Maßnahme: kdcrem aufrufen.
 - Die Anwendung wurde nicht vor shutdown des Betriebssystems beendet.
Maßnahme: kdcrem aufrufen.
 - Die Anwendung lief als Produktiv-Anwendung mit Schalter TEST. Nach dem normalen Beenden der Anwendung wird der utmmain-Prozess nicht automatisch beendet.
Maßnahme: utmmain-Prozess mit kill -9 beenden.
- U224** &OBJ1 Beendigung von KDCROOT
- U225** utmmain: Fehler beim Einrichten der Pipe &FNAM,errno: &ERRNO
Es existiert noch eine Pipe von einem vorhergehenden Anwendungslauf.
Maßnahme: kdcrem aufrufen
- U227** &OBJ1 UTM-Anwendung &APPL beendet durch kdcrem
- U228** utmmain: Fehler beim Lesen aus der Pipe, errno: &ERRNO
- U229** utmmain: &OBJ1-Prozess gestorben, pid: &PID, &SIGEXIT(&STRTIME)
- U230** utmmain: utmwork-Prozess gestorben, pid: &PID utmwork wird nachgestartet (&STRTIME)
- U231** utmmain: utmwork-Prozess gestorben, pid :&PID
Unerwarteter exitcode: &EXTCODE &SIGEXIT(&STRTIME)
Diese Meldung wird erzeugt, wenn sich ein utmwork-Prozess nicht unter Kontrolle von openUTM beendet, z.B. bei Beendigung durch kill -9.
Für die UTM-Anwendung geht dadurch ein Workprozess verloren und wird nicht nachgestartet. Zur Vermeidung von Inkonsistenzen wird in diesem Fall die UTM-Anwendung abnormal beendet. Das wird mit der Meldung U221 angezeigt.

Die folgende Liste zählt einige der Ursachen auf, die dazu führen, dass sich Workprozesse nicht unter Kontrolle von openUTM beenden:

- Start der Anwendung mit STXIT = OFF und Auftreten eines Signals
- utmwork-Prozesse erhalten Signale außerhalb von openUTM
- Ein Teilprogrammlauf überschreibt Signalroutinen
- Die Laufzeitumgebung von COBOL oder C++ veranlasst eine Prozessbeendigung
- Die Datenbank beendet den Prozess

Maßnahme:

Beheben Sie das Problem das zur Beendigung des Prozesses geführt hat und starten Sie die UTM-Anwendung neu.

U232 &OBJ1 Fehler &UERRNO waehrend utmmain-Ablauf

Die Werte von &UERRNO und deren Bedeutung sind in der folgenden Tabelle aufgelistet. Zu den zugehörigen Maßnahmen steht in den meisten Fehlerfällen zusätzliche Information auf *stderr* zur Verfügung.

&UERRNO	Bedeutung	Maßnahme
1	Stand-alone Anwendung: FILEBASE-Parameter in Startparameterdatei nicht vorhanden.	Anwenderfehler
	UTM-Cluster-Anwendung: CLUSTER-FILEBASE-Parameter in Startparameterdatei nicht vorhanden.	
2	utmmain konnte sich nicht an die Shared Memory-Bereiche der Anwendung anmelden.	Anwenderfehler
11	Erster utmwork-Prozess unmittelbar nach dem Start beendet.	Siehe Meldungen von utmwork
12	utmwork-Prozess undefiniert beendet.	Siehe Meldungen von utmwork
13	Ungültiger Auftrag von utmwork empfangen.	PM + Unterlagen
21	Speicherengpass beim Anfordern der Lockverwaltung für Cluster-Konfigurationsdatei (nur bei UTM-Cluster-Anwendungen).	Speicherbedarf prüfen
22	utmmain konnte keinen Shared Lock für die Cluster-Konfigurationsdatei setzen (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
23	Speicherengpass beim Anfordern des Bereichs zum Einlesen der Cluster-Konfigurationsdatei (nur bei UTM-Cluster-Anwendungen).	Speicherbedarf prüfen

&UERRNO	Bedeutung	Maßnahme
24	Cluster-Konfigurationsdatei hat inkonsistenten Inhalt (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
25	Ungültige Version in Cluster-Konfigurationsdatei (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
26	Ungültiger Typ des Betriebssystems in Cluster-Konfigurationsdatei (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
27	Ungültiger Bit Mode in Cluster-Konfigurationsdatei (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
28	Name des lokalen Rechners konnte nicht ausgewertet werden (nur bei UTM-Cluster-Anwendungen).	PM + Unterlagen
29	Knoteneintrag für lokalen Rechner konnte nicht in Cluster-Konfigurationsdatei gefunden werden (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
30	Speicherengpass beim Anfordern der Lockverwaltung für KDCFILE (nur bei UTM-Cluster-Anwendungen).	Speicherbedarf prüfen
31	utmmain konnte keinen Shared Lock für die KDCFILE setzen (nur bei UTM-Cluster-Anwendungen).	Anwenderfehler
32	Datei ist keine Cluster-Konfigurationsdatei	Anwenderfehler
33	Der bei der Knoten-Recovery angegebene Knotenname konnte nicht in der Cluster-Konfigurationsdatei gefunden werden.	Anwenderfehler

U233 &OBJ1 &UREAS

U234 &OBJ1 fork() Fehler &ERRNO; Prozesstyp: &PRTYPE

U235 &OBJ1 Startfehler &ERRNO fuer Datei &FNAM aufgetreten

U236 &OBJ1 Starten von &FNAM, pid: &PID

U237 &OBJ1 Fehler beim Loeschen eines Semaphors, key: &SEMKEY, error: &ERRNO

U238 &OBJ1 Fehler beim Anmelden an ein Shared Memory, key: &SHMKEY, error: &ERRNO

U239 &OBJ1 Fehler beim Loeschen eines Shared Memories, key: &SHMKEY, error: &ERRNO

U240 utmmain zum Testen im Dialog gestartet

U241 &OBJ1 <FILEBASE> mehr als 29 Zeichen oder max. Nr. Prozesse

U242 &OBJ1 &FBTYPE Namen inkonsistent, Grund: &UERRNO
&FILEBAS1
&FILEBAS2

Das Insert &FBTYPE hat folgende Bedeutung:

&FBTYPE	Bedeutung
FILEBASE	FILEBASE-Angabe in Startparameterdatei
ARGUMENT	FILEBASE-Angabe als Argument von utmmain
CLUSTER-FILEBASE	CLUSTER-FILEBASE-Angabe in Startparameterdatei
CONFIG-FILE	FILEBASE-Angabe in Cluster-Konfigurationsdatei
NODE-RECOVERY-DIR	Verzeichnis für Knoten-Recovery

Das Insert &UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung
1	Angabe des Namens syntaktisch falsch (ohne '=').
2	Fehlende Angabe nach dem '=' Zeichen.
3	Fehler beim Wechseln des Dateiverzeichnisses von &FILEBAS1 nach &FILEBAS2.
4	Der vollqualifizierte Verzeichnisname ist zu lang.
5	Die Angabe des Filebase-Namen als Argument von utmmain stimmt nicht mit der Angabe in der Startparameterdatei (bei stand-alone Anwendungen) bzw. in der Cluster-Konfigurationsdatei (bei UTM-Cluster-Anwendungen) überein.
6	Das Verzeichnis, in dem utmmain gestartet wurde, stimmt nicht mit dem filebase-Verzeichnis der Knoten-Anwendung überein, für die eine Knoten-Recovery durchgeführt werden soll.

Die Inserts &FILEBAS1 und &FILEBAS2 haben folgende Bedeutung:

Insert	Bedeutung
&FILEBAS1	Angabe in Startparameterdatei
&FILEBAS2	Angabe als Argument

U244 utmmain: Bitte &FNAM starten mit Argumenten:

&VERS &APPL &FILEBASE &FNAM &PID &WID &STIND &WTYP

U245 &OBJ1 Die System-Datei stdout wird umgeschaltet von &FNAM auf Datei &FNAM.

U246 &OBJ1 Die System-Datei stderr wird umgeschaltet von &FNAM auf Datei &FNAM.

U247 &OBJ1 Neue System-Datei &FNAM geöffnet. Vorige System-Datei war &FNAM.

5.8.7 Meldungen des Dienstprogramms kdcuslog und kdcslog

- U251** &PARAM10 Anzahl der Argumente ungueltig
- U252** &PARAM10 FILEBASE-name &FILEBASE ungueltig
- U253** &PARAM10 Wert &GENUSL fuer Anzahl der Generationen ungueltig
- U254** &PARAM10 Wert &ARG2 fuer Art der Dateifuehrung ungueltig
- U255** &PARAM10 Fehler &DMSE beim Einrichten des Dateiverzeichnisses &DIRECT
- U256** &PARAM10 Dateiverzeichnis &DIRECT eingerichtet
- U257** &PARAM10 Fehler &DMSE beim Einrichten der Dateien fuer &FNAM
- U258** &PARAM10 FGG-Dateien fuer &FNAM eingerichtet
- U259** &PARAM10 &OBJ1 waehrend Anwendungslauf nicht erlaubt

5.8.8 Meldungen des Dienstprogramms kdccsysl

- U271** kdccsysl: Anzahl der Argumente ungueltig
- U272** kdccsysl: Dateiname &FNAM ungueltig
- U273** kdccsysl: Eingabedatei = Ausgabedatei ist nicht erlaubt
- U274** kdccsysl: Fehler &ERRNO beim Oeffnen der Datei &FNAM
- U275** kdccsysl: Fehler &ERRNO beim Kreieren der Datei &FNAM
- U276** kdccsysl: Datei &FNAM ist leer
- U277** kdccsysl: Fehler &ERRNO beim Lesen der Datei &FNAM
- U278** kdccsysl: Datei &FNAM ist keine SYSLOG-Datei
- U279** kdccsysl: Fehler &ERRNO beim Schreiben der Datei &FNAM
- U280** kdccsysl: Datei &FNAM kann nicht geschrieben werden
- U281** kdccsysl: normale Beendigung - SYSLOG-Datei konvertiert

5.8.9 Meldungen des Netzprozesses

- U301** &OBJ1 (pid: &PID): Anzahl der Argumente ungueltig
- U302** &OBJ1 (pid: &PID, &TNSNAME): &TNSPROP : Fehler &TNSCODE &TNSCLASS &TNSVALUE
- U303** &OBJ1 (pid: &PID, &NETPROC): &TNSPROP existiert nicht fuer &TNSNAME
- U304** &OBJ1 (pid: &PID, &TNSNAME): &NETFCT-Aufruf: Fehler &NETERR

Die Fehlerursachen in U304 beziehen sich auf die UTM-interne Abbildung der CMX-Transportschnittstelle auf die Socket-Schnittstelle im Prozess utmnets. Die Bedeutung der Inserts NETFCT und NETERR finden Sie in der Tabelle auf [Seite 395](#).

- U305** &OBJ1 (pid: &PID): CMX-Anwendung &BCAP wurde bereits angemeldet
- U306** &OBJ1 (pid: &PID, &TNSNAME): Fehler &UERRNO waehrend Prozess-Ablauf

Das Insert UERRNO hat folgende Bedeutung:

&UERRNO	Bedeutung	Maßnahme
1	Parameter-Fehler beim Anmelden, Netzprozess inkonsistent gebunden	PM
2	Netzprozess zu oft gestartet	Anwenderfehler
4	Pufferengpass beim Anmelden	PM
9	Unerwarteter Returncode beim Anmelden	PM
11	Parameter-Fehler beim Empfangen, Netzprozess inkonsistent gebunden	PM
12	Workprozess beim Empfangen nicht vorhanden	normales Verhalten bei Anwendungsende
13	ungültige Nachricht vom Workprozess empfangen	PM
14	konkurrierender Verbindungsabbau	normales Verhalten
19	Unerwarteter Returncode beim Empfangen	PM
21	Parameter-Fehler beim Senden, Netzprozess inkonsistent gebunden	PM
22	Workprozess beim Senden nicht vorhanden, da Anwendungsende	normales Verhalten bei Anwendungsende
23	Pufferengpass beim Senden an Workprozess	normales Verhalten bei hoher Last
24	UTM Puffer beim Senden gesperrt	PM
25	Verbindung wurde von openUTM abgebaut	normales Verhalten

&UERRNO	Bedeutung	Maßnahme
26	Beim Senden wurde Verbindungsabbau erkannt	normales Verhalten
27	Beim Senden wurde auf die Freigabe eines Betriebsmittels gewartet. Hinweis: Diese Meldung wird ab der zweiten Sekunde pro Sekunde einmal ausgegeben.	normales Verhalten bei hoher Last
29	Unerwarteter Returncode beim Senden	PM
32	Verbindung konnte wegen CPU-Engpass nicht umgelenkt werden	normales Verhalten bei hoher Last
34	Speicherengpass beim Empfangen von Daten im Netzprozess	normales Verhalten bei hoher Last
35	Speicherengpass im Netzprozess beim Abspeichern einer Ausgangs-Nachricht wegen eines Sendeverbots	normales Verhalten bei hoher Last
36	Abgespeicherte Ausgangs-Nachricht nicht gefunden.	PM
37	Anmeldung für einen lokalen Kommunikationsendpunkt misslungen	Anwenderfehler: Generierung falsch oder unvollständig, oder Port von einer anderen Anwendung belegt. Wenn es sich um den impliziten BCAMAPPL handelt, der aus dem Applinamen gebildet wird, ist dies der Port 0. Der implizite BCAMAPPL kann auch explizit mit einem Port definiert werden; siehe U320 / U323
38	Ungültige Aufforderung zum Senden von Daten	PM
39	Länge der von ICMX(L) empfangenen Nachricht ist größer als 64K	Anwenderfehler: Partnersystem sendet zu große Nachricht
41	ungültiger Absender beim Verbindungsaufbauwunsch von openUTM	PM
42	Pufferengpass bei der Bearbeitung eines Verbindungsaufbauwunsches von openUTM	PM
43	interner Zustand passt nicht zur Verbindungsaufbaubestätigung von openUTM	PM

&UERRNO	Bedeutung	Maßnahme
44	Für den Hostnamen des Empfängers des Verbindungsaufbauwunsches von openUTM konnte keine IP-Adresse ermittelt werden	Anwenderfehler: TNS-Eintrag für Empfänger fehlt, Generierung falsch oder unvollständig
45	Konkurrierender Verbindungsabbau: Die Partner-Anwendung hat eine Verbindung bereits abgebaut, über die openUTM noch eine Nachricht senden will.	normales Verhalten
46	Ungültige Portnummer beim Verbindungsaufbauwunsch von openUTM	Anwenderfehler: Generierung falsch oder unvollständig
47	Fehler beim Holen der Adresse des Empfängers des Verbindungsaufbauwunsches von openUTM mit t_getaddr()	PM
48	Fehler beim Aufbau der Adresse des Empfängers des Verbindungsaufbauwunsches von openUTM mit t_setaddr()	PM
51	Keine Verbindungsumlenkungsanzeige erhalten	PM
52	Daten vor Verbindungsumlenkung empfangen	PM
53	Aufforderung zum Senden von Daten vor Verbindungsumlenkung erhalten	PM
54	Verbindungsabbau vor Verbindungsumlenkung erhalten	PM
55	ungültige Verbindungsumlenkungsanzeige erhalten	PM
56	tref von CMX stimmt nicht mit tref von redin überein	PM
57	Ungültiger Aufruf der callback Funktion	PM
58	Längenkonflikt beim Empfangen von Daten (> 64K)	Anwenderfehler: Länge der Daten ändern
82	Absender des Verbindungsaufbauwunsches von ICMX(L) konnte nicht ermittelt werden	Anwenderfehler: Generierung falsch oder unvollständig
83	Empfänger des Verbindungsaufbauwunsches von ICMX(L) ist nicht bekannt	PM
84	Pufferengpass beim Ermitteln der Verbindungs-Identifikation bei der Bearbeitung eines Verbindungsaufbauwunsches von ICMX(L)	Anwenderfehler: Anzahl Semaphore oder Anzahl maximaler Verbindungen erhöhen

&UERRNO	Bedeutung	Maßnahme
85	Workprozess beim Ermitteln der Verbindungs-Identifikation für eine ICMX(L)-Verbindung nicht vorhanden.	
86	Pufferengpass bei der Bearbeitung eines Verbindungsaufbauwunsches von ICMX(L)	Anwenderfehler: Anzahl Semaphore erhöhen
87	interner Zustand passt nicht zur Verbindungsaufbaubestätigung von ICMX(L)	PM
88	ungültige Verbindungsaufbaubestätigung von ICMX(L)	PM
89	ungültiger Verbindungsabbau von ICMX(L)	PM
100	Fehler beim Semaphor-Aufruf im IPC-Empfangsthread	PM bzw. normales Verhalten bei Anwendungsende
101	Fehler beim Senden einer Socket-Nachricht im IPC-Empfangsthread	PM
201	t_attach in Socketnetzprozess. Parameterfehler: Option Pointer ist NULL	PM
202	t_attach in Socketnetzprozess. Parameterfehler: ungültige Optionstruktur	PM
203	t_attach in Socketnetzprozess. Initialisierung socket-Umgebung misslungen	PM
204	t_attach in Socketnetzprozess. Zu wenig frei verfügbare Sockets	Anzahl Socket Netzprozesse erhöhen.
211	t_event in Socketnetzprozess. Parameterfehler: Option Pointer ist NULL	PM
212	t_event in Socketnetzprozess. Parameterfehler: ungültige Optionstruktur	PM
213	t_event in Socketnetzprozess. Parameterfehler: ungültiger Wert für Cmode	PM
214	t_event in Socketnetzprozess. Parameterfehler: ungültiger Verbindungsabbau gespeichert	PM
215	t_event in Socketnetzprozess. Zu wenig frei verfügbare Sockets	Anzahl Socket Netzprozesse erhöhen.
216	t_event in Socketnetzprozess. Beim select Aufruf kommt es zu einem unerwarteten Except socket.	Client darf keine Out-of-Band Daten schicken.
221	t_conrq in Socketnetzprozess. Parameterfehler: ungültige Optionstruktur	PM

&UERRNO	Bedeutung	Maßnahme
222	t_conrq in Socketnetzprozess. Zu wenig frei verfügbare Sockets	Anzahl Socket Netzprozesse erhöhen.
223	t_conrq in Socketnetzprozess. bind()-Aufruf für die Absenderadresse war nicht erfolgreich.	Generierung / Netz prüfen
224	t_conrq in Socketnetzprozess. Ungültige Empfängeradresse.	Generierung prüfen
226	t_conin in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
227	t_conin in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegt kein Verbindungsaufbau vor	PM
228	t_conin in Socketnetzprozess. Parameterfehler: die angegebene Transportreferenz ist nicht mehr gültig	PM
231	t_concf in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
232	t_concf in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegt bereits ein Verbindungsabbau vor	normales Verhalten
235	t_conrs in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
236	t_conrs in Socketnetzprozess. Parameterfehler: ungültige Optionstruktur	PM
237	t_conrs in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegt bereits ein Verbindungsabbau vor	normales Verhalten
238	t_conrs in Socketnetzprozess. Parameterfehler: die angegebene Transportreferenz ist nicht mehr gültig	normales Verhalten
241	t_datain in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
242	t_datain in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegen keine Daten vor	PM

&UERRNO	Bedeutung	Maßnahme
243	t_datin in Socketnetzprozess. Parameterfehler: ungültiger Wert für Datenlänge	PM
244	t_datin in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegt bereits ein Verbindungsabbau vor	normales Verhalten
251	t_datarq in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
252	t_datarq in Socketnetzprozess. Parameterfehler: ungültiger Wert für Datenlänge	PM
253	t_datarq in Socketnetzprozess. Parameterfehler: für die angegebene Transportreferenz liegt bereits ein Verbindungsabbau vor	normales Verhalten
261	t_disin in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
266	t_disrq in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
271	t_datastop in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
272	t_datastop in Socketnetzprozess. Parameterfehler: die angegebene Transportreferenz ist nicht mehr gültig	PM
276	t_datago in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
277	t_datago in Socketnetzprozess. Parameterfehler: die angegebene Transportreferenz ist nicht mehr gültig	PM
281	t_info in Socketnetzprozess. Parameterfehler: ungültiger Wert für Transportreferenz	PM
282	t_info in Socketnetzprozess. Parameterfehler: Option Pointer ist NULL	PM

&UERRNO	Bedeutung	Maßnahme
283	t_info in Socketnetzprozess. Parameterfehler: ungültige Optionstruktur	PM

- U307** &OBJ1 (pid: &PID, &TNSNAME): ungueltiges Ereignis &EVENT
- U308** &OBJ1 (pid: &PID, &TNSNAME): UTM-Anwendung &APPL hat sich beendet
- U309** &OBJ1 (pid: &PID, &TNSNAME): KCSTRMA mit Grund &TRMA aufgerufen (&STRTIME)
- U310** &OBJ1 (pid: &PID, &APPL): fork() Fehler errno: &ERRNO
- U311** &OBJ1 (pid: &PID, &APPL): Startfehler errno: &ERRNO
- U312** &OBJ1 (pid: &PID, &TNSNAME): Betriebsmittel-Engpass beim Senden
- U313** &OBJ1 (pid: &PID, &NETPROC): Warnung: &TNSNAME konnte nicht im TNS gefunden werden
- U315** &OBJ1 (pid: &PID): Netzmeldung: &TNSNAME &NETREAS

Das Insert &NETREAS hat dabei folgende Bedeutung:

&NETREAS	Bedeutung
1	Verbindungsaufbaubestätigung für eine bereits vom Partner abgebaute Verbindung
2	Zurückweisung des Verbindungsaufbauwunsches für eine bereits vom Partner abgebaute Verbindung
3	Verbindungsabbau für eine bereits vom Partner abgebaute Verbindung
6	Anwendungsende vom Netzmainprozess erkannt
7	Nachricht von openUTM für eine bereits abgebaute Verbindung erhalten
8	Verbindungsabbau von openUTM beim Anmelden erkannt
11	Ablehnung eines Verbindungsaufbauwunsches durch openUTM
12	Verbindungsabbau von openUTM empfangen
13	Verbindungsabbau vom Partner empfangen
14	Verbindungsabbau vom Partner beim Senden von Daten erkannt
15	Verbindungsabbau vom Partner beim Empfangen von Daten erkannt
20	aktiver Verbindungsaufbau
21	passiver Verbindungsaufbau

U316 &OBJ1 (pid: &PID, &NETPROC): TSEL-FORMAT(&NETNAME) Aenderung: &TFOLD nach &TFNEW.

&TFOLD entspricht dem bisherigen Wert von TSEL-FORMAT (Format der Transport-Selektoren), &TFNEW entspricht dem neuen Wert von TSEL-FORMAT.

Die Insert haben dabei folgende Bedeutung:

&TFOLD /&TFNEW	Bedeutung
?	undefiniertes TSEL-FORMAT(nicht bei &TFNEW)
T	TRANSDATA-Format
E	EBCDIC-Zeichenformat
A	ASCII-Zeichenformat



Die KDCDEF Generierung sollte überprüft werden.
Empfehlung: Definiertes Setzen des TSEL-FORMATS in den entsprechenden KDCDEF-Anweisungen.

U317 &OBJ1 (pid: &PID, &NETPROC): &TNSNAME CMX Fehler(&CMXERR) in &CMXFUNC.

Das Insert &CMXERR enthält die CMX-Fehlercodes (siehe CMX-Handbuch).

U318 &OBJ1 (pid: &PID, &NETPROC): Listen-port(&TNSNAME) : &LPOLD ist ungueltig.

Das Insert &LPOLD entspricht dem aktuellen Wert von LISTENER-PORT (Listener Portnummer), wobei nur die Portnummern 102, 1025 bis 32767 erlaubt sind.



Die KDCDEF Generierung sollte überprüft werden.
Empfehlung: Definiertes Setzen des LISTENER-PORTs in den entsprechenden KDCDEF-Anweisungen.

U319 &OBJ1 (pid: &PID, &NETPROC): IP Adresse (0.0.0.0) fuer Prozessname &PRNM ist ungueltig.

Für den Partner &NETPROC ist eine ungültige IP-Adresse gespeichert.



Die KDCDEF Generierung bzw. der Name Service (in Unix-/Windows-Systemen z.B. in der hosts-Datei) sollte überprüft werden.
Empfehlung: Nach der Überprüfung sollte mit Hilfe der dynamischen Administration oder WinAdmin/WebAdmin durch die Funktion KC_UPDATE_IPADDR die IP- Adresse neu angepasst werden.

U320 &OBJ1 (pid: &PID, &TNSNAME): &SOCKFCT-Aufruf: Fehler &ERRNO

Das Insert ERRNO entspricht dem Insert UERRNO der Meldung U306, siehe Tabelle bei U306.

U321 &OBJ1 (pid: &PID, &TNSNAME): Betriebsmittel-Engpass beim Anmelden

U322 &OBJ1 (pid: &PID, &TNSNAME): Keine Antwort vom Partner in Wartezeit erhalten

U323 &OBJ1 (pid: &PID): &BCAP Berechtigung fuer Port &PORT nicht vorhanden

5.8.10 Meldungen von kdckaa

U341 kdckaa: Anzahl der Argumente ungueltig

5.8.11 Meldungen des UTM-Tools kdcshut

U351 kdcshut: Anzahl der Argumente ungueltig

U352 kdcshut: ungueltiger Anwendungsname &PARAM10

U353 kdcshut: Fuer die Anwendung &APPL ist bereits ein kdcshut-Prozess vorhanden

U354 kdcshut: Anwendung &APPL ist nicht vorhanden in &OBJ3

U355 kdcshut: Fehler &UERRNO waehrend kdcshut-Ablauf

&UERRNO	Bedeutung
21	Parameter-Fehler beim Senden
22	Workprozess beim Senden nicht vorhanden, da Anwendungsende
23	Lock Fehler beim Senden
29	Unerwarteter Returncode beim Senden
31	Parameter-Fehler beim Abmelden
32	Kommunikations Shared Memory war beim Abmelden gesperrt

U356 kdcshut: Angegebene Zeit &PARAM10 ist nicht numerisch

U357 kdcshut: Angegebene Zeit &PARAM10 ist zu gross

U358 kdcshut: Ungueltiger Wert &ARG2 fuer shutdown Typ

5.8.12 Meldungen des UTM-Tools kdcrem

- U361** kdcrem: Anzahl der Argumente ungueltig
- U362** kdcrem: Anwendung &APPL in &OBJ3 nicht gefunden
- U363** kdcrem: normale Beendigung

5.8.13 Meldungen des UTM-Tools kdcprog

- U370** Benutzung:
kdcprog CREATE <filebase> <Anzahl FGG Eintraege>
INFO <filebase>
TRANSFER <filebase> [<fgg-nummer>]
SWITCH <filebase> <neue basis generation>
- U371** kdcprog: Ungueltiges Kommando &CMD
- U372** kdcprog: FILEBASE-name &FILEBASE ungueltig
- U373** kdcprog: Wert &GENUSL als Generationsnummer ungueltig
- U374** kdcprog: filebase Verzeichnis &FILEBASE nicht zugreifbar
- U375** kdcprog: Fehler &DMSE waehrend des Anlegens der FGG-Dateien fuer &FNAM
- U376** kdcprog: FGG-Dateien fuer &FNAM erzeugt.
- U377** kdcprog: Kann Information fuer &FNAM nicht anzeigen. Returncode ist &DMSE
- U378** INFO fuer FGG &FNAM
FGG Maximal Anzahl Versionen &GENUSL
FGG Basis &PRV1
FGG erste Generation &PRV2
FGG letzte Generation &PRV3
- U379** Datei PROG/&PRV1 ist PROG(&PRV3) &ARROW
- U380** Die folgenden Programmdateien sind verfuegbar:
- U381** kdcprog kann nicht nach &FNAM uebertragen. Format der FGG-Nummer ungueltig.
- U382** kdcprog kann nicht uebertragen. Returncode ist &DMSE
- U383** kdcprog: TRANSFER : &PRCMD
- U384** kdcprog: Transfer fehlgeschlagen; Returncode von cp ist &ERRNO
- U385** kdcprog: Transfer fehlgeschlagen; Returncode von chmod ist &ERRNO
- U386** kdcprog: &CMD nicht zulaessig, weil Application &APPL aktiv ist.

- U387** kdcprog kann SWITCH fuer &FNAM nicht durchfuehren. Returncode von KCSSWGG ist &DMSE
- U388** kdcprog: Neue Basis der Program FGG &FNAM ist &GENUSL
- U389** kdcprog: TRANSFER erfolgreich
- U390** kdcprog: SWITCH misslungen; Falsches Format der Basisnummer
- U391** kdcprog: TRANSFER fuer KDCAPPL PROG=NEW angestossen
- U392** kdcprog: Datei &FILEBASE nicht zugreifbar

5.9 Fehlercodes bei Dateibearbeitung (DMS-Fehler)

Bei der Dateibearbeitung und dem Aufruf anderer C-Laufzeitroutinen werden im Fehlerfall in den Meldungen Fehlercodes der Form yxxx ausgegeben. Im Zusammenhang mit der Dateibearbeitung werden diese auch DMS-Fehler genannt.

Folgende Inserts sind davon betroffen:

- &DMSE
- &ERRNO
- &UERRNO

Die DMS-Fehler haben folgende Bedeutung:

- y Das erste Zeichen y bezeichnet die Funktion, bei deren Ausführung der Fehler aufgetreten ist.
y kann folgende Werte annehmen:
- | | |
|---|---|
| A | Fehler beim Laden eines Shared Memories in den Adressraum |
| C | Fehler beim close-Aufruf |
| D | Fehler beim Abmelden von einem Shared Memory |
| E | Fehler beim remove-Aufruf |
| F | Fehler beim fstat/stat-Aufruf |
| G | Fehler beim Einrichten eines Shared Memories |
| L | Fehler beim lseek-Aufruf |
| M | Fehler beim mkdir-Aufruf |
| O | Fehler beim open-Aufruf |
| R | Fehler beim read-Aufruf |
| S | Fehler beim system-Aufruf |
| W | Fehler beim write-Aufruf |
| X | Fehler beim create-Aufruf |
- xxx Die drei Zeichen xxx stellen abdruckbar die Fehlernummer dar, die vom Betriebssystem in der externen Variablen 'errno' hinterlegt wird. Dabei wird die Fehlernummer, falls notwendig, mit führenden Nullen auf die Länge von drei Zeichen erweitert. Die Bedeutung der einzelnen Fehlernummern ist in den System-Handbüchern bei den entsprechenden Aufrufen und im Headerfile `errno.h` beschrieben.

Der Fehlercode O002 bedeutet beispielsweise, dass beim Öffnen einer Datei (O=open) die Datei nicht vorhanden war (2=errno ENOENT).

Zusätzlich gibt es noch folgende Fehlercodes:

CONS	Der Inhalt der Datei ist inkonsistent.
GPOS	GPOS bedeutet, dass mit <i>fgetpos()</i> nicht die Position im Stream der Startparameterdatei geholt werden konnte.
LERR	lseek konnte nicht auf die gewünschte Stelle positioniert werden.
OERR	Es wurde versucht, ein Dateiverzeichnis als normale Datei zu öffnen.
REND	Beim Lesen aus einer Datei wurde das Dateiende erreicht.
RERR	Es konnten nicht genügend Bytes gelesen werden.
WERR	Es konnten nicht genügend Bytes geschrieben werden.
LOCK	Die Datei kann nicht geschrieben werden, weil sie gesperrt ist.
MARK	Die erwarteten Datei-Markierungen konnten nicht gefunden werden. Vermutlich ist die Datei zerstört.
USED	Die Datei kann nicht geschrieben werden, weil sie gerade verwendet wird.
VERS	Die erwartete Versionsnummer konnte nicht gefunden werden. Eventuell ist die Datei zerstört.

5.10 Standard-Meldungsdefinitionsdatei, Inserts

5.10.1 Konstanten der Standard-Meldungsdefinitionsdatei

Konstantenname	Konstantenwert	Bemerkung
NL	0A	NEW LINE
NP	0C	NEW PAGE

NEW PAGE kann bei einigen PTYPEs nicht unterstützt werden. Es erscheint stattdessen ein Schmierzeichen.

5.10.2 Inserts in Meldungen

Die Spalte „Länge“ enthält die Ausgabelänge der einzelnen Inserts in Byte, d.h. das Insert belegt bei der Ausgabe des Meldungstextes soviele Zeichen, wie in der Spalte „Länge“ angegeben. Die Länge der Inserts ist insbesondere für die Erstellung von Meldungstexten mit KDCMMOD und KDCMTXT von Bedeutung.

Die Abkürzungen in der Spalte 'Daten-Typ' bedeuten dabei folgendes:

Char abdruckbare Zeichen

Int numerisches Feld

Hexa hexadezimale Information

5.10.2.1 Inserts in K- und P-Meldungen

Insert-Name	Datentyp	Länge	Bedeutung
AAID	Hexa	128	FIRST 64 BYTE OF ATOMIC ACTION IDENTIFIER
AAIS	Int	4	ATOMIC ACTION IDENTIFIER SIZE
ACPNT	Char	8	ACCESS-POINT-NAME
ACTION	Char	6	SYSTEM ACTION
ADTC	Char	8	ADMINISTRATION TAC
AGUS	Char	8	JOB-SUBMITTING USER
AMOD	Char	1	APPLICATION MODE
APPL	Char	8	APPLICATION NAME
ATAC1	Char	8	ASYNCHRONOUS TAC
ATAC2	Char	10	NUMBER OF UNPROCESSED ASYNCHRONOUS TACS
ATTR	Char	11	ATTRIBUT OF LOAD-MODULE/PROGRAM
ATYP	Char	1	APPLICATION TYPE (STANDALONE/CLUSTER)
BCAP	Char	8	APPLICATION NAME
BCMOPCD	Hexa	8	BCMM-OPCODE
BCMRTCD	Hexa	8	BCMM-RETURNCODE
BMD1	Char	8	BIT MODE OF SYSTEM
BMD2	Char	8	BIT MODE OF SYSTEM
CBRC	Hexa	8	RETURN CODE
CHAIN	Char	3	CHAINED MESSAGE INFORMATION
CID	Char	8	PRINTER CONTROL ID
CLSIGT	Int	2	CLUSTER COMMUNICATION SIGNAL TYPE
CMD	Char	8	COMMAND NAME
CNTR	Char	6	NUMBER OF LPUT RECORDS
CON	Char	8	CONNECTION NAME
COND	Char	3	CONDITION
CONU	Char	10	NUMBER OF CONNECTED USERS
COTM	Int	10	ELAPSED CONNECTION TIME IN SECONDS
CPTM	Int	10	CPU TIME SINCE SIGN-ON IN MILLISECONDS
CPUBEGIN	Hexa	8	CPU TIME AT TAC START IN MILLISECONDS
CPUCLNT	Hexa	8	CPU TIME USED OF THIS CLIENT
CPUEND	Hexa	8	CPU TIME AT TAC END IN MILLISECONDS
CPUREAS	Char	1	INTERNAL REASON

Insert-Name	Datentyp	Länge	Bedeutung
CPUTEXT	Char	8	TEXT: OVERFLOW OR NEGATIV
CPUUSED	Hexa	8	CPU TIME USED OF THIS TAC IN MILLISECONDS
CTYP	Char	4	TYPE OF PROGRAM EXCHANGE
DBCALL	Char	12	FUNCTION-CALL OF IUTMDB INTERFACE
DBCON	Char	8	DATABASE CONNECTION MODULE
DBV1	Char	8	VERSION OF DB CONNECTION MODULE
DBV2	Char	8	VERSION OF KDCDB MACRO
DEFVER	Char	5	VERSION NUMBER IN KAA
DEST	Char	8	DESTINATION OF ASYNCHRONOUS MSG
DEVC	Hexa	2	DEVICE TYPE
DIA1	Int	11	DIAGNOSTIC INFORMATION
DIA2	Int	11	DIAGNOSTIC INFORMATION
DIA3	Int	11	DIAGNOSTIC INFORMATION
DIA5	Char	80	INTERNAL DIAGNOSTIC INFORMATION
DLDATE	Char	3	DAY OF KDCS CALL PADM DL/DA
DLTIME	Char	8	TIME OF KDCS CALL PADM DL /DA
DMSE	Char	4	DMS ERROR CODE
DPID	Char	8	ASYNCHRONOUS MESSAGE ID
DTM2	Char	18	TIME STAMP 2
DTTM	Char	18	TIME STAMP
EBSR	Char	4	ACTUAL BRACKET STATE
EBSS	Char	4	SAVED BRACKET STATE
ENCPW	Hexa	16	ENCRYPTED PASSWORD
ERCD1	Hexa	4	ERROR CODE (IUTMHLL)
ERCD2	Hexa	4	INFO RETURN CODE (IUTMHLL)
ERCD3	Char	4	ERROR CODE (IUTMDB)
ERCD4	Char	4	ERROR CODE (IUTMFORM)
ERCD5	Char	4	INFO RETURN CODE (IUTMFORM)
ERCD6	Char	4	ERROR CODE (ROOT)
ERPRT	Char	1	PRINT ERROR CODE
ERRCODE	Char	16	RETURN CODE OF FAULTY FUNCTION
ERRNAME	Char	8	NAME OF FAULTY FUNCTION
ESQR	Hexa	10	ACTUAL REQUEST SEQUENCE NUMBER

Insert-Name	Datentyp	Länge	Bedeutung
ESQS	Hexa	8	SAVED SEQUENCE NUMBER
ESRR	Hexa	10	ACTUAL RESPONSE SEQUENCE NUMBER
EXIT	Char	10	CURRENT ACTIVE EXIT
FBASUPD	Char	42	FILE BASE NAME KDCUPD
FIL1A	Hexa	2	APPLICATION STATE
FIL1B	Hexa	2	BCAM REQUEST OR ANNO TYPE / UTM ANNO TYPE
FIL2A	Hexa	2	LTERM STATE
FIL2B	Hexa	8	DIAGNOSTIC WORD
FIL3	Hexa	4	PTERM STATE
FMH7	Char	80	ERROR RECOVERY PROCEDURE MESSAGE
FNAM	Char	54	FILE NAME
FNKT	Char	6	FUNCTION
FNOD	Char	1	FIRST NODE (Y/N) IN CLUSTER APPLICATION
FORM	Char	8	FORMAT NAME (FOR K015 ONLY)
GBLNBR	Int	11	NUMBER OF LOCKED GSSB
GLOBALSG	Char	1	CLUSTER GLOBAL SIGNON/SIGNOFF
GNDATE	Char	3	GENERATION DATE ASYNCHRONOUS MESSAGE
GNTIME	Char	8	GENERATION TIME ASYNCHRONOUS MESSAGE
GNUUSER	Char	8	USER NAME OF ASYNCHRON. MESSAGE GENERATION
GTRID	Hexa	128	FIRST 64 BYTE OF GLOBAL TRANSACTION ID
HITR	Char	3	CACHE HIT RATE
HST1	Char	8	HOST NAME
HST2	Char	8	HOST NAME
HST3	Char	8	HOST NAME
HSTACK	Int	2	HEIGHT OF STACK
IDFRFC	Hexa	16	RETURNCODE OF INVERSE KDCDEF
IDX1	Char	4	CLUSTER NODE INDEX
IDX2	Char	4	CLUSTER NODE INDEX
IDX3	Char	4	CLUSTER NODE INDEX
IMPVER	Char	5	VERSION NUMBER IN KAA OF KDCFILE TO IMPORT
IMSG1	Char	10	NUMBER OF TERMINAL INPUT MESSAGES
IMSG2	Hexa	64	FIRST PART OF INPUT MESSAGE
INF1	Char	65	ADDITIONAL INFORMATION

Insert-Name	Datentyp	Länge	Bedeutung
INF2	Char	65	ADDITIONAL INFORMATION
INSTNUM	Int	2	RM-INSTANCE NUMBER
INTTAID	Hexa	130	INTERNAL TRANSACTION ID
IOMS	Int	11	DURATION OF IO IN MILLISECONDS
IOPG	Int	11	NUMBER PAGES OF IO
IPADDR	Char	39	IPV4 (123.456.789.012) OR IPV6 (1234:5678:9ABC:DEF0:1234:5678:9ABC:DEF0) ADDRESS
LPAP	Char	8	LPAP NAME
LSES	Char	8	LSES NAME
LTAC	Char	8	TAC OR LTAC
LTACINDX	Hexa	8	LTAC INDEX
LTHGTRID	Int	2	LENGTH OF GLOBAL TRANSACTION ID
LTRM	Char	8	LTERM NAME
LWRT	Char	5	NUMBER OF USLOG FILE WRITES
MOD	Char	7	MODULE NAME
MSG	Char	80	MESSAGE TEXT
MSG2	Char	100	MESSAGE TEXT
MSTACK	Int	2	MAXIMUM STACK HEIGHT
MTYPE	Char	4	MESSAGE TYPE
MXLT	Char	8	MUX LTERM
MXP1	Char	4	MUX PROTOCOL VERSION (LOWER BOUNDARY)
MXP2	Char	4	MUX PROTOCOL VERSION (UPPER BOUNDARY)
MXPR	Char	8	MUX PROCESSOR
MXPT	Char	8	MUX PTERM
NCVST	Char	1	NEW CONVERSATION STATE
NMSG	Int	11	NUMBER OF MESSAGES
NNM1	Char	8	CLUSTER NODE NAME
NNM2	Char	8	CLUSTER NODE NAME
NNM3	Char	8	CLUSTER NODE NAME
NTAST	Char	1	NEW TRANSACTION STATE
NUMDAYS	Char	2	NUMBER DAYS PASSWORD VALID
NUMMSG	Int	11	NUMBER OF WAITING OUTPUT MESSAGES
OBJ1	Char	10	OBJECT NAME

Insert-Name	Datentyp	Länge	Bedeutung
OBJ2	Char	10	OBJECT NAME
OBJ3	Char	54	OBJECT OR FILENAME
OCVST	Char	1	OLD CONVERSATION STATE
OMSG1	Char	10	NUMBER OF TERMINAL OUTPUT MESSAGES
OMSG2	Char	74	BROADCAST MESSAGE
OPCD1	Char	4	OPCODE
OPCD2	Char	35	OPCODE (IUTMHLL)
OPCD3	Char	5	OPCODE (IUTMFORM)
OSLPAP	Char	8	OSI-LPAP NAME
OST1	Char	24	TYPE OF OPERATING SYSTEM
OST2	Char	24	TYPE OF OPERATING SYSTEM
OTAST	Char	1	OLD TRANSACTION STATE
PALTRM	Char	8	LTERM NAME PRINT ADMIN STATION
PAS1	Char	20	SPACE FOR PASSWORD
PAS2	Char	20	SPACE FOR PASSWORD
PAS3	Char	20	SPACE FOR PASSWORD
PGPOOL	Char	16	(NODE/CLUSTER) PAGEPOOL
PGS1	Int	11	NUMBER OF UTM PAGES
PGS2	Int	11	NUMBER OF UTM PAGES
PHAXAPTP	Char	14	INIT or START/RESTART of XAP-TP
PID	Int	11	UNIX/NT PROCESS ID
PRCN	Char	200	PROCEDURE/SCRIPT/COMMAND-FILE NAME
PRGVERS	Int	11	PROGRAM VERSION IN CASE OF PROGRAM EXCHANGE
PRNM	Char	8	PROCESSOR NAME
PROG	Char	32	PROGRAM OR LOAD MODULE NAME
PSQN	Hexa	8	SAVED PET SEQUENCE NUMBER
PTCID	Char	27	PTC IDENTIFICATION
PTRM	Char	8	PTERM NAME
PVER	Char	24	PROGRAM VERSION
RBCAUSER	Char	14	CAUSER OF ROLLBACK
RCCC	Char	3	KCRCCC
RCCC2	Char	4	STARTUP ERROR CODE
RCDC	Char	4	KCRCDC

Insert-Name	Datentyp	Länge	Bedeutung
RCF1A	Char	4	KCRCDC
RCF1B	Char	3	RETURN CODE 1
RCF1C	Char	4	RETURN CODE 1
RCF2A	Char	4	INTERNAL RETURN CODE
RCF2B	Char	4	RETURN CODE 2
RCHX	Hexa	8	RETURNCODE IN HEX-FORM
RCVDANNO	Hexa	8	FIRST 4 BYTES OF RECEIVED ANNO
RCXAPTP	Int	3	RETURNCODE XAP-TP STARTFUNCTIONS
REA1	Hexa	2	REASON
REA2	Char	2	REASON
REA3	Char	136	ERROR MESSAGE
REA4	Char	1	DIAGNOSTIC INFORMATION
REA6	Hexa	2	DIAGNOSTIC INFORMATION (DISCONNECT USER REASON)
REA7	Int	2	DIAGNOSTIC INFORMATION (REJECT USER REASON)
REST	Char	1	RESTART INDICATOR OF LTERM
RMSTAT	Char	8	CONNECTION STATUS OF AN RM
RQM	Int	11	REQUESTED NUMBER OF BYTES
RSES	Char	8	RSES NAME
RSLT	Char	1	RESULT
RSPTC	Char	1	RESET-PTC (Y/N) FOR NODE RECOVERY
RTAANZ	Int	2	NUMBER OF RECOVERED TRANSACTIONS
RTCD	Hexa	8	RETURN CODE
SATRC	Hexa	8	SAT RETURNCODE
SESSCNTR	Int	3	SESSION COUNTER OF ACTUAL SERVICE
SRFG	Hexa	8	SAVED SESSION STATE
STA2	Char	1	STATE
STATE	Char	1	STATE
STDHEAD	Hexa	16	BS2000 STANDARDHEADER
STMT	Char	11	STATEMENT OF KDCDEF
STRTPAR	Char	8	STARTPAR LTH = 8
STS1	Hexa	4	STSN-REQ SEQUENCE NUMBER RCV-CNT
STS2	Hexa	4	STSN-REQ SEQUENCE NUMBER SEND-CNT

Insert-Name	Datentyp	Länge	Bedeutung
STS3	Hexa	4	STSN-RSP SEQUENCE NUMBER SLU-PLU
STS4	Hexa	4	STSN-RSP SEQUENCE NUMBER PLU-SLU
STSK	Char	1	SYSTEM TASK (Y/N)
SUFF	Char	5	FILE SUFFIX
SWNR	Int	11	NUMBER OF JOURNAL SWITCHES
SYN	Char	50	SYNTAX ERROR
SYSD	Hexa	4	SYSTEM SENSE DATA
SYST	Char	4	SYSTEM
TAC	Char	8	TRANSACTION CODE
TACINDX	Hexa	8	TAC INDEX
TACNTR	Int	5	TA CNTR OF ACTUAL SERVICE
TACTYPE	Char	1	TAC TYPE
TASK	Char	4	PID (MAX. 4 DIGITS) OF UTM PROCESS
TCPCL	Char	18	SOCKET FUNCTION
TCPMS	Int	11	DURATION OF SOCKET FUNCTION IN MILLISECONDS
TCPRC	Hexa	8	SOCKET DIAGNOSTIC WORD
TCVG	Char	8	CONVERSATION TAC
TERM	Char	1	TERMINATION TYPE
TEXT32	Char	32	STANDARD-TEXTPUFFER
TPRIO	Int	3	EXTERNAL TASK-PRIORITY
TRMA	Char	6	TERM APPLICATION REASON
TSNPID	Char	10	TSN (BS2000) / FULL RECESSION PID (UNIX/WIN)
UERCODE	Char	8	ERROR CODE
UERINFO	Char	8	ERROR INFORMATION
UKCHSTA	Int	5	HEIGHT OF STACK
UKCLM	Int	10	LENGTH OF KCLM USED BY KDCUPD
UKCOP	Char	4	OPCODE OF KDCS CALL USED BY KDCUPD
UKCRN	Char	8	REFERENCE NAME USED BY KDCUPD
ULLNBR	Int	11	NUMBER OF LOCKED ULS
UPCPROT	Hexa	8	UPIC PROTOCOLL
UPCREAS	Hexa	2	UPIC ERROR REASON
UPCSTAT	Hexa	4	USRTNSR UPIC STATE
UPDERR	Char	5	UPD ERROR CODE

Insert-Name	Datentyp	Länge	Bedeutung
UPDMODUL	Char	8	UPD MODULE READxxxx/WRITxxxx
UPDPRO	Int	3	PERCENT USED PAGES IN NEW FILE
UPDTYP	Char	6	TYPE OF KCRN FOR UPDATE (LTERM, TAC, LPAP)
UPPENC2	Hexa	4	UPIC ENCRYPTION PTRMDYN INFO
UPVENC1	Hexa	4	UPIC ENCRYPTION VGTDYN INFO
USER	Char	8	USER/LSES/OSI-ASS NAME
USRTYPE	Char	10	USER-TYP: CLIENT/CONNECTION
USSD	Hexa	4	USER SENSE DATA
USTYPPTC	Char	1	TYPE OF USER IN PTC
UTMDEVT	Char	7	UTM-D EVENT
VER1	Char	6	VERSION NUMBER
VER2	Char	6	VERSION NUMBER
VERS	Char	8	UTM VERSION
VGCNTR	Int	11	SERVICE COUNTER OF ACTUAL SERVICE
VTRC	Hexa	8	RETURN CODE
WLEV	Char	1	WARN LEVEL OF PAGE POOL
WTBF	Char	3	CACHE WAITS FOR BUFFER
XACALL	Char	12	FUNCTION-CALL OF XA-CAE INTERFACE
XADBC1	Char	8	TEXT FOR DB-XA-CALLS
XADBC2	Char	8	TEXT FOR DB-XA-CALLS
XAFLAG	Char	8	FLAGS FOR XA-CALLS
XASPEC	Char	12	VERSION OF XA-SPECIFICATION
XATXT	Char	16	READABLE XA-RETURNCODE
XP0OBID	Int	11	OSI-TP OBJECT IDENTIFIER 0
XP1BOOL	Char	5	OSI-TP CCR V2 NOT AVAILABLE
XP1DIA	Int	11	OSI-TP DIAGNOSTIC INFORMATION 1
XP1INFO	Int	11	OSI-TP ADDITIONAL INFORMATION 1
XP1OBID	Int	11	OSI-TP OBJECT IDENTIFIER 1
XP2BOOL	Char	5	OSI-TP PROTOCOL VERSION INCOMPATIBILITY
XP2DIA	Int	11	OSI-TP DIAGNOSTIC INFORMATION 2
XP2INFO	Int	11	OSI-TP ADDITIONAL INFORMATION 2
XP2OBID	Int	11	OSI-TP OBJECT IDENTIFIER 2
XP3BOOL	Char	5	OSI-TP CONTENTION WINNER ASSIGNMENT REJECTED

Insert-Name	Datentyp	Länge	Bedeutung
XP3DIA	Int	11	OSI-TP DIAGNOSTIC INFORMATION 3
XP3INFO	Char	40	OSI-TP ADDITIONAL INFORMATION 3
XP3OBID	Int	11	OSI-TP OBJECT IDENTIFIER 3
XP4BOOL	Char	5	OSI-TP BID MANDATORY REJECTED
XP4OBID	Int	11	OSI-TP OBJECT IDENTIFIER 4
XP5BOOL	Char	5	OSI-TP NO REASON GIVEN
XP5OBID	Int	11	OSI-TP OBJECT IDENTIFIER 5
XP6OBID	Int	11	OSI-TP OBJECT IDENTIFIER 6
XP7OBID	Int	11	OSI-TP OBJECT IDENTIFIER 7
XP8OBID	Int	11	OSI-TP OBJECT IDENTIFIER 8
XP9OBID	Int	11	OSI-TP OBJECT IDENTIFIER 9
XPAPDU	Char	20	OSI-TP APDU TYPE
XPASST	Char	20	ASSOCIATION STATE
XPCCLS	Int	11	CMX ERROR CLASS
XPCORR	Int	11	MESSAGE CORRELATOR NUMBER
XPCPSEL	Char	16	OSI-TP P-SEL OF PARTNER (CHAR)
XPCRES	Int	4	OSI-TP NEGATIVE CONFIRMATION RESULT
XPCSSEL	Char	16	OSI-TP S-SEL OF PARTNER (CHAR)
XPCTYPE	Int	11	CMX ERROR TYPE
XPCVAL	Int	11	CMX ERROR VALUE
XPERR	Int	11	OSI-TP ERROR CODE
XPEVT	Char	10	XAPTP EVENT
XPFSMN	Char	10	OSI-TP FSM NAME
XPFUNC	Char	20	CALLED OSI-TP FUNCTION
XPHPSEL	Hexa	32	OSI-TP P-SEL OF PARTNER (HEX)
XPHSSEL	Hexa	32	OSI-TP S-SEL OF PARTNER (HEX)
XPINI	Int	11	OSI-TP INITIATOR
XPLNK	Int	11	OSI-TP LINK
XPLPSEL	Int	2	OSI-TP LENGTH P-SEL OF PARTNER
XPLSSEL	Int	2	OSI-TP LENGTH S-SEL OF PARTNER
XPLTH	Int	11	OSI-TP INVALID LENGTH
XPNDIA	Int	4	OSI-TP NEGATIVE DIAGNOSTICS
XPNSEL	Char	8	OSI-TP N-SEL OF PARTNER

Insert-Name	Datentyp	Länge	Bedeutung
XPOSAS	Int	8	OSI-TP ASSOCIATION REFERENCE
XPPDU	Int	11	OSI-TP PDU TYPE
XPPTYP	Int	11	OSI-TP PRIMITIVE TYPE
XPRET	Int	11	OSI-TP RETURN CODE
XPRJCT	Int	4	OSI-TP ASSOCIATION REASON FOR REJECT
XPSRC	Int	4	OSI-TP RESULT SOURCE FROM PARTNER
XPTRFAIL	Int	11	OSI-TP WRITE TRACE FAILURE REASON
XPTSEL	Char	8	OSI-TP T-SEL OF PARTNER

5.10.2.2 Inserts in U-Meldungen

Insert-Name	Datentyp	Länge	Bedeutung
APPL	Char	8	APPLICATION NAME
ARG2	Char	2	ARGUMENT
ARROW	Char	2	POINTS TO CURRENT ENTRY IN FGG
BCAP	Char	8	APPLICATION NAME
CMD	Char	8	COMMAND NAME
CMXERR	Int	4	CMX ERROR
CMXFUNC	Char	20	CMX FUNCTION
DIRECT	Char	29	DIRECTORY NAME
DMSE	Char	4	DMS ERROR CODE
ERRNO	Int	4	UNIX/NT ERROR NUMBER
EVENT	Int	4	CMX OR NEABX EVENT
EXITC	Int	4	EXIT CODE OF SCRIPT UTMLP
EXTCODE	Int	4	EXIT CODE OF TERMINATED PROCESS
FBTYPE	Char	20	FILEBASENAME
FERRNO	Int	4	FORMATSYSTEM ERROR NUMBER
FILEBAS1	Char	80	FILEBASENAME
FILEBAS2	Char	80	FILEBASENAME
FILEBASE	Char	40	NAME OF FILEBASE
FMTN	Char	8	FORMAT NAME
FNAM	Char	54	FILE NAME
GENUSL	Int	11	NUMBER OF USLOG GENERATIONS
IPCOBJ	Char	4	IPC OBJECT TYPE

Insert-Name	Datentyp	Länge	Bedeutung
IPCREAS	Char	10	IPC ERROR REASON
LOGREC	Hexa	12	UNIX/NT UTM LOGGING RECORD
LPOLD	Int	5	OLD LISTEN(ER)-PORT
NBRBYTES	Int	11	UNIX/NT NUMBER BYTES
NETERR	Hexa	8	ERROR VALUE RETURNED BY T_ERROR()
NETFCT	Char	10	CMX FUNCTION NAME
NETNAME	Char	17	SHORT NET CONNECTION NAME
NETPROC	Char	17	NET PROCESS NAME
NETREAS	Int	4	NET MESSAGE REASON
NLSLANG	Char	14	NLS LANGUAGE VARIABLE
OBJ1	Char	10	OBJECT NAME
OBJ2	Char	10	OBJECT NAME
OBJ3	Char	54	OBJECT OR FILENAME
PARAM10	Char	10	PARAMETER LTH=10
PID	Int	11	UNIX/NT PROCESS ID
PNAME	Char	8	PARTNER NAME
PORT	Int	5	PORT NUMBER
PRCMD	Char	100	SYSTEM CMD FOR COPYING
PRNM	Char	8	PROCESSOR NAME
PRTYPE	Char	8	PHYSICAL PRINTER TYPE OR PROCESS TYPE
PRV1	Char	4	INFO ABOUT FGG
PRV2	Char	4	INFO ABOUT FGG
PRV3	Char	5	INFO ABOUT FGG
PTRM	Char	8	PTERM NAME
RETVALUE	Int	11	UNIX/NT RETURNED VALUE BY SYSTEM CALLS
RNAME	Char	8	REFERENCE NAME
SCALL	Char	40	SYSTEM CALL
SEMKEY	Int	11	SEMAPHOR KEY
SHMKEY	Int	11	SHARED MEMORY KEY
SHMLTH	Int	11	SIZE OF SHM
SIGEXIT	Char	10	EXIT CODE OF TERMINATED PROCESS
SOCKFCT	Char	10	SOCKET FUNCTION NAME
STIND	Char	6	UTMWORK START INDICATOR

Insert-Name	Datentyp	Länge	Bedeutung
STRTIME	Char	30	DATE AND TIME AS CHAR ARRAY
TFNEW	Char	1	TRANSPORT SELECTOR NEW
TFOLD	Char	1	TRANSPORT SELECTOR OLD
TNSCLASS	Int	4	ERRCLASS IN TNS STANDARD HEADER
TNSCODE	Int	5	RETCODE IN TNS STANDARD HEADER
TNSNAME	Char	83	NET CONNECTION NAME
TNSPROP	Char	8	TNS PROPERTY NAME
TNSVALUE	Int	4	ERRVALUE IN TNS STANDARD HEADER
TPROT	Char	8	TRANSPORT PROTOCOL TYPE
TPROT1	Char	8	TRANSPORT PROTOCOL TYPE
TRMA	Char	6	TERM APPLICATION REASON
TSEL	Char	1	TRANSPORT SELECTOR FORMAT
UABNREAS	Char	20	ABNORMAL TERMINATION REASON
UDIA1	Int	11	DIAGNOSTIC INFORMATION
UDIA2	Int	11	DIAGNOSTIC INFORMATION
UERRNO	Int	4	UTM ERROR NUMBER
UREAS	Char	200	DETAILED ERROR REASON
VERS	Char	8	UTM VERSION
WID	Hexa	8	WORK PROCESS ID
WTYP	Char	10	WORK APPLICATION TYPE

5.11 Ziele der UTM-Meldungen

Die folgende Tabelle zeigt für jede Meldung, welche Angaben möglich sind. Es sind nur die Meldungen berücksichtigt, die vom Anwender verändert werden können.

Die Angaben in den Spalten der Tabelle bedeuten auf die einzelnen Meldungen bezogen:

- R (Required)
Das Meldungsziel ist der Meldung fest zugeordnet; es kann nicht verändert werden.
- + Das Meldungsziel ist für diese Meldung erlaubt.
- D (Default)
Das Meldungsziel ist für diese Meldung angegeben, es kann ihr entzogen werden.
- Das Meldungsziel ist für diese Meldung nicht erlaubt.

in der Spalte COMPRESS:

- Y (Yes)
Überflüssige Leerzeichen werden aus der Meldung entfernt.
- N (No)
Überflüssige Leerzeichen verbleiben in der Meldung.

Damit Sie Fehler, die beim Start einer UTM-Anwendung oder eines Folgeprozesses auftreten, leichter diagnostizieren können, werden alle K-Meldungen von openUTM, die in der Startphase auftreten, auf *stderr* und *stdout* ausgegeben, unabhängig davon, welche Meldungsziele für diese Meldungen festgelegt wurden.

Unter Windows-Systemen werden Meldungen mit dem Meldungsziel CONSOLE in die Datei CONSOLE.TXT im Verzeichnis *filebase* geschrieben.

Die Funktionseinheit ist für den Anwender modifizierbar.

Msg-Id	Inserts	S t a t i o n	S Y S T E M	P a r t n e r	S Y S T E M L O G	M S G T A C	S Y S T E M O U T	S Y S T E M I N P U T	C o n s o l e	U S E R D E S T	C o m p r e s s
K001 ¹	PTRM , PRNM , BCAP , LTRM , APPL	R	-	+	+	+	+	+	+	-	Y
K002 ¹	PTRM , PRNM , BCAP , LTRM , APPL	R	-	-	+	+	+	+	+	+	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K003	PTRM , PRNM , BCAP , LTRM , CMD	R	D	-	+	+	+	+	+	+	N
K004 ¹	PTRM , PRNM , BCAP , LTRM , USER , REA7	R	-	-	+	+	+	+	+	+	N
K005 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	N
K006 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	N
K007 ¹	PTRM , PRNM , BCAP , LTRM , USER , REA7	R	-	-	+	+	+	+	+	+	N
K008 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	-	Y
K009	PTRM , PRNM , BCAP , LTRM , USER , TAC , RCDC	R	D	D	+	+	+	+	+	+	N
K010	PTRM , PRNM , BCAP , LTRM , USER , TAC	R	D	D	+	+	+	+	+	+	N
K011	PTRM , PRNM , BCAP , LTRM , USER , ATAC1	R	+	+	+	+	+	+	+	+	N
K012	NUMMSGs	R	R	-	-	-	-	-	-	-	N
K013	PTRM , PRNM , BCAP , LTRM , CMD	R	R	-	+	+	+	+	+	+	N
K016	PTRM , PRNM , BCAP , LTRM , USER	R	D	+	+	+	+	+	+	+	N
K017	PTRM , PRNM , BCAP , LTRM , USER , TCVG , RCCC , RCDC , TAC	R	-	D	D	+	+	+	+	+	Y
K018	PTRM , PRNM , BCAP , LTRM , APPL	R	-	-	+	+	+	+	+	+	N
K019	PTRM , PRNM , BCAP , LTRM , APPL	R	R	-	+	+	+	+	+	+	N
K020	PTRM , PRNM , BCAP , LTRM , USER	R	R	-	+	+	+	+	+	+	N

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K021	PTRM , PRNM , BCAP , LTRM	-	-	D	+	+	+	+	+	+	N
K022	PTRM , PRNM , BCAP , LTRM	R	-	+	+	+	+	+	+	+	Y
K023	OMSG2	-	-	+	-	-	-	-	-	-	N
K024	PTRM , PRNM , BCAP , LTRM , USER	R	R	D	+	+	+	+	+	+	N
K025	PTRM , PRNM , BCAP , LTRM	R	-	D	D	+	+	+	+	+	Y
K026	PTRM , PRNM , BCAP , LTRM , USER	R	+	+	+	+	+	+	+	+	N
K027	PTRM , PRNM , BCAP , LTRM	R	+	D	+	+	+	+	+	+	N
K028	PTRM , PRNM , BCAP , LTRM , USER , PAS1	R	-	-	+	+	+	+	+	-	N
K029	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	-	N
K030 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	N
K031 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	N
K032	CON , PRNM , BCAP , LPAP , USER , RCF1B , RCF2B	-	-	-	D	+	D	+	+	+	N
K033	PTRM , PRNM , BCAP , LTRM , USER , REST , GLOBALSG	-	-	-	D	+	+	+	+	+	N
K034		R	R	+	-	-	-	-	-	-	N
K035		R	R	+	-	-	-	-	-	-	N
K036	PTRM , PRNM , BCAP , LTRM , RSLT , REA1	-	-	-	D	+	+	+	+	+	N
K038	SYN	-	-	-	-	-	R	+	-	-	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K039	STRTPAR	-	-	-	-	-	R	+	-	-	Y
K040	WLEV , PGPOOL	-	-	-	D	+	R	+	+	+	Y
K041	WLEV , PGPOOL	-	-	-	D	+	R	+	+	+	Y
K042		-	-	-	-	-	R	+	-	-	Y
K043	DMSE , FNAM	-	-	-	D	+	R	D	+	+	Y
K044		-	-	-	-	-	R	+	-	-	Y
K045	PTRM , PRNM , BCAP , LTRM , PALTRM , CID	-	-	-	-	+	-	-	-	+	N
K046	PTRM , PRNM , BCAP , LTRM , PALTRM , CID , DPID , ERPRT , IMSG2	-	-	-	+	+	+	+	+	+	N
K047	OPCD1 , RTCD	-	-	-	-	-	R	+	-	-	Y
K049	RCCC2	-	-	-	-	-	D	D	-	-	Y
K050	APPL , VERS , AMOD , TERM , OST1 , BMD1 , ATYP , FNOD	-	-	-	D	+	R	+	+	+	Y
K051	APPL , VERS , AMOD , TERM , OST1 , BMD1 , ATYP , FNOD	-	-	-	D	+	R	+	+	+	Y
K052	TASK , APPL , PID , PRGVERS , STSK	-	-	-	D	+	R	+	+	+	Y
K053	CNTR	-	-	-	D	+	R	+	+	+	Y
K054		-	-	-	D	+	R	+	+	+	Y
K055	ATAC1 , RCCC , RCDC , USER , LTRM	-	-	-	D	+	R	+	+	+	Y
K056	TASK , PID , RSLT	-	-	-	D	+	D	+	+	+	Y
K057		-	-	-	D	-	D	+	+	-	Y
K058	TASK , PID	-	-	-	+	+	D	+	+	+	Y
K059		-	-	-	+	-	D	+	+	-	Y
K060	TRMA	-	-	-	D	-	R	D	+	-	Y
K061	FNAM	-	-	-	D	-	R	D	+	-	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K062		-	-	-	D	-	R	D	+	-	Y
K064	PTRM , PRNM , BCAP , LTRM , DEVC , FIL1A , FIL2A , FIL3 , VTRC , IMSG2 , REA1 , CBRC	-	-	-	D	+	+	+	+	+	Y
K065	PTRM , PRNM , BCAP , LTRM , FIL1B , FIL2B	-	-	-	D	+	+	+	+	+	N
K066		-	-	-	-	-	R	+	-	-	Y
K067	MOD , ERCD1 , ERCD2 , OPCD2	-	-	-	-	-	R	+	-	-	Y
K068	DBCON , DBV1 , DBV2	-	-	-	-	-	R	+	-	-	Y
K069	PTRM , PRNM , BCAP , LTRM , COTM , REA4 , REA6	-	-	-	D	+	+	+	+	+	N
K070	PTRM , PRNM , BCAP , LTRM , USER , COTM , CPTM , GLOBALSG	-	-	-	D	+	+	+	+	+	N
K071	OPCD1 , ERCD3	-	-	-	-	-	R	+	-	-	Y
K072	STMT	-	-	-	-	-	D	D	-	-	Y
K073	ATTR , STMT , PROG	-	-	-	-	-	D	D	-	-	Y
K074	CTYP , PROG , PVER	-	-	-	D	+	R	+	+	+	Y
K075	CTYP , PROG , PVER , TASK , PID	-	-	-	D	+	D	+	+	+	Y
K076	RCCC , RCDC , ADTC , USER , LTRM	-	-	-	D	+	R	+	+	-	Y
K077	PTRM , PRNM , BCAP , LTRM , CLSIGT	-	-	-	D	-	D	D	-	-	Y
K078	ERRNAME , ERRCODE , REA3	-	-	-	-	-	R	D	-	-	Y
K079	REA2	-	-	-	D	+	+	+	+	+	Y
K080		-	-	-	D	+	+	+	+	+	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K081	IMSG1, OMSG1, CONU, ATAC2, LWRT, HITR, WTBF	-	-	-	D	+	+	+	+	+	N
K082	FNAM	-	-	-	-	-	R	D	-	-	Y
K083	FNAM	-	-	-	-	-	R	D	-	-	Y
K084	OBJ1, VER1, OBJ2, VER2, OST1, BMD1, OST2, BMD2	-	-	-	-	-	R	D	-	-	Y
K085	FNKT	-	-	-	-	-	R	D	-	-	Y
K086	PTRM, PRNM, BCAP, LTRM, USER, SYSD, USSD, FMH7, AGUS	-	-	-	D	+	+	+	+	+	N
K088	LSES, RSES, LPAP, SRFG, PSQN, ESQS, EBSS, ESQR, ESRR, EBSR	-	-	-	D	+	+	+	+	+	N
K089	GNDATE, GNTIME, DEST, GNUSER, USER, DLDATE, DLTIME, CHAIN	-	-	-	D	+	+	+	+	+	N
K090	DEST, USER, DLDATE, DLTIME	-	-	-	D	+	+	+	+	+	N
K091	PTRM, PRNM, BCAP, LTRM, USER, REA7	R	+	+	+	+	+	+	+	+	N
K092	PTRM, PRNM, BCAP, LTRM, USER, PAS1, PAS2, PAS3	R	-	-	+	+	+	+	+	-	N
K093	PTRM, PRNM, BCAP, LTRM, USER, HSTACK, MSTACK	R	R	+	+	+	+	+	+	+	Y
K094	PTRM, PRNM, BCAP, LTRM, USER, RCF1B, REA4	-	-	-	D	+	+	+	+	+	N
K095		D	R	-	-	-	-	-	-	-	N

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K096		R	R	-	-	-	-	-	-	-	N
K097 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	Y
K098	PTRM , PRNM , BCAP , LTRM , USER , RCF1C , RCF2B	R	R	-	+	+	+	+	+	+	N
K099	MSG	-	-	-	-	-	R	+	-	-	Y
K101	PTRM , PRNM , BCAP , LTRM , USER , REA1 , DEST	R	R	+	D	+	D	+	+	+	Y
K104	UTMDEVT , LSES , LPAP , AGUS , OCVST , OTAST , ACTION , NCVST , NTAST , SESSCNTR, VGCNTR , TACNTR , RCVDANNO , TACINDX , LTACINDX	-	-	-	D	+	+	+	+	+	N
K105	LSES , LPAP , AGUS , SYST	-	-	-	D	+	+	+	+	+	N
K119	OSLPAP , USER , TAC , DIA1 , DIA2 , DIA3 , SESSCNTR , VGCNTR , TACNTR , AAIS , AAID	-	-	-	D	+	+	+	+	+	Y
K120 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	D	+	+	+	+	+	N
K121	PTRM , PRNM , BCAP , LTRM , USER , PAS1 , PAS2 , PAS3 , NUMDAYS	R	-	-	+	+	+	+	+	-	N
K122	PTRM , PRNM , BCAP , LTRM , USER , NUMDAYS	+	R	-	D	+	+	+	+	-	Y
K123 ¹	LTRM , TAC , USER	R	+	-	D	+	+	+	+	+	N
K124	RCXAPTP , PHAXAPTP	-	-	-	D	-	D	D	+	-	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K125 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	D	+	+	+	+	+	Y
K126	SATRC	-	-	-	D	+	D	D	+	+	N
K127	ERCDC6	-	-	-	-	-	R	+	-	-	Y
K128	CON , PRNM , BCAP , LPAP , LSES , REA1 , RCDC , TAC	-	-	-	D	+	+	+	+	+	N
K129	CMD	-	-	-	-	-	R	+	-	-	Y
K130	TPRIO , TASK	-	-	-	D	+	+	+	+	+	Y
K132	OBJ1 , OBJ2	-	-	-	-	-	R	+	-	-	Y
K133	EXIT	-	-	-	-	-	R	+	-	-	Y
K134	DEST , NMSG	-	-	-	D	+	+	+	+	+	Y
K135	PTRM , PRNM , BCAP , LTRM , UPCREAS , UPCSTAT , UPCPROT , UPVENC1 , UPPENC2	-	-	-	D	+	+	+	+	+	N
K136	FNAM	-	-	-	-	-	D	+	+	+	Y
K137	FNAM	-	-	-	+	+	D	+	+	+	Y
K138	FNAM	-	-	-	-	+	D	+	+	+	Y
K139	FNAM	-	-	-	+	+	D	+	D	+	Y
K140	PTRM , PRNM , BCAP , LTRM , MXP1 , MXP2	-	-	-	D	+	+	+	+	+	Y
K141	PTRM , PRNM , BCAP , LTRM , MXP1	-	-	-	D	+	+	+	+	+	Y
K142	PTRM , PRNM , BCAP , LTRM , MXPT , MXPR , MXLT	-	-	-	D	+	+	+	+	+	Y
K143	PTRM , PRNM , BCAP , LTRM , STS1 , STS2 , STS3 , STS4	-	-	-	D	+	+	+	+	+	Y
K145 ¹	PTRM , PRNM , BCAP , LTRM , USER	R	-	+	+	+	+	+	+	+	N

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K146	BCMOPCD , BCMRTCD , STDHEAD , TASK , BCAP	-	-	-	D	+	+	+	+	+	Y
K147	PTRM , PRNM , BCAP , LTRM , USER , REA7 , USRTYPE	-	-	-	D	+	D	+	+	+	Y
K149	DIA5	-	-	-	-	-	R	+	+	-	Y
K151	IDEFRC , DMSE , FNAM	-	-	-	D	+	D	D	+	+	Y
K152	COND , MTYPE , OSLPAP , USER , LTAC , AAIS , AAID , SESSCNTR, VGCNTR , TACNTR	-	-	-	D	+	+	+	+	+	Y
K154	PTRM , PRNM , BCAP , LTRM , TCPCL , TCPRC	-	-	-	D	+	D	+	+	+	N
K155	PTRM , PRNM , BCAP , LTRM , USER , PAS1 , PAS2 , PAS3	R	-	-	+	+	+	+	+	-	N
K156	RSLT	-	-	-	D	+	+	R	+	+	N
K158	PTRM , PRNM , BCAP , LTRM , USER , CPUTEXT , CPUBEGIN, CPUEND , CPUUSED , CPUCLNT , CPUREAS	-	-	-	D	+	+	+	+	+	Y
K159	USER , ENCPW	-	-	-	+	+	+	+	+	-	Y
K160	PTRM , PRNM , BCAP , LTRM , USER , TCVG , TAC , TACNTR , RBCAUSER , RCCC , RCDC , TASK , SESSCNTR, VGCNTR , LTHGTRID, GTRID , PID	-	-	-	D	+	+	+	+	+	Y
K161	TASK , PID	-	-	-	+	+	+	+	+	-	N
K162	TASK , IOPG , IOMS , PID	-	-	-	D	-	R	D	+	-	Y
K163	TASK , IOPG , IOMS , PID	-	-	-	D	+	R	D	+	-	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K169	ACTION , IDX1 , HST1 , STATE , IDX2 , HST2 , IDX3 , HST3 , NNM1 , NNM2 , NNM3	-	-	-	R	-	R	D	+	-	Y
K170	DTTM , STATE	-	-	-	R	-	R	D	+	-	Y
K174	DIA1	-	-	-	R	-	R	D	+	-	Y
K175	FNAM	-	-	-	R	-	R	D	+	-	Y
K176	PRCN , MSG2 , RCHX	-	-	-	R	+	R	D	+	+	Y
K178	ACTION , STATE , STA2 , PGS1 , PGS2 , SWNR	-	-	-	R	-	R	D	+	-	Y
K179		-	-	-	R	+	D	D	+	+	Y
K180		-	-	-	R	+	D	D	+	+	Y
K181	FNAM	-	-	-	R	+	D	D	+	+	Y
K182	FNAM	-	-	-	R	+	D	D	+	+	Y
K183	FNAM	-	-	-	R	+	D	D	+	+	Y
K184		-	-	-	R	+	D	D	+	+	Y
K185		-	-	-	R	+	D	D	+	+	Y
K186		-	-	-	R	+	D	D	+	+	Y
K187	FNAM	-	-	-	R	+	D	D	+	+	Y
K188	OBJ1 , DTTM , OBJ2 , DTM2	-	-	-	R	-	R	D	+	-	Y
K189	PTRM , PRNM , BCAP , LTRM , USER , HST1	R	+	-	D	+	+	+	+	+	Y
K190	DIA1 , INF1 , INF2 , SUFF	-	-	-	R	-	R	D	+	-	Y
K191	SUFF , RQM	-	-	-	D	-	D	D	+	-	Y
K192	NNM1 , HST1 , RSPTC	-	-	-	-	-	D	D	-	-	Y
K193	PTCID , USER , LPAP , LSES , USTYPPTC	-	-	-	D	-	D	D	+	-	Y
K194	GBLNBR , ULLNBR	-	-	-	D	-	D	D	+	-	Y
K195	MSG2	+	+	+	+	+	+	+	+	+	N

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K196	MSG2	+	+	+	+	+	+	+	+	+	N
K197	MSG2	+	+	+	+	+	+	+	+	+	N
K198	MSG2	+	+	+	+	+	+	+	+	+	N
K199	MSG2	+	+	+	+	+	+	+	+	+	N
K200	MSG2	-	-	-	+	+	+	+	+	+	N
K201	TSNPID , XASPEC	-	-	-	-	-	R	D	-	-	Y
K202	TSNPID , INSTNUM , TEXT32 , RMSTAT	-	-	-	-	-	R	D	-	-	Y
K203	TSNPID , TEXT32 , INSTNUM , RTAANZ	-	-	-	-	-	R	D	-	-	Y
K204	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y
K205	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y
K206	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y
K207	TSNPID , INTTAID , INSTNUM	-	-	-	-	-	R	D	-	-	Y
K210	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y
K211	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y
K212	TSNPID , XATXT , XAFLAG , INTTAID	-	-	-	-	-	R	D	-	-	Y
K213	TSNPID , XATXT , XAFLAG , INTTAID	-	-	-	-	-	R	D	-	-	Y
K214	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y
K215	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y
K216	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K217	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y
K218	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y
K220	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y
K221	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y
K222	TSNPID , XASPEC , TEXT32	-	-	-	-	-	R	D	-	-	Y
K223	TSNPID	-	-	-	-	-	R	D	-	-	Y
K224	TSNPID , XACALL , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y
K225	TSNPID , XADBC1 , XADBC2	-	-	-	-	-	R	D	-	-	Y
K230	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y
K231	TSNPID	-	-	-	-	-	R	D	-	-	Y
K232	TSNPID	-	-	-	-	-	R	D	-	-	Y
K233	TSNPID , INSTNUM , XACALL , DBCALL , XATXT , LTHGTRID, GTRID	-	-	-	D	+	+	+	+	+	Y
K235	TCPCL , PRNM , TCPRC , TCPMS , IPADDR	-	-	-	D	+	D	+	+	+	Y
K251	IMPVER , FBASUPD , DEFVER	-	-	-	D	+	D	D	+	+	Y
K252	UPDERR	-	-	-	D	+	D	D	+	+	Y
K255	DMSE , FNAM	-	-	-	D	+	D	D	+	+	Y
K256	FNAM	-	-	-	D	+	D	D	+	+	Y
K257		-	-	-	D	+	D	D	+	+	Y
K258	FNAM	-	-	-	D	+	D	D	+	+	Y
K260	DEFVER , FBASUPD	-	-	-	D	+	D	D	+	+	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
K261	FNAM	-	-	-	D	+	D	D	+	+	Y
K262	FNAM	-	-	-	D	+	D	D	+	+	Y
K263	FNAM	-	-	-	D	+	D	D	+	+	Y
K269	OBJ1 , OST1 , BMD1 , OBJ3 , OST2 , BMD2	-	-	-	D	+	D	D	+	+	Y
K273	TRMA , UPDMODUL	-	-	-	D	+	D	D	+	+	Y
K274		-	-	-	D	+	D	D	+	+	Y
K277		-	-	-	D	+	D	D	+	+	Y
K278		-	-	-	D	+	D	D	+	+	Y
K279		-	-	-	D	+	D	D	+	+	Y
K300	UPDPRO , PGPOOL	-	-	-	D	+	D	D	+	+	Y
K303	UKCOP , UKCRN , UPDTYP , UKCLM	-	-	-	D	+	D	D	+	+	N
K304	USER , TACTYPE , UKCHSTA	-	-	-	D	+	D	D	+	+	N
K305	UPDTYP , PGS1 , PGS2 , UKCRN , PGPOOL	-	-	-	D	+	D	D	+	+	Y
K306	UPDTYP , PGS1 , PGPOOL	-	-	-	D	+	D	D	+	+	Y
K310	UPDTYP , UKCRN	-	-	-	D	+	D	D	+	+	N
K311	UPDTYP , UKCRN	-	-	-	D	+	D	D	+	+	N
K314		-	-	-	D	+	D	D	+	+	N
K317	UKCOP , UKCRN , UPDTYP , UKCLM , RCCC , RCDC , LTRM , USER	-	-	-	D	+	D	D	+	+	N
K318	UPDTYP , UKCRN , LTRM , USER	-	-	-	D	+	D	D	+	+	N
K320	USER , TACTYPE , UERCODE , UERINFO , RCDC	-	-	-	D	+	D	D	+	+	N

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
P001	XPFUNC , ACPNT , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	D	+	+	+	Y
P002	XPFUNC , ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	+	+	+	+	Y
P003	ACPNT , XPRJCT , XPLTH	-	-	-	D	+	+	+	+	+	Y
P004	ACPNT , OSLPAP , XPRJCT	-	-	-	D	+	+	+	+	+	Y
P005	ACPNT , XPNSEL , XPTSEL , XPLSSEL , XPCSSEL , XPHSSEL , XPLPSEL , XPCPSEL , XPHPSEL	-	-	-	D	+	+	+	+	+	Y
P006	ACPNT , OSLPAP , XP0OBID , XP1OBID , XP2OBID , XP3OBID , XP4OBID , XP5OBID , XP6OBID , XP7OBID , XP8OBID , XP9OBID	-	-	-	D	+	+	+	+	+	Y
P007	ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	+	+	+	+	Y
P008	ACPNT , OSLPAP , XPOSAS	-	-	-	D	+	+	+	+	+	Y
P009	ACPNT , OSLPAP , XPRJCT , XPLTH , XPOSAS	-	-	-	D	+	+	+	+	+	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
P010	ACPNT , OSLPAP , XPNSEL , XPTSEL , XPLSSEL , XPCSSEL , XPHSSEL , XPLPSEL , XPCPSEL , XPHPSEL , XPOSAS	-	-	-	D	+	+	+	+	+	Y
P011	ACPNT , OSLPAP , XP0OBID , XP1OBID , XP2OBID , XP3OBID , XP4OBID , XP5OBID , XP6OBID , XP7OBID , XP8OBID , XP9OBID , XPOSAS	-	-	-	D	+	+	+	+	+	Y
P012	XPCTYPE , XPCCLS , XPCVAL , XPCORR	-	-	-	D	+	+	+	+	+	Y
P013	ACPNT , OSLPAP , XPCRES , XPSRC , XPNDIA , XP1BOOL , XP2BOOL , XP3BOOL , XP4BOOL , XP5BOOL , XPOSAS	-	-	-	D	+	+	+	+	+	Y
P014	XPFUNC , ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPOSAS , XPCORR	-	-	-	D	+	+	+	+	+	Y
P015	XPFUNC , ACPNT , OSLPAP , XPLNK , XPSRC , XPNDIA , XPINI , XP1INFO , XP2INFO , XPOSAS , XPCORR	-	-	-	D	+	+	+	+	+	Y
P016	ACPNT , OSLPAP , XPLNK , XPNDIA , XPOSAS	-	-	-	D	+	+	+	+	+	Y
P017	XPPDU , XP1DIA , XP2DIA , XP3DIA	-	-	-	+	+	+	+	+	+	Y

Msg-Id	Inserts	S t a t i o n	S Y S L I N E	P a r t n e r	S Y S L O G	M S G T A C	S Y S O U T	S Y S L S T	C o n s o l e	U S E R D E S T	C o m p r e s s
P018	ACPNT , OSLPAP , XPPTYTYP , XPFSMN	-	-	-	D	+	+	+	+	+	Y
P019	ACPNT , OSLPAP , XPAPDU , XP3INFO	-	-	-	D	+	+	+	+	+	Y
P020	XPTRFAIL	-	-	-	D	+	+	+	+	+	Y
P021	XPEVT , ACPNT , OSLPAP , XPOSAS , XPASST	-	-	-	D	+	+	+	+	+	Y

¹ Wird in einer UTM-Anwendung mit einem Anmelde-Vorgang gearbeitet, dann werden diese Meldungen nicht erzeugt, d.h. sie werden auch nicht an die Meldungsziele MSGTAC oder SYSLOG ausgegeben.

Meldungsziele für K251- K322 und K800 - K899 (Meldungen von KDCUPD):

K800 sowie K252 - K300 und K802 - K850 werden auf *stderr* und *stdout* ausgegeben. K801 wird nur auf *stdout* ausgegeben. Die Meldungsziele für K303 - K320 und K851 - K858 können mit der LIST-Anweisung des Dienstprogramms KDCUPD gesteuert werden, Standard sind *stdout* und *stderr*.

Meldungsziele für K400 - K522 (Meldungen von KDCDEF):

Alle Meldungen werden auf *stdout* ausgegeben.

Meldungsziele für K600 - K622 (Meldungen von KDCCSYSL/KDCPSYSL):

Alle Meldungen werden auf *stdout* ausgegeben.

Meldungsziele für K650 - K698 (Meldungen von KDCMMOD/KDCMTXT):

K660, K661 und K686 bis K688 werden auf *stdout* ausgegeben, alle anderen Meldungen auf *stdout* und *stderr*.

Meldungsziele für U101 - U511

Die Meldungen mit den Meldungsnummern:

U101 bis U125	werden vom Dialog-Terminalprozess auf <i>stdout</i> ausgegeben.
U151 und U154 bis U157	werden vom Druckerprozess auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U171 bis U177	werden vom utmlog-Prozess auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U181 und U182	werden von verschiedenen Prozessen auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U184 bis U193	werden von verschiedenen Prozessen auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U201 bis U206	werden vom Timerprozess auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U221 bis U244	werden vom Mainprozess auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U245	wird vom Mainprozess auf <i>stdout</i> ausgegeben.
U246 und U247	werden vom Mainprozess auf <i>stderr</i> ausgegeben.
U251 bis U259	werden von kdcuslog auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U271 bis U281	werden von kdccsysl auf <i>stdout</i> und <i>stderr</i> ausgegeben.
U301 bis U323	werden von den Netzprozessen auf <i>stdout</i> und <i>stderr</i> ausgegeben.

- U341, U351 bis U363 werden von `kdckaa`, `kdcshut` bzw. `kdcrem` auf `stdout` und `stderr` ausgegeben.
- U370 bis U392 werden von `kdcprog` auf `stderr` ausgegeben.
- U500 bis U511 werden von der KDCDEF Unterstützung für die TNS-Generierung erzeugt. Die Meldungen treten nur auf, wenn die Option `CHECKTNS=YES` angegeben ist.

Die Meldungen sind selbsterklärend und werden zusätzlich in der Datei `utmgentns` bei der betroffenen Zeile abgelegt.

In Unix-Systemen ist eine detaillierte Information zu den in den Meldungen enthaltenen Unix-Fehlernummern (`&ERRNO`) aus dem C-Header-File `errno.h` zu entnehmen.

In Windows-Systemen können mit der Funktion `GetLastError()` die entsprechenden Fehlernummern angefordert werden.

5.12 Windows Event Logging-Meldungen

Die Installation, Deinstallation und der Betrieb von openUTM-Services werden in der Ereignisanzeige (Event Viewer) in Form von Ereignissen (Events) protokolliert. Die openUTM-Ereignisse befinden sich im Event Viewer im Bereich Application, die Quelle ist openUTM.

Alle 24 Stunden wird eine Event-Meldung ausgegeben (Messageld = 5), die darüber informiert, dass der openUTM-Service noch aktiv ist.

Es handelt sich um folgende Ereignisse:

MessageId	SymbolicName =	Ereignistext	Beschreibung
0	UTM_SERVICE_INF_MAIN	<Service Name>: main() started.	Die Routine main() wurde gestartet.
1	UTM_SERVICE_INF_READREG	<Service Name>: service_main(): ReadRegistryParameters() ok.	Die notwendigen Registry-Einträge sind erfolgreich eingelesen.
5	UTM_SERVICE_INF_REPORT_STATUS	<Service Name>: SetServiceStatus to SERVICE_RUNNING ok.	Der Dienst hat seinen Status erfolgreich auf SERVICE_RUNNING gesetzt.
6	UTM_SERVICE_INFO_RUN_SERVICE	<Service Name>: RunService(): called.	Die Routine RunService() wurde aufgerufen.
7	UTM_SERVICE_INF_SETDIR	<Service Name>: RunService(): directory changed successful.	In der Routine RunService() ist das Wechseln in das <filebase>-Verzeichnis erfolgreich durchgeführt.
8	UTM_SERVICE_INF_READY	<Service Name>: RunService(): service ready.	Die Routine RunService() meldet, dass der Service zur Verfügung steht.
9	UTM_SERVICE_INF_STOP	<Service Name>: ServiceStop() called.	Die Routine ServiceStop() wurde aufgerufen.
11	UTM_SERVICE_INF_MAIN_RETURN	<Service Name>: End of run: return from main().	Die Routine main() ist durchgelaufen.
20	UTM_SERVICE_INF_INSTALL	<Service Name>: service installed successful.	Der openUTM-Service wurde erfolgreich installiert.
21	UTM_SERVICE_INF_REMOVE	<Service Name>: service removed successful.	Der openUTM-Service wurde erfolgreich entfernt.
101	UTM_SERVICE_ERR_READREG	<Service Name>: service_main(): ReadRegistryParameters() failed. Additional information: GetLastError(): <Fehlercode>	Die Routine service_main() meldet, dass das Lesen der notwendigen Registry-Einträge gescheitert ist. Zusätzliche Information erhalten Sie durch den <Fehlercode>.

MessageId	SymbolicName =	Ereignistext	Beschreibung
103	UTM_SERVICE_ERR_REGCNTRL	<Service Name>: service_main(): RegisterServiceCtrlHandler() failed: return. Additional information: GetLastError(): <Fehlercode>	Die Routine service_main() meldet, dass das Registrieren des Services gescheitert ist. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
104	UTM_SERVICE_ERR_MAIN_REPORT	<Service Name>: service_main(): ReportStatusToSCMgr() failed: return. Additional information: GetLastError(): <Fehlercode>	Die Routine service_main() meldet, dass das Berichten des Services an den ServiceControlManager gescheitert ist. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
105	UTM_SERVICE_ERR_REPORT_STATUS	<Service Name>: <Fehlercode> failed. Additional information: GetLastError(): <Fehlercode>	Das Setzen von Service Status ist gescheitert. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
107	UTM_SERVICE_ERR_SETDIR	<Service Name>: RunService(): directory change failed. Additional information: GetLastError(): <Fehlercode>	Die Routine RunService() meldet, dass das Wechseln in das <filebase>-Verzeichnis gescheitert ist. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
108	UTM_SERVICE_ERR_DISPATCH	<Service Name>: MainMain(): StartServiceCtrlDispatcher failed. Additional information: GetLastError(): <Fehlercode>	Die Routine MainMain() meldet, dass der Start des ServiceControlDispatchers gescheitert ist. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
120	UTM_SERVICE_ERR_INSTALL	<Service Name>: service not installed Additional information: GetLastError(): <Fehlercode>	Die Installation des openUTM-Services ist gescheitert. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
121	UTM_SERVICE_ERR_REMOVE	<Service Name>: service not removed Additional information: GetLastError(): <Fehlercode>	Das Entfernen des openUTM-Services ist gescheitert. Zusätzliche Information erhalten Sie durch den <Fehlercode>.
122	UTM_SERVICE_ERR_UTMWORK	<Service Name>: utmwork.exe not found Additional information: GetLastError(): <Fehlercode>	Die Datei utmwork.exe wurde nicht gefunden. Zusätzliche Information erhalten Sie durch den <Fehlercode>.

MessageId	SymbolicName =	Ereignistext	Beschreibung
123	UTM_SERVICE_ERR_KDCA	<Service Name>: KDCA file not found Additional information: GetLastError(): <Fehlercode>	Die KDCA-Datei wurde nicht gefunden. Zusätzliche Information erhalten Sie durch den <Fehlercode>.

6 UTM-Returncodes

6.1 KDCS-Returncodes in KCRCCC

Folgende Kategorien sind zu unterscheiden:

Code	Kategorie	Maßnahmen
000	ohne Fehler	Operation wurde fehlerfrei ausgeführt.
01Z - 09Z	Bemerkungen	Operation wurde ausgeführt.
10Z - 19Z	Warnung bzw. leichte Fehler	Nach Ausführung geeigneter Maßnahmen kann der Teilprogrammmlauf sinnvoll fortgesetzt werden. Die Operation wurde nicht ausgeführt.
20Z - 39Z	Sonderfunktionen	Eine KDCS-Sonderfunktion wird signalisiert. Sonst wie Warnungen.
40Z - 69Z	Fehler	Die Operation wurde nicht ausgeführt. Der Programmmlauf wird meistens nicht mehr sinnvoll weiterarbeiten können. Kommunikation mit der Dialogstation ist noch möglich.
70Z - 99Z	Schwerwiegende Fehler	Eine Fortsetzung des Programmmlaufs ist nicht möglich. UTM setzt die Transaktion zurück und bricht den Vorgang ab. UTM erzeugt eine System-spezifische Fehler-Nachricht und gibt sie bei einem Dialog-Vorgang auf die Dialogstation aus.

Beachten Sie bitte folgende Punkte:

- Eine genauere Erklärung der Returncodes in KCRCCC finden Sie im openUTM-Handbuch „Anwendungen programmieren mit KDCS“ bei jedem einzelnen KDCS-Aufruf.
- Treten mehrere Fehler gleichzeitig auf, so werden die Fehler der jeweils höchsten Kategorie angezeigt. Wurde eine Operation nicht ausgeführt (Returncode > 09Z), so wird der Inhalt in <parm2> nicht verändert.
- Die Auswertung eines UTM-Dumps ist im [Kapitel „Der UTM-Dump“ auf Seite 59](#) beschrieben.

Folgende Returncodes sind definiert (Angabe der Feldnamen: COBOL-Name/C-Name):

Code	Bedeutung
000	Operation erfolgreich ausgeführt
01Z	Längenkonflikt bezüglich KCLA/kcla bzw. KCLKBPRG/kclcapa
02Z	Längenkonflikt bezüglich KCLPAB/kclspa
03Z	Name in KCMF/kcfn ist ungültig
04Z	Name in KCRN/kcrn wechselt
05Z	Eingabeformatierung wurde nicht mit dem in KCMF/kcfn angegebenen Formatkennzeichen durchgeführt. Zeilenmodus: 1.Zeichen von KCMF/kcfn ungleich Space
06Z	Zeitangabe wechselt innerhalb der Nachricht (Aufruf DPUT)
07Z	Längenkonflikt bezüglich KCLI/kcli
08Z	Beim Lesen mit Warten im Aufruf DGET: Es liegt zurzeit keine Nachricht vor.
10Z	Nachricht wurde schon vollständig gelesen
11Z	Beim Lesen ohne Warten im Aufruf DGET: Es liegt keine Nachricht vor.
12Z	von der angegebenen Vorgangs-Identifikation liegen keine Nachrichten (mehr) vor, oder es existiert kein Vorgangsstapel mit der angegebenen Nummer.
14Z	Name in KCRN/kcrn nicht gefunden
16Z	die Operation ist nicht zulässig und wurde nicht ausgeführt
19Z	Funktionstaste bzw. Sonderfunktion nicht generiert
20Z bis 39Z	KDCS-Sonderfunktionen (Kurznachrichten)
40Z	Operation kann vom System nicht durchgeführt werden (Generierungsfehler, Systemfehler, Deadlock, langdauernde Sperren)
41Z	Operation an dieser Stelle nicht erlaubt
42Z	Operationsmodifikation ungültig
43Z	Längenangabe in KCLM/kclm, KCLI/kcli, KCLA/kcla oder KCWTIME/kcwtime ungültig.
44Z	Name in KCRN/kcrn ungültig
45Z	Formatkennzeichen KCMF/kcfn oder Erzeugungszeitpunkt (DGET) ungültig
46Z	Name in KCLT/kclt oder KCPA/kcpa oder KCLANGIDkclangid, KCTERRID/kcterrid, KCCSNAME/kccsname oder KCQMODE/kcqmode ist ungültig
47Z	Speicherbereich <parm2> fehlt, die Bereichsadresse ist ungültig, oder der Bereich ist in der angegebenen Länge nicht lesbar/beschreibbar
48Z	ungültige Schnittstellenversion
49Z	Nicht verwendete Parameter sind ungleich binär Null.
51Z	Reihenfolge beim DPUT-Aufruf nicht eingehalten

Code	Bedeutung
52Z	Nachrichtenziel in KCRN ist nicht erlaubt (Aufruf DPUT)
53Z	Wert in KCDPID/kcdpid bzw. KCGTM/kcgtm ist ungültig (Aufruf DGET).
54Z	Wert in KCNORPLY/kcnoreply ist ungültig (bei CTRL PR oder PE: Wert ungleich Y oder binär 0).
55Z	Name in KCPI/kcpi ist ungültig (Aufruf APRO)
56Z	Wert von KCMOD oder Zeitangabe in KCTAG/kcday,...,KCSEK/kceec ist ungültig (Aufruf DADM, DPUT)
57Z	Wert in KCPOS/kcpos ist ungültig (Aufruf MCOM).
58Z	Wert in KCOF/kcof ungültig (Aufruf APRO) oder der Wert in KCNEG/kcneg ist ungültig (Aufruf MCOM).
70Z	Operation kann vom System nicht durchgeführt werden (Generierungsfehler, Systemfehler)
71Z	Operation an dieser Stelle nicht erlaubt, z.B. noch kein INIT
72Z	Operationsmodifikation ungültig
73Z	Längenangabe in KCLA/kcla, KCLM/kclm oder KCLI/kcli ungültig
74Z	Name in KCRN/kcrn ungültig
75Z	Formatkennzeichen KCMF/kcfn ungültig
77Z	Speicherbereich <parm2> fehlt, die Bereichsadresse ist ungültig, oder der Bereich ist in der angegebenen Länge nicht lesbar/beschreibbar
79Z	Operationscode nicht interpretierbar *
80Z	Generierungsfehler vor Programmstart
81Z	Ziel am PEND in Widerspruch mit Ziel am MPUT
82Z	PEND-Variante widerspricht dem Operanden KCOM/kcom oder KCRN/kcrn im MPUT-Aufruf
83Z	Vor einem PEND KP,RE,FI,ER,FR wurde in einem Dialog-Programm kein MPUT gegeben oder vor einem PEND KP,RE wurde in einem Asynchron-Programm kein MPUT gegeben oder vor einem PGWT PR wurde ein MPUT gegeben.
84Z	PEND fehlt
86Z	Ein Nachrichten-Komplex wurde nicht abgeschlossen oder nach einem APRO AM wurde kein FPUT/DPUT für den adressierten Auftragnehmer- Vorgang gegeben.
87Z	Operationsmodifikation ist unzulässig
88Z	Ungültige Schnittstellenversion
89Z	Nicht verwendete Parameter sind ungleich binär Null.

* Der Returncode 79Z (Operationscode nicht interpretierbar) ist keiner speziellen Operation zuzuordnen.

6.2 Interner Returncode KCRCDC

Der inkompatible KDCS-Returncode KCRCDC besteht aus 4 abdruckbaren Zeichen und enthält genauere Angaben zur Fehlerursache als der kompatible Returncode KCRCCC.

Der inkompatible Returncode kann in folgenden Fällen gesetzt sein:

- wenn ein KDCS-Aufruf den kompatiblen Returncode 40Z liefert.
- wenn openUTM einen Vorgang mit PEND ER abnormal beendet ($KCRCCC \geq 70Z$)
- wenn eine Transaktion implizit durch openUTM zurückgesetzt wird.

Der Returncode KCRCDC hat folgenden Aufbau:

sm ##

Das erste Zeichen (s) gibt an, welcher Teil von openUTM den Fehler gemeldet hat:

A	Administration
K	UTM-Systemcode oder ROOT-Code
U	UPIC (Server-Teil)

Bei allen Returncodes, die mit dem Zeichen K beginnen, bezeichnet der zweite Buchstabe (m) das openUTM-Modul, das den Fehler meldet (siehe folgende Liste).

3	KCSPEND (Operationen PEND und RSET)
6	KCSMPUT (Operation MPUT)
7	KCSFPUT (Operationen PEND und RSET)
8	KCSSSB (Operationen SGET, SPUT, SREL, PTDA, GTDA und UNLK)
9	KCSLPUT (Operation LPUT)
A	KCSWAIT
B	KCSDGET (Operation DGET)
C	KCDCTRL (Operation CTRL)
D	KCDAPRO (UTM-D-Operationen APRO, MPUT, .MGET, ...)
E	KCSPADM (Operation PADM)
F	KCSINFO (Operation INFO)
G	KCSDADM (Operation DADM)
H	KCSSIGN (Operation SIGN)
I	KCSISLP (interne Operationen)
K	allgemeiner DC-Code
L	KCSSTRT
M	KCSTRHD (interne Operationen)
N	KCSDLO
Q	KCSCRO (Operation QCRE)
R	KDCRTMM (ROOT-Code)
S	KCDSEFU (UTM-D Servicefunktionen)
T	KCDOSTM (OSI TP Transaktionsbeendigung)
U	KDCRTDB (Bearbeitung von DB-Prozessen)

V KCSTUTM (Operation MGET, MPUT für UPIC- und Socket-Clients)
 W KCDOSSF (OSI TP Spezialfunktionen)
 Y KCDSYPM (LU6.1/OSI TP Transaktionsbeendigung)

Die folgende Tabelle beschreibt die Fehlerursachen und ggf. die Maßnahmen, die zur Fehlerbehebung notwendig sind.

In den mit ⁽¹⁾ gekennzeichneten Fällen wird bei TESTMODE=OFF der PEND ER-Dump unterdrückt. Gleichzeitig unterbleibt das Nachladen, es sei denn für den aktuellen TAC sind PGWT-Aufrufe erlaubt und der KDCS-Aufruf, bei dem der Fehler auftrat, war kein PEND-Aufruf.

Code	Modul	Fehlerursache und ggf. Maßnahme
A010	KCSSADM	User ist kein SAT-Administrator.
A011	KCSSADM	TAC hat keine SAT-Berechtigung.
A012	KCSSADM	Subsystem SAT nicht verfügbar.
A013	KCSSADM	SAT-Version ist nicht kompatibel.
A015	KCSSADM	Ein Inverser KDCDEF läuft oder soll gestartet werden.
A016	KCSSADM	Die globale Cluster-Administration ist gesperrt.
A100	KCSADMI	Beim Aufruf der Programmschnittstelle der Administration ist die für den Parameterbereich angegebene Adresse entweder gar nicht oder nicht in der Länge des Parameterbereichs zugreifbar oder nicht auf Wortgrenze ausgerichtet. Maßnahme: Überprüfung der Adresse, die im Teilprogramm für den Parameterbereich angegeben wurde.
A101	KCSADMI	Beim Aufruf der Programmschnittstelle der Administration wurde intern ein ungültiger Returncode zurückgeliefert. Systemfehler. Maßnahme: Bitte Systemdienst informieren. Zur Diagnose wird der PEND-ER-Dump benötigt.
EXIT	KDCROOT	– In einem C-Programm wurde unzulässigerweise die exit()-Funktion aufgerufen. – In einem COBOL-Programm wurde die Anweisung STOP RUN ausgeführt. – Das COBOL-Laufzeitsystem hat in einem COBOL-Programm einen Datenfehler, einen Indexfehler o.ä. erkannt. Die Analyse des Aufrufstacks im zugehörigen core gibt Hinweise auf den Verursacher.
K300	KCSPEND	Ungültige Operationsmodifikation beim PEND-Aufruf.
K301	KCSWAIT	(Siehe KA00) Der Puffer für die Eingabenachricht ist zu klein. Maßnahme: Generierung mit KDCDEF, Pufferlänge mit MAX TRMSGLTH=länge größer definieren. Es wird kein PEND ER Dump erzeugt.

Code	Modul	Fehlerursache und ggf. Maßnahme
K302	KCSPEND	Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
K303	KCSPEND	Ungültige Operationsmodifikation beim PGWT-Aufruf.
K304	KCSPEND	Das DB-System verlangt einen CLOSE DB-Aufruf vor PEND RE/FI, der aber nicht aufgerufen wurde.
K305	KCSPEND	Es wurde ein PEND KP aufgerufen bzw. ein PEND PA/PR mit TASK-Wechsel, obwohl das DB-System, bei dem eine Transaktion geöffnet wurde, einen PEND KP nicht erlaubt.
K306	KCSPEND	Die Transaktion und der Vorgang mussten zurückgesetzt werden, weil das DB-System die DB-Transaktion beim Schließen zurückgesetzt hat.
K307	KCSPEND	PEND PS ist nur im ersten Teil des Anmeldevorgangs erlaubt.
K308	KCSPEND	Vor einem PEND KP oder PEND PA/PR mit TASK-Wechsel wurde die DB-Transaktion geschlossen.
K309	KCSPEND	Der in KCRN angegebene TAC ist nicht als Folgetac erlaubt.
K310	KCSPEND	Fehler beim Speichern der MPUT-Nachricht beim PEND PA/PR mit TASK-Wechsel.
K311	KCSPEND	Bei einem PEND RE/FI wird ein vorläufiges Transaktionsende (PTC) erreicht, aber das DB-System, bei dem eine Transaktion offen ist erlaubt den Transaktionsstatus nicht.
K313	KCSPEND	In einem Teilprogrammablauf wurde ein SIGN OF/OB gegeben, aber die Ausgabenachricht ist für einen Auftragnehmer bestimmt oder der Programmablauf wurde mit einer nicht erlaubten PEND-Variante beendet.
K314	KCSPEND	Ein Teilprogrammablauf in einem Anmeldevorgang wurde nach einem erfolgreichen SIGN ON nicht mit PEND PS beendet.
K315	KCSPEND	Bei Vorgangskellerung wurde vor PEND FI ein MPUT PM mit KCLM/kclm > 0 gegeben. Die letzte Ausgabe vom Vorgängervorgang war eine LINEMODE-Nachricht und kann nur unverändert ausgegeben werden.
K316	KCSPEND	Es wurde PEND RS im Teilprogramm aufgerufen oder es ist eine Situation (z.B. Verbindungsverlust) eingetreten, die das Rücksetzen der Transaktion durch UTM erfordert, aber PEND RS ist nicht erlaubt, weil kein Vorgangswiederanlauf möglich ist. Dies ist der Fall, wenn - der Vorgang durch einen UPIC-Client oder einen OSI TP Auftraggeber, der nicht die Functional Unit Commit ausgewählt hat, gestartet wurde, und auf auf der Verbindung/Association kein Benutzer mit Wiederanlaufeigenschaft angemeldet ist und auch kein lokaler Vorgangswiederanlauf möglich ist, weil der letzte Sicherungs-Punkt nicht mit PEND SP oder PEND FC (nur UPIC) gesetzt wurde - oder der letzte Sicherungspunkt mit PGWT CM gesetzt wurde. Es wird kein PENDER-Dump geschrieben.

Code	Modul	Fehlerursache und ggf. Maßnahme
K318	KCSPEND	PGWT ist in einem Teilprogramm aufgerufen worden, für dessen TAC - keine TAC-Klasse generiert ist oder - falls TAC-Klassen generiert sind: für die TAC-Klasse ist nicht PGWT=YES generiert oder - falls TAC-PRIORITIES generiert ist: der TAC ist nicht mit PGWT=YES generiert Maßnahme: Generierung mit KDCDEF korrigieren.
K319	KCSPEND	Anzahl der Tasks ist nicht ausreichend für die Nutzung von PGWT oder aus einer System-Task wurde PGWT aufgerufen (nur von einem generierten privilegierten LTERM aus möglich). Maßnahme: Anzahl der Tasks erhöhen.
K320	KCSPEND	Nach einem RSET in einer verteilten Transaktion wurde die Transaktion nicht zurückgesetzt, obwohl UTMD RSET = GLOBAL generiert wurde.
K322	KCSPEND	Die Operationsmodifikation FC ist nicht in Asynchron- oder Server-Vorgängen erlaubt.
K323	KCSPEND	Die Operationsmodifikation FC ist nicht im Anmeldevorgang erlaubt, wenn ein Vorgangswiederanlauf notwendig ist.
K324	KCSPEND	KCRN ist nicht mit Leerzeichen belegt bei PEND RS oder PEND FR.
K325	KCSPEND	Die ausgewählte Operationsmodifikation ist nicht im Anmeldevorgang oder MSGTAC-Programm erlaubt oder der Resource Manager des XA-Anschluss fordert beim xa_end ein Rücksetzen der Transaktion im Anmeldevorgang oder MSGTAC-Programm.
K326	KCSPEND	Unerlaubter Wechsel zwischen Dialog- und Asynchron-Transaktionscode oder Folgetac bei PEND FC ist kein Vorgangstac oder Folgetac bei PEND PA/PR, PS, KP, RE oder SP ist kein Folgetac
K327	KCSPEND	Die Operationsmodifikation FC ist im Anmeldevorgang nicht erlaubt, wenn der letzte Anmeldeversuch nicht erfolgreich war.
K328	KCSPEND	Der Anmeldevorgang kann nicht normal beendet werden, wenn die Gültigkeitsdauer des Passwort abgelaufen ist, aber das Passwort wurde nicht geändert wurde.
K329	KCSPEND	Der Anmeldevorgang kann nicht normal beendet werden, das mit KDCUPD übertragene Passwort die geforderte Komplexitätsstufe nicht erfüllt oder zu kurz ist und nicht mit SIGN CP geändert wurde.
K330	KCSPEND	Ein SIGN OB im Anmeldevorgang für Terminal ist nur in Kombination mit einem MPUT NT/NE erlaubt.
K331	KCSPEND	Wird ein Anmeldevorgang mit anschließendem Vorgangswiederanlauf ohne Abmeldung des Benutzers beendet, so ist kein MPUT NT/NE erlaubt.
K332	KCSPEND	Wird der Anmeldevorgang mit anschließendem Vorgangswiederanlauf mit PEND FI ohne vorhergehenden MPUT beendet, so beendet openUTM den offenen Vorgang

Code	Modul	Fehlerursache und ggf. Maßnahme
K333	KCSPEND	Es soll auf eine DGET-Nachricht gewartet werden; das Folgeteilprogramm liegt aber in keiner TAC-Klasse. Maßnahme: Folge-TAC mit TAC-Klasse generieren.
K334	KCSPEND	Es soll auf eine DGET-Nachricht gewartet werden; es sind nur PEND PA/PR/RS/ER/FR und PGWT RB erlaubt.
K335	KCSPEND	Der Vorgang wird abnormal beendet, da beim Timeout an der PGWT-Börse festgestellt wurde, dass nicht mehr genügend Prozesse aktiv sind, um den Vorgang normal fortzusetzen. Es wird kein PENDER-Dump geschrieben.
K336	KCSPEND	Vor PEND KP/RE/FI/FR/ER oder PGWT KP wurde kein MPUT aufgerufen, obwohl er erforderlich ist.
K337	KCSPEND	Vor PGWT PR wurde ein MPUT aufgerufen.
K338	KCSPEND	Die Transaktion wurde zurückgesetzt, weil die Anwendung beendet wird.
K339	KCSPEND	Das MSGTAC-Teilprogramm wurde beendet ohne mit FGET eine Meldung gelesen zu haben.
K340	KCSPEND	Beim Zurücksetzen einer Transaktion hat die Datenbank einen Fehler gemeldet, der das Nachladen des Anwendungsprogramm notwendig macht.
K341	KCSPEND	Nur in Cluster-Anwendungen: In der aktuellen Transaktion wurde ein User kreiert oder gelöscht. Die User-Datei konnte nicht gesperrt werden, da gerade durch einen KDCDEF-Lauf auf die User-Datei zugegriffen wird.
K342	KCSPEND	Nur in Cluster-Anwendungen: In der aktuellen Transaktion wurde ein User kreiert oder gelöscht. Die User-Datei konnte nicht gesperrt werden, da gerade ein anderer Prozess der selben Knotenanwendung oder eine andere Knotenanwendung die User-Datei gesperrt hat.
K343	KCSPEND	PET-Status abgelehnt, weil Vorgang durch die Administration gesperrt.
K344	KCSPEND	Der CR konnte nicht geschrieben werden, weil globales ADM-Lock gesetzt ist.
K345	KCSPEND	Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
K346	KCSPEND	Das Beenden eines offenen Vorgangs durch den Anmelde-Vorgang ist nicht erlaubt, wenn eine Transaktion des offenen Vorgangs im Zustand "vorläufiges Traksaktionsende (PTC)" ist. Der Anmelde-Vorgang wird abnormal beendet.
K347	KCSPEND	In einer Cluster Anwendung will eine Transaktion in den Status "vorläufiges Traksaktionsende (PTC)" gehen, aber beim Schreiben des Status in die User-Datei kam es zu einem Fehler.
K348	KCSPEND	Der Cluster Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF; Cluster Pagepool mit CLUSTER PGPOOL=(anzahl,...) größer einrichten.
K349	KCSPEND	Die Verbindung zum Partner wurde abgebaut.

Code	Modul	Fehlerursache und ggf. Maßnahme
K350	KCSPEND	Ungültige Operationsmodifikation nach einer abnormalen Beendigung eines Dialogs, auf dem die Commit Functional Unit ausgewählt wurde, mit CTRL AB.
K351	KCSPEND	Die Transaktion ist zum Rücksetzen markiert, es wurde aber ein PEND/PGWT-Aufruf zum Vorsetzen einer Transaktion verwendet.
K360	KCSPEND	Die Transaktion wurde mit PGWT RB zurückgesetzt, weil KCSPEND vom KDCROOT mit PGWT RB aufgerufen wurde.
K361	KCSPEND	Die Transaktion wurde mit PEND RS zurückgesetzt, weil KCSPEND vom KDCROOT mit PEND RS aufgerufen wurde.
K362	KCSPEND	Die Transaktion wurde mit PEND ER zurückgesetzt, weil KCSPEND vom KDCROOT mit PEND ER aufgerufen wurde.
K363	KCSPEND	Die Transaktion wurde mit RSET zurückgesetzt, weil KCSPEND vom KDCROOT mit RSET aufgerufen wurde.
K601	KCSMPUT	Puffer für Dialognachrichten zu klein generiert. Maßnahme: Generierung ändern, Puffer mit MAX NB=länge größer wählen. ¹
K602	KCSMPUT	Der MPUT-Aufruf verlangt eine Format-Ausgabe, es wurde aber keine Formatierung generiert. ¹
K603	KCSMPUT	MPUT mit Bildschirmfunktion KCCARD, die DSS hat jedoch keinen Ausweisleser. Maßnahme: Generierung oder Teilprogramm ändern. ¹
K604	KCSMPUT	MPUT mit Bildschirmfunktion KCCARD u. Benutzererkennung hat Operanden CARD=(position,string): Ausweisleser kann nicht für KDCSIGN-Prüfung und Dateneingabe zugleich verwendet werden. Maßnahme: Generierung oder Teilprogramm ändern.
K605	KCSMPUT	MPUT mit Bildschirmfunktion KCCARD und Ausgabe im Formatmodus. Maßnahme: Programm ändern.
K606	KCSMPUT	Folgeteil einer MPUT-Nachricht im Formatmodus und KCDF ist ungleich binär 0. Maßnahme: Teilprogramm ändern
K607	KCSMPUT	Die MPUT-Nachricht (angegeben durch Adresse und Länge) überdeckt teilweise einen ROOT-internen Pufferbereich, den MPUT-Buffer. Maßnahme: Programm ändern (Länge des MPUT zu groß?)
K608	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
K610	KCSMPUT	Die Operationsmodifikation ES des MPUT-Aufrufs ist nur für UPIC- und SOCKET-Partner erlaubt.
K611	KCSMPUT	Beim Aufruf MPUT ES ist das Feld KCRN nicht mit Leerzeichen belegt.
K612	KCSMPUT	Im MSGTAC-Teilprogramm wurde MPUT aufgerufen

Code	Modul	Fehlerursache und ggf. Maßnahme
K613	KCSMPUT	Operationsmodifikation ungültig
K614	KCSMPUT	MPUT PM im Asynchronvorgang
K615	KCSMPUT	MPUT PM im ersten Teil des Anmeldevorgangs
K616	KCSMPUT	MPUT PM mit KCLM/kclm ungleich 0 im Anmeldevorgang
K617	KCSMPUT	MPUT PM, aber der Vorgang ist weder eingeschoben noch ein Anmeldevorgang.
K618	KCSMPUT	MPUT PM im Anmeldevorgang für einen UPIC-Verbindungs-User. MPUT PM im ersten Teil des Anmeldevorgangs
K701	KCSFPUT	UTM lehnt eine asynchrone Nachricht ab, weil im Pagepool bereits Level 2 erreicht ist. Maßnahme: Siehe K345
K702	KCSFPUT	In KCRN wurde der Name eines DIALOG-TACs angegeben. Maßnahme: Generierung oder Programm ändern
K703	KCSMPUT	(Siehe K602) Der MPUT-Aufruf verlangt eine Format-Ausgabe, es wurde aber keine Formatierung generiert. ¹
K704	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
K705	KCSFPUT	Eine asynchrone Nachricht soll mit FPUT an einen LTERM- oder (OSI)-LPAP-Partner geschickt werden, für die in der KDCDEF-Generierung LTERM ...,QAMSG=N angegeben ist und zu der keine Verbindung besteht. UTM nimmt die Nachricht deshalb nicht an. Maßnahme: Verbindung herstellen
K706	KCSFPUT	Eine asynchrone Nachricht soll mit FPUT/DPUT an eine Dialog-UTM-Datenstation im Zeilenmodus geschickt werden. Die Nachricht ist länger als der Pufferbereich, der bei der Generierung im Parameter MAX NB festgelegt wurde. UTM nimmt die Nachricht nicht an. Maßnahme: Angabe bei MAX NB vergrößern oder Nachrichtenteile verkürzen
K707	KCSFPUT	Die bei einem DPUT-Aufruf angegebene Zielzeit verletzt die durch die Generierung festgelegten Grenzwerte (MAX DPUTLIMIT1= , DPUTLIMIT2=) Maßnahme: Zeitangabe beim DPUT-Aufruf oder Generierung ändern

Code	Modul	Fehlerursache und ggf. Maßnahme
K708	KCSFPUT	Eine asynchrone Nachricht soll mit FPUT/DPUT an eine LTERM- oder (OSI-)LPAP-Partner, einen Asynchron-TAC, eine TAC-Queue, eine User-Queue, eine temporäre Queue oder ein (OSI-)LPAP geschickt werden. An diesem Ziel liegen bereits so viele asynchrone Nachrichten vor, wie bei der Generierung als Maximalwert festgelegt wurde. (KDCDEF: LTERM QLEV=, TAC QLEV=, USER QLEV=, QUEUE QLEV=, LPAP QLEV=, OSI-LPAP QLEV=)
K711	KCSFPUT	In einem DPUT-Aufruf enthält KCRN ein neues Ziel, obwohl die vorangehende DPUT Nachricht an ein anderes Ziel noch nicht abgeschlossen ist.
K712	KCSFPUT	In KCRN wurde der Name eines UTM-D-Partners, einer MUX-Verbindung, oder eines UPIC-Partners angegeben. Maßnahme: Generierung oder Programm korrigieren
K713	KCSFPUT	Ursache ist ein FPUT/DPUT-Aufruf an einen Ausweisleser. Maßnahme: Teilprogramm ändern
K714	KCSFPUT	Bei einem FPUT wurde in KCRN der Name eines Message-Komplexes angegeben. Maßnahme: Programm korrigieren
K715	KCSFPUT	In KCRN wurde der Name eines Message-Komplexes angegeben; dieser Name passt jedoch nicht zu dem momentan eröffneten Message-Komplex Maßnahme: Programm korrigieren
K716	KCSFPUT	In KCRN wurde der Name eines Message-Komplexes angegeben; z.Zt. ist jedoch kein Message-Komplex eröffnet. Maßnahme: Programm korrigieren
K717	KCSFPUT	In KCRN wurde der Name eines TACs angegeben; dies ist unvereinbar mit der Angabe in KCOM. Maßnahme: Programm korrigieren
K718	KCSFPUT	Ein Asynchronvorgang, der unter einem gelöschten LTERM abläuft, hat versucht einen FPUT oder DPUT abzusetzen.
K719	KCSFPUT	Ein Asynchronvorgang, der unter einem gelöschten USER abläuft, hat versucht einen FPUT oder DPUT abzusetzen.
K720	KCSFPUT	Bei DPUT-Aufruf an eine USER-Queue: Es gibt keinen USER mit dem in KCRN angegebenen Namen oder der USER wurde gelöscht.
K721	KCSFPUT	Bei DPUT-Aufruf an eine USER-Queue: Der KSET des Benutzers und die Schreib-ACL der USER-Queue haben keinen gemeinsamen Schlüssel
K722	KCSFPUT	Zeitgesteuerte DPUT-Aufrufe für USER-Queues sind nicht möglich. (KCMOD ungleich Leerzeichen)
K723	KCSFPUT	Bei DPUT-Aufruf an ein QUEUE-Objekt: Es gibt kein QUEUE-Objekt mit dem in KCRN angegebenen Namen.
K724	KCSFPUT	Bei DPUT-Aufruf an eine USER-Queue: Der KSET des LTERMs und die Schreib-ACL der USER-Queue haben keinen gemeinsamen Schlüssel

Code	Modul	Fehlerursache und ggf. Maßnahme
K725	KCSFPUT	Zeitgesteuerte DPUT-Aufrufe für QUEUE-Objekte sind nicht möglich. (KCMOD ungleich Leerzeichen)
K726	KCSFPUT	Ungültiger Wert in KCQTYP.
K731	KCSFPUT	An einen TAC, der mit CALL=NEXT generiert ist, darf keine Asynchronnachricht geschickt werden.
K732	KCSFPUT	An KDCMSGLT darf keine Asynchronnachricht geschickt werden.
K733	KCSFPUT	An ein implizit für die interne Cluster-Kommunikation generiertes LTERM darf keine Asynchronnachricht geschickt werden.
K801	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
K802	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
K804	KCSSSB	Es wurden keine GSSBs generiert oder es wurden mehr GSSBs erzeugt als per Generierung erlaubt.
K805	KCSSSB	Es wurden mit SPUT mehr LSSBs erzeugt als bei der Generierung angegeben wurde.
K810	KCSSSB	Beim Zugriff auf einen GSSB, TLS oder ULS: Der Bereich kann nach Ablauf einer Wartezeit von der Transaktion nicht belegt werden. Maßnahme: Wartezeit größer machen, dazu in KDCDEF-Generierung Parameter RESWAIT=(zeit1,...) erhöhen.
K811	KCSSSB	Beim Zugriff auf einen GSSB, TLS oder ULS: Der Bereich ist derzeit von einer anderen Transaktion "unbestimmt lange" gesperrt, d.h. sie hat den Bereich belegt und danach einen PEND KP oder PGWT KP Aufruf gegeben.
K812	KCSSSB	Beim Zugriff auf einen ULS: Die Benutzerkennung, auf deren ULS zugegriffen werden soll, wird gelöscht.
K813	KCSSSB	Die Anwendung wird beendet
K820	KCSSSB	Das Warten auf einen globalen sekundären Speicherbereich würde zu einem Deadlock führen.
K822	KCSPEND	(Siehe K348) Der Cluster Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF; Cluster Pagepool mit CLUSTER PGPOOL=(anzahl,...) größer einrichten.
K823	KCSSSB	Nur in Cluster-Anwendungen: Bei der Anforderung der Datei-Sperre auf die Verwaltungsdatei für GSSB oder ULS kam es zu einem Timeout.

Code	Modul	Fehlerursache und ggf. Maßnahme
K824	KCSSSB	Beim Zugriff auf einen GSSB, TLS oder ULS: Der Bereich ist momentan gesperrt und die Task kann nicht auf die Freigabe der Sperre warten, weil bereits zuviele Tasks in einem Wartezustand sind. Maßnahme: - Mehr Tasks starten. - Die Anzahl der Tasks für die TACs begrenzen, die auf GSSB oder ULS zugreifen, und mehr Tasks als diese Grenze starten. Die Informationen, wie Sie UTM-Services, die GSSB bzw. ULS Speicherbereiche verwenden, auf die Tasks einer UTM-Cluster-Anwendung verteilen, finden Sie im openUTM-Handbuch "Einsatz von openUTM-Anwendungen"
K825	KCSSSB	Nur in Cluster-Anwendungen: Beim Zugriff auf einen GSSB oder ULS: Die Warteschlange an dem angeforderten Bereich hat bereits die maximale Länge erreicht; der Vorgang kann z.Zt. nicht auf diesen Bereich warten.
K826	KCSSSB	Nur in Cluster-Anwendungen: Beim Zugriff auf einen GSSB oder ULS: Bei der Anforderung eines internen Betriebsmittels kam es zu einem transienten Fehler; ein Zugriff auf den angeforderten Bereich ist z.Zt. nicht möglich.
K827	KCSSSB	Nur in Cluster-Anwendungen: Beim Zugriff auf einen GSSB oder ULS: Der Bereich ist momentan durch einen ausgefallenen Knoten gesperrt; ein Warten auf diesen Bereich ist z.Zt. nicht sinnvoll.
K901	KCSLPUT	UTM lehnt LPUT-Aufruf ab, weil im Pagepool bereits Level 2 erreicht ist. Maßnahme: Programm ändern oder per Generierung Pagepool vergrößern
K902	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
K903	KCSLPUT	UTM lehnt LPUT-Aufruf ab, da beim letzten Schreibvorgang auf die Benutzer-Protokolldatei(en) ein DMS-Fehler aufgetreten ist. Der DMS-Fehler ist mit der K043-Meldung ausgegeben worden. Die im Pagepool gepufferten Sätze bleiben erhalten. Maßnahme: K043-Meldung und DMS-Fehlercode auswerten, Benutzer-Protokolldatei(en) restaurieren oder neu anlegen, anschließend mit dem Administrationskommando KDCLOG fortfahren (oder über entsprechende Programmschnittstelle). Die LPUT-Sätze im Pagepool werden dann ausgegeben. Die Sperre für den LPUT-Aufruf wird aufgehoben.
KA00	KCSWAIT	Der Puffer für die Eingabenachricht ist zu klein. Maßnahme: Generierung mit KDCDEF, Pufferlänge mit MAX TRMSGLTH=länge größer definieren. Es wird kein PEND ER Dump erzeugt.

Code	Modul	Fehlerursache und ggf. Maßnahme
KA01	KCSWAIT	Ein schwerwiegender Fehler in der Kommunikation zwischen Auftraggeber und Auftragnehmer (UTM-D) ist aufgetreten. Die Kommunikation kann nicht weitergeführt werden, UTM bricht den Auftragnehmer-Vorgang mit PEND ER ab. Mögliche Ursache: - PEND ER durch den Auftraggeber. - Timeout auf der Verbindung zum Auftraggeber Diagnosehilfen: - VTV-Fehlernachricht K086 auswerten - bei PEND ER im AG-Vorgang Ursache im Dump suchen.
KA02	KCSWAIT	Der Pagepool in KDCFILE ist voll, deshalb konnten der im PGWT wartenden Task die Nachrichten nicht übergeben werden. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten.
KA03	KCSWAIT	UTM hat eine unverschlüsselte Nachricht empfangen, obwohl eine verschlüsselte Nachricht erwartet wurde.
KA04	KCSWAIT	Die Verbindung zum Partner wurde abgebaut.
KA05	KCSWAIT	Eine Transaktion im Zustand PET wurde auf Anforderung der Administration zurückgesetzt.
KB01	KCSDGET	Im ersten Teil des Anmeldevorgangs sind DGET-Aufrufe nicht erlaubt.
KB02	KCSDGET	Es muss bereits auf eine DGET-Nachricht gewartet werden; daher sind keine weiteren DGET-Aufrufe erlaubt.
KB03	KCSDGET	Ungültiger Wert von KCQTYP.
KB04	KCSDGET	Zu dem in KCRN angegebenen Namen vom Typ KCQTYP existiert kein Objekt bzw. dieses Objekt wurde gelöscht.
KB05	KCSDGET	Nachrichten an Dialog- oder Asynchron-TACs können nicht mittels DGET-Aufruf gelesen werden.
KB06	KCSDGET	Die Lese-ACL der TAC- bzw. USER-Queue und der KSET des LTERMs des Benutzers haben keinen gemeinsamen Schlüssel.
KB07	KCSDGET	Die Lese-ACL der TAC- bzw. USER-Queue und der KSET des Benutzers haben keinen gemeinsamen Schlüssel.
KB08	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KB10	KCSDGET	Negative Wartezeit beim Aufruf DGET FT/BF oder: Wartezeit ungleich 0 beim Aufruf DGET NT/BN/PF/PN.
KB11	KCSDGET	Für das MSGTAC-Programm sind DGET-Aufrufe mit Warten nicht erlaubt.

Code	Modul	Fehlerursache und ggf. Maßnahme
KB12	KCSDGET	Beim Aufruf DGET NT/BN/PN passen Name oder Typ der angegebenen Queue nicht zum vorherigen DGET-Aufruf des aktuellen Teilprogrammablaufes.
KB13	KCSDGET	Es wurde versucht, von einer USER Queue zu lesen, obwohl die Anwendung ohne Benutzer generiert wurde.
KB14	KCSDGET	Es wurde versucht, von einer TAC Queue mit STATUS= HALT oder KEEP zu lesen.
KB15	KCSDGET	Es wurde versucht, von einer TAC Queue mit ADMIN=Y zu lesen; der Benutzer ist aber kein Administrator.
KB16	KCSDGET	Operationsmodifikation in KCOM ungültig.
KB17	KCSDGET	Bei DGET BF/PF: KCDPID enthält keine gültige DPUT-ID.
KB18	KCSDGET	Bei DGET FT/NT: KCMF/kcfn enthält keine Leerzeichen.
KB19	KCSDGET	Bei DGET NT/BN/PN (next): - KCOM passt nicht zum vorigen DGET-Aufruf oder - es wurde in diesem Programmablauf noch kein DGET FT/BF/PF (first) gegeben oder - es wurde dazwischen ein PGWT aufgerufen.
KB20	KCSDGET	Bei DGET NT/BN/PN: Seit dem letzten DGET-Aufruf wurde die DGET-Queue gelöscht und neu erzeugt.
KB21	KCSDGET	Bei DGET BN/PN (next): Seit dem letzten DGET-Aufruf wurde der in KCRN und KCQTYP angegebene USER bzw. die Temporäre Queue gelöscht und neu erzeugt.
KB22	KCSDGET	Bei DGET BN: Es existiert keine Nachricht mit der in KCGTM angegebenen Erzeugungszeit oder diese wurde inzwischen verarbeitet.
KB23	KCSDGET	Bei DGET PF: Es existiert keine Nachricht mit der in KCGTM angegebenen Erzeugungszeit oder diese wurde inzwischen verarbeitet.
KB24	KCSDGET	Bei DGET BF/PF: KCDPID passt nicht zu den Angaben in KCRN und KCQTYP.
KB25	KCSDGET	Im Anmeldevorgang sind DGET-Aufrufe mit Warten nicht erlaubt.
KB26	KCSDGET	Es wurde versucht, mit DGET FT/NT/PF/PN von der Dead Letter Queue KDCDLETQ zu lesen.
KC01	KCDCTRL	VTV ist nicht generiert.
KC02	KCDCTRL	Das erste Zeichen der in KCRN angegebenen Vorgangs-ID (VGID) ist nicht '>'.
KC03	KCDCTRL	Der Aufruf wurde für einen Asynchronvorgang gegeben, d.h die in KCRN angegebene VGID wurde mit einem APRO AM Aufruf definiert.
KC04	KCDCTRL	Die in KCRN angegebene VGID ist ungültig.
KC06	KCDCTRL	Der CTRL-Aufruf wurde an einen Partner gerichtet, mit dem nicht über das OSI-TP Protokoll kommuniziert wird.

Code	Modul	Fehlerursache und ggf. Maßnahme
KC07	KCDCTRL	CTRL PR wurde an einen Partner gerichtet, für den die Commit Functional Unit nicht ausgewählt wurde.
KC08	KCDCTRL	CTRL PE wurde an einen Partner gerichtet, für den die Commit Functional Unit nicht ausgewählt wurde.
KC09	KCDCTRL	CTRL AB wurde an einen Partner gerichtet, an den zuvor mit MPUT eine Nachricht geschickt wurde.
KC10	KCDCTRL	Die Operationsmodifikation OM ist weder PR, noch PE oder AB.
KC11	KCDCTRL	KCLA/kcla ist nicht Null.
KC12	KCDCTRL	KCLM/kclm ist nicht Null.
KC13	KCDCTRL	KCMF/kcfn ist nicht mit Spaces (Leerzeichen) versorgt.
KC14	KCDCTRL	KCDF/kcdf ist nicht mit binär null versorgt.
KC15	KCDCTRL	Der erweiterte Parameterbereich (EXTENT) ist nicht mit binär null versorgt.
KC16	KCDCTRL	Der Aufruf ist an einen Partner gerichtet, an den zuvor ein MPUT HM abgesetzt wurde.
KC17	KCDCTRL	Der Aufruf CTRL PR ist in einem mittleren Knoten an einen Partner gerichtet, ohne dass der lokale Vorgang zuvor einen Prepare von seinem Auftraggeber erhalten hat.
KC18	KCDCTRL	Der Aufruf CTRL PE ist in einem mittleren Knoten an einen Partner gerichtet, ohne dass der lokale Vorgang zuvor einen Prepare von seinem Auftraggeber erhalten hat.
KC19	KCDCTRL	Der Aufruf CTRL AB wurde an einen Partner gerichtet, für den die Commit Functional Unit nicht ausgewählt wurde und der Vorgangstatus ist nicht O
KC20	KCDCTRL	Der Aufruf CTRL AB wurde an einen Partner gerichtet, für den die Commit Functional Unit ausgewählt wurde und der Vorgangstatus ist weder O noch C.
KC21	KCDCTRL	Der Aufruf CTRL PR ist nicht erlaubt, weil der Vorgangstatus nicht O ist.
KC22	KCDCTRL	Der Aufruf CTRL PE ist nicht erlaubt, weil der Vorgangstatus nicht O ist.
KC23	KCDCTRL	Der Aufruf CTRL PR ist nicht erlaubt, weil der Transaktionsstatus P ist.
KC24	KCDCTRL	Der Aufruf CTRL PE ist nicht erlaubt, weil der Transaktionsstatus P ist.
KC25	KCDCTRL	Der nicht verwendete Teil des erweiterten Parameterbereich (EXTENT) ist nicht mit binär Null versorgt.
KC26	KCDCTRL	Das Feld KCNORPLY hat weder den Wert Y noch binär Null.
KD00	KCDAPRO	KCRN enthält keine gültige Vorgangsidentifikation (MGET, MPUT, FPUT, DPUT)
KD01	KCDAPRO	LTAC ist nicht definiert
KD02	KCDAPRO	LTAC darf nicht benutzt werden, LTAC ist gesperrt

Code	Modul	Fehlerursache und ggf. Maßnahme
KD03	KCDAPRO	Für das Schloss des LTACs hat der Benutzer keinen Schlüssel an seinem Schlüsselbund
KD04	KCDAPRO	Es wurde ein APRO DM aufgerufen, aber der LTAC ist als Asynchron LTAC generiert, bzw. APRO AM und LTAC als Dialog LTAC generiert
KD05	KCDAPRO	Es wurden mit APRO mehr Auftragnehmervorgänge erzeugt als bei der Generierung angegeben wurde. (KDCDEF Generierung: UTMD MAXJR=)
KD06	KCDAPRO	MPUT an AN-Vorgang und KCDF ungleich binär 0
KD08	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
KD09	KCDAPRO	Es wurde ein APRO DM aufgerufen, aber zur entfernten Anwendung wurden die logischen Verbindungen mit dem Administrationskommando KDCLPAP ACT=QUIET abgebaut.
KD10	KCDAPRO	Es wurde ein APRO DM aufgerufen, aber zur entfernten Anwendung ist keine logische Verbindung aufgebaut.
KD11	KCDAPRO	Es wurde ein APRO DM aufgerufen, aber die entfernte Anwendung ist als "contention winner" generiert und die Wartezeit für Sessionbelegung ist 0 (nur bei LU6.1).
KD12	KCDAPRO	Nach einem MPUT NE/HM an einen AN-Vorgang wurde ein weiterer MPUT an den AN-Vorgang gegeben.
KD13	KCDAPRO	Bei einem FPUT (DPUT mit KCMOD=' ') an einen Auftragnehmer-Vorgang besteht zur entfernten Anwendung keine Verbindung und die Wartezeit für die Belegung einer Session bzw. Association ist 0.
KD14	KCDAPRO	Ein asynchroner Vorgang in einer anderen Anwendung soll mit APRO adressiert werden. An diese Anwendung liegen bereits so viele asynchrone Nachrichten vor, wie bei der Generierung als Maximalwert festgelegt wurde. (LPAP QLEV= bzw. OSI-LPAP QLEV=)
KD15	KCDAPRO	Bei einem MPUT EM oder MPUT HM ist das Ziel in KCRN kein OSI TP Kommunikationspartner.
KD16	KCDAPRO	Vor einem MPUT HM wurde kein MPUT NT gegeben.
KD17	KCDAPRO	Der RTAC wurde für einen OSI-TP-Partner generiert.
KD18	KCDAPRO	Eine ausgewählte Functional Unit wird von dieser UTM Version nicht unterstützt.
KD22	KCDAPRO	Es wurde die Commit Functional Unit ausgewählt, aber für den Partner ist die abstract syntax CCR nicht generiert.
KD23	KCDAPRO	Mischbetrieb von LU6.1 und OSI-TP innerhalb einer verteilten Transaktion
KD24	KCDAPRO	Mischbetrieb von LU6.1 und OSI-TP innerhalb einer verteilten Transaktion

Code	Modul	Fehlerursache und ggf. Maßnahme
KD25	KCDAPRO	Es wird bei einer verteilten Transaktion mit OSI-TP mehr als ein ACCESS-POINT benutzt
KD26	KCDAPRO	Es wird bei einem OSI-TP-Partner eine nicht generierte abstrakte Syntax bei MPUT/FPUT/DPUT in KCMF/kcfn angegeben. Die Syntaxnamen "CCR" und "OSITP" sind nicht zulässig
KD27	KCDAPRO	KCLM/kclm muss 32 sein.
KD28	KCDAPRO	KCLM/kclm muss Null sein.
KD29	KCDAPRO	KCLM/kclm muss gleich der Länge der Datenstruktur wie in COPY KCAPROC resp. Include kcapro.h sein.
KD30	KCDAPRO	Ungültige Werte in APRO Datenbereich
KD31	KCDAPRO	Bei KCFUCOM = 'N' muss KCFUCHN Leerzeichen enthalten
KD32	KCDAPRO	Es wurde Security Type Same oder Program ausgewählt, aber für den Partner ist die abstract syntax UTMSEC nicht generiert.
KD33	KCDAPRO	Bei Security Type Program wurde eine ungültige Länge für Userid oder Password angegeben.
KD34	KCDAPRO	Fehler beim Encodieren der Security Daten
KD35	KCDAPRO	KCFUHS = 'Y' ist nur bei Dialogpartner erlaubt
KD36	KCDAPRO	Nach einem CTRL PR oder PE wurde ein MPUT HM an den gleichen Partner gegeben.
KD37	KCDAPRO	Nach einem CTRL AB wurde ein MPUT an den gleichen Partner gegeben.
KD38	KCDAPRO	MPUT an Auftraggeber, aber KCSEND = NO.
KD39	KCDAPRO	Die nicht benutzten Felder für die Security-Funktion bei KCSEC TYP ungleich 'P' sind nicht gelöscht. (CHAR-Felder auf Blank, numerische Felder auf 0).
KD40	KCDAPRO	Dem OSI-LPAP beim APRO-Aufruf ist keine aktive Verbindung zugeordnet.
KD41	KCDAPRO	Das Master-LPAP ist gesperrt
KD42	KCSPEND	(Siehe K348) Der Cluster Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF; Cluster Pagepool mit CLUSTER PGPOOL=(anzahl,...) größer einrichten.
KE01	KCSPADM	Berechtigung für Aufruf fehlt: Der Benutzer ist kein Administrator und das Terminal ist keine Druckersteuerstation bzw. stimmt nicht mit der im KDCS-Parameterbereich angegebenen (KCLT/kclt) Datenstation überein.
KE02	KCSPADM	Der Druckersteuerstation ist kein Drucker (PTERM) zugeordnet. Generierung überprüfen!
KE03	KCSPADM	Ein PADM-Aufruf zur Druckbestätigung oder zur Druckwiederholung (KCOM=OK/PR) erfolgte, obwohl keine Druckausgabe für den angegebenen Drucker zu bestätigen war.

Code	Modul	Fehlerursache und ggf. Maßnahme
KE04	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KE05	KCSPADM	Ein Drucker soll einem anderen LTERM zugeordnet werden (KCOM=CA), ist aber mit der Anwendung verbunden.
KE06	KCSPADM	Verbindung zu einem Drucker soll aufgebaut werden, der Drucker ist aber gesperrt.
KE07	KCSPADM	Ein Inverser KDCDEF läuft oder soll gestartet werden.
KE08	KCSPADM	Die Cluster-globale Administration ist zur Zeit nicht möglich, da Knoten-Anwendungen mit unterschiedlicher Generierung laufen.
KF01	KCSINFO	Der Benutzer ist ohne Ausweiskarte generiert
KF02	KCSINFO	Die Information ist nicht mehr verfügbar, z.B. wegen Verbindungsverlust (INFO CD).
KF05	KCSINFO	Der Vorgang läuft nicht unter einer "echten" Benutzerkennung.
KF06	KCSINFO	Fehler beim Verschlüsseln des Passworts
KG01	KCSDADM	Berechtigung für Aufruf fehlt: Der Benutzer ist kein Administrator und das Terminal ist keine Druckersteuerstation bzw. stimmt nicht mit dem im KDCS-Parameterbereich angegebenen (KCLT/kclt) Terminal überein.
KG02	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KG03	KCSDADM	Ein DADM-Aufruf zum Löschen aller freilaufenden Nachrichten (KCOM=DA) erfolgte, obwohl für das angegebene Ziel Nachrichten in Bearbeitung waren.
KG04	KCSDADM	Es wurde versucht eine in Bearbeitung befindliche Nachricht zu administrieren.
KG05	KCSDADM	Nach einem Aufruf zum Löschen von Nachrichten (KCOM=DL/DA) erfolgt ein weiterer Löschauftrag bzw. ein Auftrag zum Umketten einer Nachricht (KCOM=DL/DA/CS).
KG06	KCSDADM	Es wurde versucht eine zeitgesteuerte Nachricht umzuketten (KCOM=CS), obwohl deren Startzeitpunkt noch nicht erreicht war.
KG07	KCSDADM	Ein Aufruf zum Lesen von Informationen über Nachrichten (KCOM=RQ) erfolgte, obwohl für das angegebene Ziel keine (oder nur in Bearbeitung befindliche) Nachrichten vorhanden waren.
KG08	KCSDADM	KCQTYP ungültig

Code	Modul	Fehlerursache und ggf. Maßnahme
KG09	KCSDADM	Bei DADM MV mit Leerzeichen in KCLT/kclt ist das ursprüngliche Ziel der Nachricht in der Dead Letter Queue nicht mehr vorhanden. Dieser Nachricht muss ein neues Ziel zugeordnet werden.
KG10	KCSDADM	Bei DADM MV ist das in KCLT/kclt angegebene Ziel nicht mehr gültig.
KH01	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KH02	KCSSIGN	Bei SIGN CP: Das neue Passwort genügt nicht der geforderten Komplexitätsstufe
KH03	KCSSIGN	Bei SIGN CP: Das neue Passwort ist zu kurz
KH04	KCSSIGN	Bei SIGN CP: Das neue Passwort ist identisch dem alten Passwort oder mit einem Passwort aus der Passwort-Historie
KH08	KCSSIGN	Ein Inverser KDCDEF läuft oder soll gestartet werden.
KH09	KCSSIGN	Bei SIGN CP: Das Passwort darf noch nicht geändert werden, da die minimale Gültigkeitsdauer noch nicht überschritten ist.
KH10	KCSSIGN	Bei SIGN CP: Für einen Benutzer, der mit Zertifikat generiert ist, darf kein Passwort vergeben bzw. geändert werden.
KH11	KCSSIGN	Der Vorgang wurde abnormal beendet, weil beim Abmelden des Benutzers die Pagelist für die Cluster Vorgangs-Daten nicht geschrieben werden konnte, weil der Cluster Pagepool voll war.
KH12	KCSSIGN	Das Fortsetzen eines offenen Cluster-Vorgangs ist nicht möglich, da durch eine Neugenerierung der Knoten-KDCFILE die Reihenfolge der TAC-Anweisungen geändert wurde oder die Eigenschaften des Folge-TAC geändert wurden oder der Folge-TAC oder der Vorgangs-TAC nicht mehr existieren.
KJ01	KCCUSF	Der Vorgang wurde abnormal beendet, weil er durch eine andere Knoten-Anwendung zum abnormalen Beenden markiert wurde. Dies kann administrativ erfolgt sein oder dadurch, dass sich der User trotz eines an diese Knoten-Anwendung gebundenen Vorgangs an eine andere Knoten-Anwendung angemeldet hat.
KK01	KCSSVCS	Der zweite Parameter des KDCS-Aufrufs fehlt.
KK02	KCSSVCS	Die Adressvalidierung des zweiten Parameters des KDCS-Aufrufs ist fehlgeschlagen.
KK03	KCSSVCS	Der KDCS-Aufruf erfolgte aus einem Teilprogramm eines Asynchron-Vorgangs.
KK04	KCSSVCS	Der KDCS-Operationscode ist unbekannt.
KK07	KCSSVCS	Der KDCS-Operationscode APRO ist nicht erlaubt.
KK08	KCSSVCS	Der Aufruf ist in UTM-Cluster-Anwendungen nicht erlaubt.

Code	Modul	Fehlerursache und ggf. Maßnahme
KL00	KCSSTRT	Beim Warmstart der Anwendung wurde keine Auftragnehmer-Session gefunden, die im PET-Zustand ist.
KL01	KCSSTRT	Beim Warmstart der Anwendung wurde für den OSI TP Vorgang kein gültiger NODE-Tabelleneintrag gefunden.
KL02	KCSSTRT	Der Benutzer, der diesen Vorgang gestartet hat, wurde gelöscht.
KL03	KCSSTRT	Beim Warmstart wurde festgestellt, dass der Vorgang nicht fortgesetzt werden kann. OSI TP Auftragnehmer-Vorgänge werden ggfs. erst nach dem Association-Aufbau zum OSI TP Partner und anschließender OSI TP Recovery beendet.
KL04	KCSSTRT	PEND ER veranlasst durch KCSSTRT.
KL05	KCSSTRT	PEND ER veranlasst durch KCSSTRT.
KL06	KCSSTRT	PEND ER veranlasst durch KCSSTRT.
KL07	KCSSTRT	PEND ER veranlasst durch KCSSTRT.
KL08	KCSSTRT	PEND ER veranlasst durch KCSSTRT.
KM01	KCSTRHD	TAC ist nicht definiert. ¹
KM02	KCSTRHD	TAC darf nicht benutzt werden, der TAC ist gesperrt. ¹
KM03	KCSTRHD	Für das Schloss des TACs gibt es keinen Schlüssel im Schlüsselbund der LTERM- oder (OSI-)LPAP-Partner. ¹
KM04	KCSTRHD	Für das Schloss des TACs hat der Benutzer keinen Schlüssel an seinem Schlüsselbund. ¹
KM05	KCSTRHD	Der TAC ist als Administrations-TAC generiert, aber der Benutzer ist kein Administrator. ¹
KM07	KCSTRHD	Für den Transaktionscode ist das zugehörige Teilprogramm nicht eingebunden oder konnte nicht geladen werden. ¹
KM08	KCSTRHD	Der Transaktionscode ist vollständig gesperrt. Er darf weder in einem PEND-Aufruf als Folge-TAC angegeben werden, noch können neue Aufträge für diesen TAC gegeben werden. ¹
KM09	KCSTRHD	Der Transaktionscode ist durch einen Encryptionlevel geschützt und der Aufrufer unterstützt diesen Encryptionlevel nicht oder die Eingabenachricht wurde nicht dem Level entsprechend verschlüsselt.
KM99	KCSTRHD	Ein TAC, der mit XOPEN-API generiert ist, darf nur von einem Auftraggeber verwendet werden, mit dem über das OSI TP, LU6.1- oder das UPIC-Protokoll kommuniziert wird.
KN01	KCSDLO	Der Benutzer, der den Vorgang gestartet hat, wurde gelöscht.
KQ01	KCSCRO	Beim Aufruf QCRE WN beginnt der Name in KCRN mit einer Ziffer.
KQ02	KCSCRO	Beim Aufruf QCRE NN wurde KCRN nicht mit Leerzeichen versorgt.

Code	Modul	Fehlerursache und ggf. Maßnahme
KQ03	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KQ04	KCSCRO	Die bei der Generierung mit der QUEUE-Anweisung reservierten Tabellen für QUEUE-Objekte sind aufgebraucht. Maßnahme: Nicht mehr benötigte QUEUE-Objekte löschen oder neu generieren.
KQ05	KCSCRO	Beim Aufruf QCRE WN enthält KCRN ein ungültiges Zeichen oder ist nicht mit Leerzeichen aufgefüllt.
KR01	KDCRTMM	Es fehlt ein Dialog-Teilprogramm (nicht gebunden), mit dem ein Vorgang fortgesetzt werden sollte. ¹
KR02	KDCRTMM	Es fehlt ein Asynchron-Teilprogramm (nicht gebunden) , für das eine Nachricht auf Verarbeitung wartet. ¹
KR04	KDCRTMM	Ein Teilprogrammmlauf hat vor dem letzten KDCS-Aufruf über das Ende des KB hinausgeschrieben. Maßnahme: Teilprogramm oder Generierung ändern.
KR05	KDCRTMM	Ein Teilprogrammmlauf hat vor dem letzten KDCS-Aufruf über das Ende des SPAB hinausgeschrieben. Maßnahme: Teilprogramm oder Generierung ändern.
KR06	KDCRTMM	Ein VORGANG-Exit hat einen KDCS-Aufruf gegeben. Der Vorgang wird mit PEND-ER abgebrochen.
KR09	KDCRTMM	Fehler beim Nachladen eines Teilprogramms. Der Vorgang wird mit PEND-ER abgebrochen.
KR10	KDCRTMM	Programm für Vorgangs-Exit fehlt. Der Vorgang wird mit PEND-ER abgebrochen.
KS00	KCDSEFU	Bei einer Transaktionsendanforderung wurden MPUTs an mehrere Partner gegeben.
KS01	KCDSEFU	Bei einer Transaktionsendanforderung hat mindestens ein Partner einen nicht erlaubten Transaktions- oder Vorgangstatus.
KS02	KCDSEFU	Bei einer Transaktionsendanforderung ist mindestens eine Session nicht benutzt worden (zuviele APRO Aufrufe).
KS03	KCDSEFU	Vom Auftragsgeber wurde MPUT an eine VGID gegeben, danach wurde die Transaktion mit PEND RE beendet, obwohl alle Auftragnehmer nicht PEND RE/FI gemeldet haben.
KS04	KCDSEFU	Zu einem LU6.1-Partner, der schon das Transaktionsende eingeleitet hat, wurde eine Nachricht mit PEND KP geschickt.

Code	Modul	Fehlerursache und ggf. Maßnahme
KS05	KCDSEFU	Ist die Session sowohl Auftragnehmer als auch Auftraggeber, darf sie an ihren Auftragnehmer erst dann PEND RE melden, wenn ihr Auftraggeber ebenfalls PEND RE gemeldet hat.
KS06	KCDSEFU	Nach einem APRO AM wurde keine freilaufende Nachricht für den adressierten Vorgang ausgegeben
KS07	KCDSEFU	Es wurde PEND SP gegeben obwohl es mindestens einen Partner mit offener Transaktion gibt.
KS08	KCDSEFU	Der OSI-TP-Client sendet eine Nachricht an den OSI-TP-Server, ohne die Transaktion zu beenden, oder fordert den OSI-TP-Server auf, die Transaktion zu beenden, obwohl der Server das Transaktionsende bereits eingeleitet hat.
KS09	KCDSEFU	Der OSI-TP-Server hat die Transaktion im Widerspruch zur Forderung des OSI-TP-Client nicht beendet.
KS10	KCDSEFU	Der OSI-TP-Server hat die Transaktion im Widerspruch zur Forderung des OSI-TP-Clients mit PEND SP beendet.
KS11	KCDSEFU	Der OSI-TP-Client sendet eine Nachricht an den OSI-TP-Server und beendet die Transaktion, obwohl er das Senderecht zum Transaktionsende nicht besitzt, da er auf einen anderen Dialog selbst OSI-TP-Server ist und sein OSI-TP-Client das Senderecht zum Transaktionsende nicht abgegeben hat.
KS12	KCDSEFU	Der OSI-TP-Server hat die Transaktion im Widerspruch zur Forderung des OSI-TP-Clients mit PEND RE beendet.
KS13	KCDSEFU	Der OSI-TP-Client beendet den Vorgang mit PEND FI obwohl er noch eine offene Server-Conversation ohne COMMIT-Funktionalität hat.
KS14	KCDSEFU	Der OSI-TP-Server hat den Vorgang im Widerspruch zur Forderung des OSI-TP-Clients mit PEND FI beendet.
KS15	KCDSEFU	Der OSI-TP-Client beendet den Vorgang mit PEND FC, obwohl er noch eine offene Server-Conversation ohne COMMIT-Funktionalität hat.
KS16	KCDSEFU	Der OSI-TP-Server sendet eine Nachricht an den OSI-TP-Client, obwohl er das Senderecht auf diesem Dialog nicht besitzt.
KS17	KCDSEFU	Der OSI-TP-Server beendet den Dialogschritt, sendet aber keine Nachricht an den OSI-TP-Client, obwohl er das Senderecht auf diesem Dialog besitzt.
KS18	KCDSEFU	Der OSI-TP-Client gibt das Senderecht zum Transaktionsende an mehr als einen OSI-TP-Server ab.
KS19	KCDSEFU	Der OSI-TP-Client fordert den OSI-TP-Server mit CTRL PR oder PE zum Transaktions- oder Vorgangsende auf, sendet eine Nachricht an diesen Partner und leitet dann selbst Transaktionsende ein.
KS20	KCDSEFU	Der lokale Vorgang hat PGWT CM oder PGWT RB aufgerufen, obwohl ein Partner an der verteilten Transaktion beteiligt ist, mit dem über das LU6.1-Protokoll kommuniziert wird.
KS21	KCDSEFU	Der OSI-TP-Server hat die Transaktion im Widerspruch zur Forderung des OSI-TP-Clients mit PGWT CM beendet.

Code	Modul	Fehlerursache und ggf. Maßnahme
KS22	KCDSEFU	Der OSI-TP-Client beendet den Vorgang mit PEND FI obwohl er eine Server-Conversation mit CTRL PR nur zum Transaktionsende aufgefordert hat.
KS23	KCDSEFU	Der OSI-TP-Client beendet den Vorgang mit PEND FC obwohl er eine Server-Conversation mit CTRL PR nur zum Transaktionsende aufgefordert hat.
KS24	KCDSEFU	Der OSI-TP-Client fordert eine Server-Conversation, an die noch keine Nachricht gesendet wurde, mit CTRL PR/PE zum Transaktions-/Vorgangsende auf.
KS25	KCDSEFU	Der Session-Wiederanlauf ist fehlgeschlagen.
KT01	KCSPEND	(Siehe K302) Der taskspezifische Pufferbereich für Wiederanlaufinformation ist zu klein (siehe UTM-Handbuch Anwendungen generieren: Wiederanlaufbereich). Maßnahme: Generierung mit KDCDEF, Pufferbereich mit MAX RECBUF=(...,länge) größer definieren. ¹
KT02	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
KT03	KCSPEND	(Siehe K306) Die Transaktion und der Vorgang mussten zurückgesetzt werden, weil das DB-System die DB-Transaktion beim Schließen zurückgesetzt hat.
KT04	KCDOSTM	ein Log-Record passt nicht in den Eingabebuffer. Maßnahme: weniger OSI-TP Auftragnehmer in dem Vorgang adressieren. ¹
KT05	KCDOSTM	Die Transaktion muss zurückgesetzt werden, weil die Datenbanktransaktion zurückgesetzt wurde.
KT06	KCDOSTM	Von der Datenbank wurde beim Transaktionsbeenden nach Prepare gemeldet, dass die Transaktion zurückgesetzt wurde.
KT07	KCDOSTM	Von einem OSI TP Server wurde eine heuristische Entscheidung gemeldet (heuristic mixed).
KT08	KCDOSTM	Von einem OSI TP Server wurde eine heuristische Entscheidung gemeldet (heuristic hazard).
KU04	KDCRTDB	Die DB-Transaktion musste zurückgesetzt werden. UTM setzt auch die UTM-Transaktion zurück: Der Teilprogrammmlauf erhält wieder die Kontrolle, wie beim UTM-Aufruf RSET.
KU08	KDCRTDB	UTM muss den Vorgang mit PEND ER beenden. Der Terminalbenutzer erhält eine Meldung über den Abbruch des Vorgangs. Die ersten 4 Bytes der DB-Fehlermeldung werden mit angezeigt.
KU0C	KDCRTDB	Das DB-System (der DBH) ist nicht oder nicht mehr verfügbar. UTM beendet die Anwendung abnormal bzw. bricht den Start der Anwendung ab.

Code	Modul	Fehlerursache und ggf. Maßnahme
KU10	KDCRTDB	Das DB-System ist als Folge einer DB-Administrator-Aktion nicht mehr verfügbar. UTM beendet die Anwendung abnormal.
KU14	KDCRTDB	Das DB-System ist zur Zeit nicht verfügbar. Nach 5 Sekunden wird ein neuer Versuch zum Verbindungsaufbau gemacht. ¹
KU18	KDCRTDB	Ein eventuell behebbarer Fehler im DB-System liegt vor.
KU1C	KDCRTDB	Ein nicht behebbarer Fehler im DB-System liegt vor. UTM beendet die Anwendung abnormal.
KU20	KDCRTDB	Das DB-System hat einen Benutzerfehler entdeckt. Der Fehler tritt beim Start der Anwendung auf, wenn die DB-spezifischen Startparameter geprüft werden. Die DB-Fehlermeldung wird auf SYSOUT ausgegeben.
KU24	KDCRTDB	Aus Sicht des DB-Systems trat ein unerwartetes Verhalten von UTM auf. Die Anwendung wird mit PEND06 beendet.
KV01	KCSTUTM	Socketpartner sendet mit USP Version 1.0. Puffer für Dialognachrichten zu klein generiert, bzw. MPUT Länge hat maximal Größe überschritten. Maßnahme: Generierung ändern, Puffer mit MAX NB=länge größer wählen.
KV02	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
KV03	KCSPEND	(Siehe K348) Der Cluster Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF; Cluster Pagepool mit CLUSTER PGPOOL=(anzahl,...) größer einrichten.
KW01	KCDOSSF	Empfang von TP-HANDSHAKE-IND.
KW02	KCDOSSF	Empfang von TP-BEGIN-DIALOGUE-CNF (negativ).
KW03	KCDOSSF	Empfang von TP-U-ERROR-IND von einem Partner, der nicht das Senderecht hat.
KW04	KCDOSSF	Ablauf eines Timers.
KW05	KCDOSSF	Empfang von TP-U-ABORT-IND oder TP-P-ABORT-IND.
KW06	KCDOSSF	Empfang von TP-ROLLBACK-IND.
KW07	KCDOSSF	Empfang von TP-END-DIALOGUE-IND vom Client wird von UTM nicht unterstützt.
KW08	KCDOSSF	Bei der Unchained Transactions Funktionseinheit wurde der Dialog mit dem Client nicht mit der 1. Transaktion beendet.
KY00	KCDSYPM	Vom Partner wurde eine negative Quittung erhalten.
KY01	KCDSYPM	Timeout beim Belegen einer Session.
KY02	KCDSYPM	Der Auftragnehmer-Vorgang wurde mit PEND ER beendet.
KY03	KCDSYPM	Der Auftraggeber-Vorgang wurde mit PEND ER beendet.

Code	Modul	Fehlerursache und ggf. Maßnahme
KY04	KCDSYPM	Mismatch mit dem Partner, und der lokale Vorgang ist im PET-Zustand.
KY05	KCDSYPM	Mismatch mit dem Partner.
KY06	KCDSYPM	Mismatch mit dem Datenbank.
KY07	KCDSYPM	Verbindungsverlust zum Partner.
NOTA	KCDRTDB	Transaktion ist beim XA-Resource Manager unbekannt. (Kann z.B. nach Timeouts mit der ORACLE-Datenbank auftreten).
UPCP	KCSPEND	(Siehe K348) Der Cluster Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF; Cluster Pagepool mit CLUSTER PGPOOL=(anzahl,...) größer einrichten.
UPDE	KCSUPIC	Es wurde Vorgangsabbruch vom UPIC-Partner angefordert.
UPPG	KCSPEND	(Siehe K345) Der Pagepool in KDCFILE ist voll. Maßnahme: Generierung mit KDCDEF, Pagepool mit MAX PGPOOL=(anzahl,...) größer einrichten. ¹
UPRS	KCSUPIC	Der UPIC Partner hat einen Vorgangswiederanlauf unterdrückt. Dieser Fall tritt dann auf, wenn ein Vorgangswiederanlauf möglich gewesen wäre, aber vom UPIC-Partner nicht angefordert wird. Dann setzt UTM den Vorgang zurück.
UPXC	KCSUPIC	In einem Teilprogramm mit X/OPEN-API soll die Nachricht an den UPIC-Auftraggeber abgeschlossen werden, obwohl schon eine Nachricht an einen Auftragnehmervorgang abgeschlossen wurde.
UPXM	KCSUPIC	In einem Teilprogramm mit X/OPEN-API soll die Nachricht an den UPIC-Auftraggeber abgeschlossen werden, obwohl schon eine Nachricht an den UPIC-Auftraggeber abgeschlossen wurde.
XTnn	KDCROOT	Es wurde die Signalroutine in KDCROOT aufgerufen, nn = Signalnummer (siehe headerfile signal.h) Falls nn der Wert von SIGLRM ist, wurde die maximal zulässige Programmlaufzeit überschritten. (siehe KDCDEF-Generierung TAC RTIME= <zeit>)

¹ Bei TESTMODE=OFF werden diese PENDER Dumps unterdrückt. Gleichzeitig unterbleibt das Nachladen, es sei denn für den aktuellen TAC sind PGWT-Aufrufe erlaubt und der KDCS-Aufruf, bei dem der Fehler auftrat, war kein PEND-Aufruf.

Fachwörter

Fachwörter, die an anderer Stelle erklärt werden, sind mit *kursiver* Schrift ausgezeichnet.

Ablaufinvariantes Programm

siehe *reentrant-fähiges Programm*.

Abnormale Beendigung einer UTM-Anwendung

Beendigung einer *UTM-Anwendung*, bei der die *KDCFILE* nicht mehr aktualisiert wird. Eine abnormale Beendigung wird ausgelöst durch einen schwerwiegenden Fehler, z.B. Rechnerausfall, Fehler in der Systemsoftware. Wird die Anwendung erneut gestartet, führt openUTM einen *Warmstart* durch.

abstrakte Syntax (OSI)

Eine abstrakte Syntax ist die Menge der formal beschriebenen Datentypen, die zwischen Anwendungen über *OSI TP* ausgetauscht werden sollen. Eine abstrakte Syntax ist unabhängig von der eingesetzten Hardware und der jeweiligen Programmiersprache.

Access-List

Eine Access-List definiert die Berechtigung für den Zugriff auf einen bestimmten *Service*, auf eine bestimmte *TAC-Queue* oder auf eine bestimmte *USER-Queue*. Eine Access-List ist als *Keyset* definiert und enthält einen oder mehrere *Keycodes*, die jeweils eine Rolle in der Anwendung repräsentieren. Benutzer, LTERMs oder (OSI-)LPAPs dürfen nur dann auf den Service oder die *TAC-Queue/USER-Queue* zugreifen, wenn ihnen die entsprechenden Rollen zugeteilt wurden, d.h. wenn ihr *Keyset* und die Access-List mindestens einen gemeinsamen *Keycode* enthalten.

Access Point (OSI)

siehe *Dienstzugriffspunkt*.

ACID-Eigenschaften

Abkürzende Bezeichnung für die grundlegenden Eigenschaften von *Transaktionen*: Atomicity, Consistency, Isolation und Durability.

Administration

Verwaltung und Steuerung einer *UTM-Anwendung* durch einen *Administrator* oder ein *Administrationsprogramm*.

Administrations-Journal

siehe *Cluster-Administrations-Journal*.

Administrationskommando

Kommandos, mit denen der *Administrator* einer *UTM-Anwendung* Administrationsfunktionen für diese Anwendung durchführt. Die Administrationskommandos sind als *Transaktionscodes* realisiert.

Administrationsprogramm

Teilprogramm, das Aufrufe der *Programmschnittstelle für die Administration* enthält. Dies kann das Standard-Administrationsprogramm *KDCADM* sein, das mit *openUTM* ausgeliefert wird, oder ein vom Anwender selbst erstelltes Programm.

Administrator

Benutzer mit Administrationsberechtigung.

AES

AES (Advanced Encryption Standard) ist der aktuelle symmetrische Verschlüsselungsstandard, festgelegt vom NIST (National Institute of Standards and Technology), basierend auf dem an der Universität Leuven (B) entwickelten Rijndael-Algorithmus. Wird das AES-Verfahren verwendet, dann erzeugt der UPIC-Client für jede Sitzung einen AES-Schlüssel.

Akzeptor (CPI-C)

Die Kommunikationspartner einer *Conversation* werden *Initiator* und Akzeptor genannt. Der Akzeptor nimmt die vom Initiator eingeleitete *Conversation* mit *Accept_Conversation* entgegen.

Anmelde-Vorgang (KDCS)

Spezieller *Dialog-Vorgang*, bei dem die Anmeldung eines Benutzers an eine *UTM-Anwendung* durch *Teilprogramme* gesteuert wird.

Anschlussprogramm

siehe *KDCROOT*.

Anwendungsinformation

Sie stellt die Gesamtmenge der von der *UTM-Anwendung* benutzten Daten dar. Dabei handelt es sich um Speicherbereiche und Nachrichten der *UTM-Anwendung*, einschließlich der aktuell auf dem Bildschirm angezeigten Daten.

Arbeitet die UTM-Anwendung koordiniert mit einem Datenbanksystem, so gehören die in der Datenbank gespeicherten Daten ebenfalls zur Anwendungsinformation.

Anwendungs-Kaltstart

siehe *Kaltstart*.

Anwendungsprogramm

Ein Anwendungsprogramm bildet den Hauptbestandteil einer *UTM-Anwendung*. Es besteht aus der Main Routine *KDCROOT* und den *Teilprogrammen*. Es bearbeitet alle Aufträge, die an eine *UTM-Anwendung* gerichtet werden.

Anwendungs-Warmstart

siehe *Warmstart*.

Apache Axis

Apache Axis (Apache eXtensible Interaction System) ist eine SOAP-Engine zur Konstruktion von darauf basierenden Web Services und Client-Anwendungen. Es existiert eine Implementierung in C++ und Java.

Apache Tomcat

Apache Tomcat stellt eine Umgebung zur Ausführung von Java-Code auf Web-Servern bereit, die im Rahmen des Jakarta-Projekts der Apache Software Foundation entwickelt wird. Es handelt sich um einen in Java geschriebenen Servlet-Container, der mithilfe des JSP-Compilers Jasper auch JavaServer Pages in Servlets übersetzen und ausführen kann. Dazu kommt ein kompletter HTTP-Server.

Application Context (OSI)

Der Application Context ist die Menge der Regeln, die für die Kommunikation zwischen zwei Anwendungen gelten sollen. Dazu gehören z.B. die *abstrakten Syntaxen* und die zugeordneten *Transfer-Syntaxen*.

Application Entity (OSI)

Eine Application Entity (AE) repräsentiert alle für die Kommunikation relevanten Aspekte einer realen Anwendung. Eine Application Entity wird durch einen global (d.h. weltweit) eindeutigen Namen identifiziert, den *Application Entity Title* (AET). Jede Application Entity repräsentiert genau einen *Application Process*. Ein Application Process kann mehrere Application Entities umfassen.

Application Entity Title (OSI)

Ein Application Entity Title ist ein global (d.h. weltweit) eindeutiger Name für eine *Application Entity*. Er setzt sich zusammen aus dem *Application Process Title* des jeweiligen *Application Process* und dem *Application Entity Qualifier*.

Application Entity Qualifier (OSI)

Bestandteil des *Application Entity Titles*. Der Application Entity Qualifier identifiziert einen *Dienstzugriffspunkt* innerhalb der Anwendung. Ein Application Entity Qualifier kann unterschiedlich aufgebaut sein. openUTM unterstützt den Typ "Zahl".

Application Process (OSI)

Der Application Process repräsentiert im *OSI-Referenzmodell* eine Anwendung. Er wird durch den *Application Process Title* global (d.h. weltweit) eindeutig identifiziert.

Application Process Title (OSI)

Gemäß der OSI-Norm dient der Application Process Title (APT) zur global (d.h. weltweit) eindeutigen Identifizierung von Anwendungen. Er kann unterschiedlich aufgebaut sein. openUTM unterstützt den Typ *Object Identifier*.

Application Service Element (OSI)

Ein Application Service Element (ASE) repräsentiert eine Funktionsgruppe der Anwendungsschicht (Schicht 7) des *OSI-Referenzmodells*.

Association (OSI)

Eine Association ist eine Kommunikationsbeziehung zwischen zwei *Application Entities*. Dem Begriff Association entspricht der *LU6.1*-Begriff *Session*.

Asynchron-Auftrag

Auftrag, der vom Auftraggeber zeitlich entkoppelt durchgeführt wird. Zur Bearbeitung von Asynchron-Aufträgen sind in openUTM *Message Queuing* Funktionen integriert, vgl. *UTM-gesteuerte Queue* und *Service-gesteuerte Queue*. Ein Asynchron-Auftrag wird durch die *Asynchron-Nachricht*, den Empfänger und ggf. den gewünschten Ausführungszeitpunkt beschrieben. Ist der Empfänger ein Terminal, ein Drucker oder eine Transportsystem-Anwendung, so ist der Asynchron-Auftrag ein *Ausgabe-Auftrag*; ist der Empfänger ein Asynchron-Vorgang derselben oder einer fernen Anwendung, so handelt es sich um einen *Hintergrund-Auftrag*. Asynchron-Aufträge können *zeitgesteuerte Aufträge* sein oder auch in einen *Auftrags-Komplex* integriert sein.

Asynchron-Conversation

CPI-C-Conversation, bei der nur der *Initiator* senden darf. Für den *Akzeptor* muss in der *UTM-Anwendung* ein asynchroner Transaktionscode generiert sein.

Asynchron-Nachricht

Asynchron-Nachrichten sind Nachrichten, die an eine *Message Queue* gerichtet sind. Sie werden von der lokalen *UTM-Anwendung* zunächst zwischengespeichert und dann unabhängig vom Auftraggeber weiter verarbeitet. Je nach Empfänger unterscheidet man folgende Typen von Asynchron-Nachrichten:

- Bei Asynchron-Nachrichten an eine *UTM-gesteuerte Queue* wird die Weiterverarbeitung komplett durch openUTM gesteuert. Zu diesem Typ gehören Nachrichten, die einen lokalen oder fernen *Asynchron-Vorgang* starten (vgl. auch *Hintergrund-Auftrag*) und Nachrichten, die zur Ausgabe an ein Terminal, einen Drucker oder eine Transportsystem-Anwendung geschickt werden (vgl. auch *Ausgabe-Auftrag*).
- Bei Asynchron-Nachrichten an eine *Service-gesteuerte Queue* wird die Weiterverarbeitung durch einen *Service* der Anwendung gesteuert. Zu diesem Typ gehören Nachrichten an eine *TAC-Queue*, Nachrichten an eine *USER-Queue* und Nachrichten an eine *Temporäre Queue*. Die User-Queue und die Temporäre Queue müssen dabei zur lokalen Anwendung gehören, die TAC-Queue kann sowohl in der lokalen als auch in einer fernen Anwendung liegen.

Asynchron-Programm

Teilprogramm, das von einem *Hintergrund-Auftrag* gestartet wird.

Asynchron-Vorgang (KDCS)

Vorgang, der einen *Hintergrund-Auftrag* bearbeitet. Die Verarbeitung erfolgt entkoppelt vom Auftraggeber. Ein Asynchron-Vorgang kann aus einem oder mehreren Teilprogrammen/Transaktionen bestehen. Er wird über einen asynchronen *Transaktionscode* gestartet.

Auftrag

Anforderung eines *Services*, der von einer *UTM-Anwendung* zur Verfügung gestellt wird, durch Angabe eines *Transaktionscodes*. Siehe auch: *Ausgabe-Auftrag*, *Dialog-Auftrag*, *Hintergrund-Auftrag*, *Auftrags-Komplex*.

Auftraggeber-Vorgang

Ein Auftraggeber-Vorgang ist ein *Vorgang*, der zur Bearbeitung eines Auftrags einen Service von einer anderen Server-Anwendung (*Auftragnehmer-Vorgang*) anfordert.

Auftragnehmer-Vorgang

Ein Auftragnehmer-Vorgang ist ein *Vorgang*, der von einem *Auftraggeber-Vorgang* einer anderen Server-Anwendung gestartet wird.

Auftrags-Komplex

Auftrags-Komplexe dienen dazu, *Asynchron-Aufträgen* *Quittungsaufträge* zuzuordnen. Ein Asynchron-Auftrag innerhalb eines Auftrags-Komplexes wird *Basis-Auftrag* genannt.

Ausgabe-Auftrag

Ausgabeaufträge sind *Asynchron-Aufträge*, die die Aufgabe haben, eine Nachricht, z.B. ein Dokument, an einen Drucker, ein Terminal oder eine Transportsystem-Anwendung auszugeben.

Ausgabeaufträge werden ausschließlich von UTM-Systemfunktionen bearbeitet, d.h. für die Bearbeitung müssen keine Teilprogramme erstellt werden.

Authentisierung

siehe *Zugangskontrolle*.

Autorisierung

siehe *Zugriffskontrolle*.

Axis

siehe *Apache Axis*.

Basis-Auftrag

Asynchron-Auftrag in einem *Auftrags-Komplex*.

Basisformat

Format, in das der Terminal-Benutzer alle Angaben eintragen kann, die notwendig sind, um einen Vorgang zu starten.

Basisname

Basisname UTM-Anwendung.

In BS2000-Systemen ist Basisname das Präfix für die *KDCFILE*, die *Benutzerprotokoll-Datei* USLOG und die *System-Protokolldatei* SYSLOG.

In Unix- und Windows-Systemen ist Basisname der Name des Verzeichnisses, unter dem die *KDCFILE*, die *Benutzerprotokoll-Datei* USLOG, die *System-Protokolldatei* SYSLOG und weitere Dateien der UTM-Anwendung abgelegt sind.

Basisname der Knoten-Anwendung

Dateinamens-Präfix bzw. Verzeichnisname für die *KDCFILE*, *Benutzerprotokoll-Datei* und *Systemprotokoll-Datei* der *Knoten-Anwendung*.

Basisname der UTM-Cluster-Anwendung

Dateinamens-Präfix bzw. Verzeichnisname für die *UTM-Cluster-Dateien*.

Benutzerausgang

Begriff ersetzt durch *Event-Exit*.

Benutzerkennung

Bezeichner für einen Benutzer, der in der *Konfiguration* der *UTM-Anwendung* festgelegt ist (optional mit Passwort zur *Zugangskontrolle*) und dem spezielle Zugriffsrechte (*Zugriffskontrolle*) zugeordnet sind. Ein Terminal-Benutzer muss bei der Anmeldung an die UTM-Anwendung diesen Bezeichner (und ggf. das zugeordnete Passwort) angeben. In BS2000-Systemen ist außerdem eine Zugangskontrolle über *Kerberos* möglich.

Für andere Clients ist die Angabe der Benutzerkennung optional, siehe auch *Verbindungs-Benutzerkennung*.

UTM-Anwendungen können auch ohne Benutzerkennungen generiert werden.

Benutzer-Protokolldatei

Datei oder Dateigeneration, in die der Benutzer mit dem KDCS-Aufruf LPUT Sätze variabler Länge schreibt. Jedem Satz werden die Daten aus dem KB-Kopf des *KDCS-Kommunikationsbereichs* vorangestellt. Die Benutzerprotokolldatei unterliegt der Transaktionssicherung von openUTM.

Berechtigungsprüfung

siehe *Zugangskontrolle*.

Beweissicherung (BS2000-Systeme)

Im Betrieb einer *UTM-Anwendung* können zur Beweissicherung sicherheitsrelevante UTM-Ereignisse von *SAT* protokolliert werden.

Bildschirm-Wiederanlauf

Wird ein *Dialog-Vorgang* unterbrochen, gibt openUTM beim *Vorgangswiederanlauf* die *Dialog-Nachricht* der letzten abgeschlossenen *Transaktion* erneut auf dem Bildschirm aus, sofern die letzte Transaktion eine Nachricht auf den Bildschirm ausgegeben hat.

Browsen von Asynchron-Nachrichten

Ein *Vorgang* liest nacheinander die *Asynchron-Nachrichten*, die sich in einer *Service-gesteuerten Queue* befinden. Die Nachrichten werden während des Lesens nicht gesperrt und verbleiben nach dem Lesen in der Queue. Dadurch ist gleichzeitiges Lesen durch unterschiedliche Vorgänge möglich.

Bypass-Betrieb (BS2000-Systeme)

Betriebsart eines Druckers, der lokal an ein Terminal angeschlossen ist. Im Bypass-Betrieb wird eine an den Drucker gerichtete *Asynchron-Nachricht* an das Terminal gesendet und von diesem auf den Drucker umgeleitet, ohne auf dem Bildschirm angezeigt zu werden.

Cache-Speicher

Pufferbereich zur Zwischenspeicherung von Anwenderdaten für alle Prozesse einer *UTM-Anwendung*. Der Cache-Speicher dient zur Optimierung der Zugriffe auf den *Pagepool* und für UTM-Cluster-Anwendungen zusätzlich auf den *Cluster-Pagepool*.

CCS-Name (BS2000-Systeme)

siehe *Coded-Character-Set-Name*.

Client

Clients einer *UTM-Anwendung* können sein:

- Terminals
- UPIC-Client-Programme
- Transportsystem-Anwendungen (z.B. DCAM-, PDN-, CMX-, Socket-Anwendungen oder UTM-Anwendungen, die als *Transportsystem-Anwendung* generiert sind)

Clients werden über LTERM-Partner an die UTM-Anwendung angeschlossen. openUTM-Clients mit Trägersystem OpenCPIC werden wie *OSI TP-Partner* behandelt.

Client-Seite einer Conversation

Begriff ersetzt durch *Initiator*.

Cluster

Eine Anzahl von Rechnern, die über ein schnelles Netzwerk verbunden sind und die von außen in vielen Fällen als ein Rechner gesehen werden können. Das Ziel des "Clustering" ist meist die Erhöhung der Rechenkapazität oder der Verfügbarkeit gegenüber einem einzelnen Rechner.

Cluster-Administrations-Journal

Das Cluster-Administrations-Journal besteht aus:

- zwei Protokolldateien mit Endungen JRN1 und JRN2 für globale Administrationsaktionen,
- der JKAA-Datei, die eine Kopie der KDCS Application Area (KAA) enthält. Aus dieser Kopie werden administrative Änderungen übernommen, die nicht mehr in den beiden Protokolldateien enthalten sind.

Die Administrations-Journal-Dateien dienen dazu, administrative Aktionen, die in einer UTM-Cluster-Anwendung Cluster-weit auf alle Knoten-Anwendungen wirken sollen, an die anderen Knoten-Anwendungen weiterzugeben.

Cluster-GSSB-Datei

Datei zur Verwaltung von GSSBs in einer *UTM-Cluster-Anwendung*. Die Cluster-GSSB-Datei wird mit dem UTM-Generierungstool *KDCDEF* erstellt.

Cluster-Konfigurationsdatei

Datei, die die zentralen Konfigurationsdaten einer *UTM-Cluster-Anwendung* enthält. Die Cluster-Konfigurationsdatei wird mit dem UTM-Generierungstool *KDCDEF* erstellt.

Cluster-Lock-Datei

Datei einer *UTM-Cluster-Anwendung*, die dazu dient, Knoten-übergreifende Sperren auf Anwenderdatenbereiche zu verwalten.

Cluster-Pagepool

Der Cluster-Pagepool besteht aus einer Verwaltungsdatei und bis zu 10 Dateien, in denen die Cluster-weit verfügbaren Anwenderdaten (Vorgangsdaten inklusive LSSB, GSSB und ULS) einer *UTM-Cluster-Anwendung* gespeichert werden. Der Cluster-Pagepool wird mit dem UTM-Generierungstool *KDCDEF* erstellt.

Cluster-Startserialisierungs-Datei

Lock-Datei, mit der die Starts einzelner Knoten-Anwendungen serialisiert werden (nur bei Unix- und Windows-Systemen).

Cluster-ULS-Datei

Datei zur Verwaltung von ULS-Bereichen einer *UTM-Cluster-Anwendung*. Die Cluster-ULS-Datei wird mit dem UTM-Generierungstool *KDCDEF* erstellt.

Cluster-User-Datei

Datei, die die Verwaltungsdaten der Benutzer einer *UTM-Cluster-Anwendung* enthält. Die Cluster-User-Datei wird mit dem UTM-Generierungstool *KDCDEF* erstellt.

Coded-Character-Set-Name (BS2000-Systeme)

Bei Verwendung des Produkts *XHCS* (**eXtended Host Code Support**) wird jeder verwendete Zeichensatz durch einen Coded-Character-Set-Namen (abgekürzt: "CCS-Name" oder "CCSN") eindeutig identifiziert.

Communication Resource Manager

Communication Resource Manager (CRMs) kontrollieren in verteilten Systemen die Kommunikation zwischen den Anwendungsprogrammen. openUTM stellt CRMs für den internationalen Standard OSI TP, für den Industrie-Standard *LU6.1* und für das openUTM-eigene Protokoll UPIC zur Verfügung.

Contention Loser

Jede Verbindung zwischen zwei Partnern wird von einem der Partner verwaltet. Der Partner, der die Verbindung verwaltet, heißt *Contention Winner*. Der andere Partner ist der Contention Loser.

Contention Winner

Der Contention Winner einer Verbindung übernimmt die Verwaltung der Verbindung. Aufträge können sowohl vom Contention Winner als auch vom *Contention Loser* gestartet werden. Im Konfliktfall, wenn beide Kommunikationspartner gleichzeitig einen Auftrag starten wollen, wird die Verbindung vom Auftrag des Contention Winner belegt.

Conversation

Bei CPI-C nennt man die Kommunikation zwischen zwei CPI-C-Anwendungsprogrammen Conversation. Die Kommunikationspartner einer Conversation werden *Initiator* und *Akzeptor* genannt.

Conversation-ID

Jeder *Conversation* wird von CPI-C lokal eine Conversation-ID zugeordnet, d.h. *Initiator* und *Akzeptor* haben jeweils eine eigene Conversation-ID. Mit der Conversation-ID wird jeder CPI-C-Aufruf innerhalb eines Programms eindeutig einer Conversation zugeordnet.

CPI-C

CPI-C (Common Programming Interface for Communication) ist eine von X/Open und dem CIW (**CPI-C Implementor's Workshop**) normierte Programmierschnittstelle für die Programm-Programm-Kommunikation in offenen Netzen. Das in openUTM implementierte CPI-C genügt der CPI-C V2.0 CAE Specification von X/Open. Die Schnittstelle steht in COBOL und C zur Verfügung. CPI-C in openUTM kann über die Protokolle OSI TP, LU6.1, UPIC und mit openUTM-LU6.2 kommunizieren.

Cross Coupled System / XCS

Verbund von BS2000-Rechnern mit *Highly Integrated System Complex Multiple System Control Facility* (HIPLEX[®] MSCF).

Dead Letter Queue

Die Dead Letter Queue ist eine *TAC-Queue* mit dem festen Namen KDCDLETQ. Sie steht immer zur Verfügung, um Asynchron-Nachrichten an *Transaktionscodes* oder TAC-Queues zu sichern, die nicht verarbeitet werden konnten. Die Sicherung von Asynchron-Nachrichten in der Dead Letter Queue kann durch den Parameter DEAD-LETTER-Q der TAC-Anweisung für jedes Nachrichtenziel einzeln ein- und ausgeschaltet werden.

DES

DES (Data Encryption Standard) ist eine internationale Norm zur Verschlüsselung von Daten. Bei diesem Verfahren wird ein Schlüssel zum Ver- und Entschlüsseln verwendet. Wird das DES-Verfahren verwendet, dann erzeugt der UPIC-Client für jede Sitzung einen DES-Schlüssel.

Dialog-Auftrag

Auftrag, der einen *Dialog-Vorgang* startet. Der Auftrag kann von einem *Client* oder - bei *Server-Server-Kommunikation* - von einer anderen Anwendung erteilt werden.

Dialog-Conversation

CPI-C-Conversation, bei der sowohl der *Initiator* als auch der *Akzeptor* senden darf. Für den *Akzeptor* muss in der *UTM-Anwendung* ein Dialog-Transaktionscode generiert sein.

Dialog-Nachricht

Nachricht, die eine Antwort erfordert oder selbst eine Antwort auf eine Anfrage ist. Dabei bilden Anfrage und Antwort einen *Dialog-Schritt*.

Dialog-Programm

Teilprogramm, das einen *Dialog-Schritt* teilweise oder vollständig bearbeitet.

Dialog-Schritt

Ein Dialog-Schritt beginnt mit dem Empfang einer *Dialog-Nachricht* durch die *UTM-Anwendung*. Er endet mit der Antwort der UTM-Anwendung.

Dialog-Terminalprozess (Unix-/Windows-Systeme)

Ein Dialog-Terminalprozess verbindet ein Unix-/Windows-Terminal mit den *Workprozessen* der *UTM-Anwendung*. Dialog-Terminalprozesse werden entweder vom Benutzer durch Eingabe von `utmdtp` oder über die LOGIN-Shell gestartet. Für jedes Terminal, das an eine UTM-Anwendung angeschlossen werden soll, ist ein eigener Dialog-Terminalprozess erforderlich.

Dialog-Vorgang

Vorgang, der einen *Auftrag* im Dialog (zeitlich gekoppelt) mit dem Auftraggeber (*Client* oder eine andere Server-Anwendung) bearbeitet. Ein Dialog-Vorgang verarbeitet *Dialog-Nachrichten* vom Auftraggeber und erzeugt Dialog-Nachrichten für diesen. Ein Dialog-Vorgang besteht aus mindestens einer *Transaktion*. Ein Dialog-Vorgang umfasst in der Regel mindestens einen *Dialog-Schritt*. Ausnahme: Bei *Vorgangskettung* können auch mehrere Vorgänge einen Dialog-Schritt bilden.

Dienst

Programm auf Windows-Systemen, das im Hintergrund unabhängig von angemeldeten Benutzern oder Fenstern abläuft.

Dienstzugriffspunkt

Im *OSI-Referenzmodell* stehen einer Schicht am Dienstzugriffspunkt die Leistungen der darunterliegenden Schicht zur Verfügung. Der Dienstzugriffspunkt wird im lokalen System durch einen *Selektor* identifiziert. Bei der Kommunikation bindet sich die *UTM-Anwendung* an einen Dienstzugriffspunkt. Eine Verbindung wird zwischen zwei Dienstzugriffspunkten aufgebaut.

Distributed Lock Manager / DLM (BS2000-Systeme)

Konkurrierende, Rechner-übergreifende Dateizugriffe können über den Distributed Lock Manager synchronisiert werden. DLM ist eine Basisfunktion von HIPLEX® MSCF.

Distributed Transaction Processing

X/Open-Architekturmodell für die transaktionsorientierte *verteilte Verarbeitung*.

Druckadministration

Funktionen zur *Drucksteuerung* und Administration von *Ausgabeaufträgen*, die an einen Drucker gerichtet sind.

Druckerbündel

Mehrere Drucker, die demselben *LTERM-Partner* zugeordnet sind.

Druckergruppe (Unix-Systeme)

Die Unix-Plattform richtet für jeden Drucker standardmäßig eine Druckergruppe ein, die genau diesen Drucker enthält. Darüber hinaus lassen sich mehrere Drucker einer Druckergruppe, aber auch ein Drucker mehreren Druckergruppen zuordnen.

Druckerprozess (Unix-Systeme)

Prozess, der vom *Mainprozess* zur Ausgabe von *Asynchron-Nachrichten* an eine *Druckergruppe* eingerichtet wird. Er existiert, solange die Druckergruppe an die *UTM-Anwendung* angeschlossen ist. Pro angeschlossener Druckergruppe gibt es einen Druckerprozess.

Druckersteuerstation

Begriff wurde ersetzt durch *Druckersteuer-LTERM*.

Druckersteuer-LTERM

Über ein Druckersteuer-LTERM kann sich ein *Client* oder ein Terminal-Benutzer an eine *UTM-Anwendung* anschließen. Von dem Client-Programm oder Terminal aus kann dann die *Administration* der Drucker erfolgen, die dem Druckersteuer-LTERM zugeordnet sind. Hierfür ist keine Administrationsberechtigung notwendig.

Drucksteuerung

openUTM-Funktionen zur Steuerung von Druckausgaben.

Dynamische Konfiguration

Änderung der *Konfiguration* durch die Administration. Im laufenden Betrieb der Anwendung können UTM-Objekte wie z.B. *Teilprogramme*, *Transaktionscodes*, *Clients*, *LU6.1-Verbindungen*, Drucker oder *Benutzerkennungen* in die Konfiguration aufgenommen, modifiziert oder teilweise auch gelöscht werden. Hierzu können die Administrationsprogramme WinAdmin oder WebAdmin verwendet werden, oder es müssen eigene *Administrationsprogramme* erstellt werden, die die Funktionen der *Programmschnittstelle der Administration* nutzen.

Einschritt-Transaktion

Transaktion, die genau einen *Dialog-Schritt* umfasst.

Einschritt-Vorgang

Dialog-Vorgang, der genau einen *Dialog-Schritt* umfasst.

Ereignisgesteuerter Vorgang

Begriff ersetzt durch *Event-Service*.

Event-Exit

Routine des *Anwendungsprogramms*, das bei bestimmten Ereignissen (z.B. Start eines Prozesses, Ende eines Vorgangs) automatisch gestartet wird. Diese darf - im Gegensatz zu den *Event-Services* - keine KDCS-, CPI-C- und XATMI-Aufrufe enthalten.

Event-Funktion

Oberbegriff für *Event-Exits* und *Event-Services*.

Event-Service

Vorgang, der beim Auftreten bestimmter Ereignisse gestartet wird, z.B. bei bestimmten UTM-Meldungen. Die *Teilprogramme* ereignisgesteuerter Vorgänge müssen KDCS-Aufrufe enthalten.

Generierung

Statische Konfiguration einer *UTM-Anwendung* mit dem UTM-Tool *KDCDEF* und Erzeugen des *Anwendungsprogramms*.

Globaler Sekundärer Speicherbereich/GSSB

siehe *Sekundärspeicherbereich*.

Hardcopy-Betrieb

Betriebsart eines Druckers, der lokal an ein Terminal angeschlossen ist. Dabei wird eine Nachricht, die auf dem Bildschirm angezeigt wird, zusätzlich auf dem Drucker abgedruckt.

Heterogene Kopplung

Bei *Server-Server-Kommunikation*: Kopplung einer *UTM-Anwendung* mit einer Nicht-UTM-Anwendung, z.B. einer CICS- oder TUXEDO-Anwendung.

Highly Integrated System Complex / HIPLEX®

Produktfamilie zur Realisierung eines Bedien-, Last- und Verfügbarkeitsverbunds mit mehreren BS2000-Servern.

Hintergrund-Auftrag

Hintergrund-Aufträge sind *Asynchron-Aufträge*, die an einen *Asynchron-Vorgang* der eigenen oder einer fernen Anwendung gerichtet sind. Hintergrund-Aufträge eignen sich besonders für zeitintensive oder zeitunkritische Verarbeitungen, deren Ergebnis keinen direkten Einfluss auf den aktuellen Dialog hat.

HIPLEX® MSCF

(MSCF = **M**ultiple **S**ystem **C**ontrol **F**acility)

stellt bei HIPLEX® die Infrastruktur sowie Basisfunktionen für verteilte Anwendungen bereit.

Homogene Kopplung

Bei *Server-Server-Kommunikation*: Kopplung von *UTM-Anwendungen*. Dabei spielt es keine Rolle, ob die Anwendungen auf der gleichen oder auf unterschiedlichen Betriebssystem-Plattformen ablaufen.

Inbound-Conversation (CPI-C)

siehe *Incoming-Conversation*.

Incoming-Conversation (CPI-C)

Eine *Conversation*, bei der das lokale CPI-C-Programm *Akzeptor* ist, heißt Incoming-Conversation. In der X/Open-Specification wird für Incoming-Conversation auch das Synonym Inbound-Conversation verwendet.

Initiale KDCFILE

In einer *UTM-Cluster-Anwendung* die *KDCFILE*, die von *KDCDEF* erzeugt wurde und vor dem Start der Knoten-Anwendungen für jeden Knoten kopiert werden muss.

Initiator (CPI-C)

Die Kommunikationspartner einer *Conversation* werden Initiator und *Akzeptor* genannt. Der Initiator baut die *Conversation* mit den CPI-C-Aufrufen *Initialize_Conversation* und *Allocate* auf.

Insert

Feld in einem Meldungstext, in das openUTM aktuelle Werte einträgt.

Inverser KDCDEF

Funktion, die aus den Konfigurationsdaten der *KDCFILE*, die im laufenden Betrieb dynamisch angepasst wurde, Steueranweisungen für einen *KDCDEF*-Lauf erzeugt. Der inverse *KDCDEF* kann "offline" unter *KDCDEF* oder "online" über die *Programmschnittstelle zur Administration* gestartet werden.

JDK

Java Development Kit
Standard-Entwicklungsumgebung von Sun Microsystems für die Entwicklung von Java-Anwendungen.

Kaltstart

Starten einer *UTM-Anwendung* nach einer *normalen Beendigung* der Anwendung oder nach einer Neugenerierung (vgl. auch *Warmstart*).

KDCADM

Standard-Administrationsprogramm, das zusammen mit openUTM ausgeliefert wird. *KDCADM* stellt Administrationsfunktionen zur Verfügung, die über Transaktionscodes (*Administrationskommandos*) aufgerufen werden.

KDCDEF

UTM-Tool für die *Generierung* von *UTM-Anwendungen*. *KDCDEF* erstellt anhand der Konfigurationsinformationen in den *KDCDEF*-Steueranweisungen die UTM-Objekte *KDCFILE* und die *ROOT*-Tabellen-Source für die Main Routine *KDCROOT*.

In *UTM-Cluster-Anwendungen* erstellt *KDCDEF* zusätzlich die *Cluster-Konfigurationsdatei*, die *Cluster-User-Datei*, den *Cluster-Pagepool*, die *Cluster-GSSB-Datei* und die *Cluster-ULS-Datei*.

KDCFILE

Eine oder mehrere Dateien, die für den Ablauf einer *UTM-Anwendung* notwendige Daten enthalten. Die KDCFILE wird mit dem UTM-Generierungstool *KDCDEF* erstellt. Die KDCFILE enthält unter anderem die *Konfiguration* der Anwendung.

KDCROOT

Main Routine eines *Anwendungsprogramms*, die das Bindeglied zwischen *Teilprogrammen* und UTM-Systemcode bildet. KDCROOT wird zusammen mit den *Teilprogrammen* zum *Anwendungsprogramm* gebunden.

KDCS-Parameterbereich

siehe *Parameterbereich*.

KDCS-Programmschnittstelle

Universelle UTM-Programmschnittstelle, die den nationalen Standard DIN 66 265 erfüllt und Erweiterungen enthält. Mit KDCS (Kompatible Datenkommunikationsschnittstelle) lassen sich z.B. Dialog-Services erstellen und *Message Queuing* Funktionen nutzen. Außerdem stellt KDCS Aufrufe zur *verteilten Verarbeitung* zur Verfügung.

Kerberos

Kerberos ist ein standardisiertes Netzwerk-Authentisierungsprotokoll (RFC1510), das auf kryptographischen Verschlüsselungsverfahren basiert, wobei keine Kennwörter im Klartext über das Netzwerk gesendet werden.

Kerberos-Principal

Eigentümer eines Schlüssels.

Kerberos arbeitet mit symmetrischer Verschlüsselung, d.h. alle Schlüssel liegen an zwei Stellen vor, beim Eigentümer eines Schlüssels (Principal) und beim KDC (Key Distribution Center).

Keycode

Code, der in einer Anwendung eine bestimmte Zugriffsberechtigung oder eine bestimmte Rolle repräsentiert. Mehrere Keycodes werden zu einem *Keyset* zusammengefasst.

Keyset

Zusammenfassung von einem oder mehrerer *Keycodes* unter einem bestimmten Namen. Ein Keyset definiert Berechtigungen im Rahmen des verwendeten Berechtigungskonzepts (Lock-/Keycode-Konzept oder *Access-List*-Konzept). Ein Keyset kann einer *Benutzerkennung*, einem *LTERM-Partner*, einem (OSI-)LPAP-Partner, einem *Service* oder einer *TAC-Queue* zugeordnet werden.

Knoten

Einzelner Rechner eines *Clusters*.

Knoten-Anwendung

UTM-Anwendung, die als Teil einer *UTM-Cluster-Anwendung* auf einem einzelnen *Knoten* zum Ablauf kommt.

Knoten-Recovery

Wenn für eine abnormal beendete Knoten-Anwendung zeitnah kein Wartstart auf ihrem eigenen *Knoten-Rechner* möglich ist, kann man für diesen Knoten auf einem anderen Knoten des UTM-Clusters eine Knoten-Recovery (Wiederherstellung) durchführen. Dadurch können Sperren, die von der ausgefallenen Knoten-Anwendung gehalten werden, freigegeben werden, um die laufende *UTM-Cluster-Anwendung* nicht unnötig zu beeinträchtigen.

Knotengebundener Vorgang

Ein knotengebundener Vorgang eines Benutzers kann nur an der Knoten-Anwendung fortgesetzt werden, an der der Benutzer zuletzt angemeldet war. Folgende Vorgänge sind immer knotengebunden:

- Vorgänge, die eine Kommunikation mit einem Auftragnehmer über LU6.1 oder OSI TP begonnen haben und bei denen der Auftragnehmervorgang noch nicht beendet wurde
- eingeschobene Vorgänge einer Vorgangskellerung
- Vorgänge, die eine SESAM-Transaktion abgeschlossen haben

Außerdem ist der Vorgang eines Benutzers knotengebunden, solange der Benutzer an eine Knoten-Anwendung angemeldet ist.

Kommunikationsbereich/KB (KDCS)

Transaktionsgesicherter KDCS-*Primärspeicherbereich*, der Vorgangs-spezifische Daten enthält. Der Kommunikationsbereich besteht aus 3 Teilen:

- dem KB-Kopf mit allgemeinen Vorgangsdaten,
- dem KB-Rückgabebereich für Rückgaben nach KDCS-Aufrufen
- dem KB-Programmbereich zur Datenübergabe zwischen UTM-Teilprogrammen innerhalb eines *Vorgangs*.

Konfiguration

Summe aller Eigenschaften einer *UTM-Anwendung*. Die Konfiguration beschreibt:

- Anwendungs- und Betriebsparameter
- die Objekte der Anwendung und die Eigenschaften dieser Objekte. Objekte sind z.B. *Teilprogramme* und *Transaktionscodes*, Kommunikationspartner, Drucker, *Benutzerkennungen*

- definierte Zugriffsschutz- und Zugangsschutzmaßnahmen

Die Konfiguration einer UTM-Anwendung wird bei der Generierung festgelegt (*statische Konfiguration*) und kann per *Administration* dynamisch (während des Anwendungslaufs) geändert werden (*dynamische Konfiguration*). Die Konfiguration ist in der *KDCFILE* abgelegt.

Logging-Prozess

Prozess in Unix- und Windows-Systemen, der die Protokollierung von Abrechnungssätzen oder Messdaten steuert.

Logische Verbindung

Zuordnung zweier Kommunikationspartner.

Log4j

Log4j ist ein Teil des Apache Jakarta Projekts. Log4j bietet Schnittstellen zum Protokollieren von Informationen (Ablauf-Informationen, Trace-Records,...) und zum Konfigurieren der Protokoll-Ausgabe. *WS4UTM* verwendet das Softwareprodukt Log4j für die Trace- und Logging-Funktionalität.

Lockcode

Code, um einen LTERM-Partner oder einen Transaktionscode vor unberechtigtem Zugriff zu schützen. Damit ist ein Zugriff nur möglich, wenn das *Keyset* des Zugreifenden den passenden *Keycode* enthält (Lock-/Keycode-Konzept).

Lokaler Sekundärer Speicherbereich/LSSB

siehe *Sekundärspeicherbereich*.

LPAP-Bündel

LPAP-Bündel ermöglichen die Verteilung von Nachrichten an LPAP-Partner auf mehrere Partner-Anwendungen. Soll eine UTM-Anwendung sehr viele Nachrichten mit einer Partner-Anwendung austauschen, kann es für die Lastverteilung sinnvoll sein, mehrere Instanzen der Partner-Anwendung zu starten und die Nachrichten auf die einzelnen Instanzen zu verteilen. In einem LPAP-Bündel übernimmt *openUTM* die Verteilung der Nachrichten an die Instanzen der Partner-Anwendung. Ein LPAP-Bündel besteht aus einem Master-LPAP und mehreren Slave-LPAPs. Die Slave-LPAPs werden dem Master-LPAP bei der Generierung zugeordnet. LPAP-Bündel gibt es sowohl für das OSI TP-Protokoll als auch für das LU6.1-Protokoll.

LPAP-Partner

Für die *verteilte Verarbeitung* über das *LU6.1*-Protokoll muss in der lokalen Anwendung für jede Partner-Anwendung ein LPAP-Partner konfiguriert werden. Der LPAP-Partner spiegelt in der lokalen Anwendung die Partner-

Anwendung wider. Bei der Kommunikation wird die Partner-Anwendung nicht über ihren Anwendungsnamen oder ihre Adresse, sondern über den Namen des zugeordneten LPAP-Partners angesprochen.

LTERM-Bündel

Ein LTERM-Bündel (Verbindungsbündel) besteht aus einem Master-LTERM und mehreren Slave-LTERMs. Mit einem LTERM-Bündel (Verbindungsbündel) verteilen Sie asynchrone Nachrichten an eine logische Partner-Anwendung gleichmäßig auf mehrere parallele Verbindungen.

LTERM-Gruppe

Eine LTERM-Gruppe besteht aus einem oder mehreren Alias-LTERMs, den Gruppen-LTERMs, und einem Primary-LTERM. In einer LTERM-Gruppe ordnen Sie mehrere LTERMs einer Verbindung zu.

LTERM-Partner

Um *Clients* oder Drucker an eine *UTM-Anwendung* anschließen zu können, müssen in der Anwendung LTERM-Partner konfiguriert werden. Ein Client oder Drucker kann nur angeschlossen werden, wenn ihm ein LTERM-Partner mit entsprechenden Eigenschaften zugeordnet ist. Diese Zuordnung wird i.A. in der *Konfiguration* festgelegt, sie kann aber auch dynamisch über Terminal-Pools erfolgen.

LTERM-Pool

Statt für jeden *Client* eine LTERM- und eine PTERM-Anweisung anzugeben, kann mit der Anweisung TPOOL ein Pool von LTERM-Partnern definiert werden. Schließt sich ein Client über einen LTERM-Pool an, wird ihm dynamisch ein LTERM-Partner aus dem Pool zugeordnet.

LU6.1

Geräteunabhängiges Datenaustauschprotokoll (Industrie-Standard) für die transaktionsgesicherte *Server-Server-Kommunikation*.

LU6.1-LPAP-Bündel

LPAP-Bündel für *LU6.1-Partner-Anwendungen*.

LU6.1-Partner

Partner der *UTM-Anwendung*, der mit der UTM-Anwendung über das Protokoll *LU6.1* kommuniziert.

Beispiele für solche Partner sind:

- eine UTM-Anwendung, die über LU6.1 kommuniziert
- eine Anwendung im IBM-Umfeld (z.B. CICS, IMS oder TXSeries), die über LU6.1 kommuniziert

Mainprozess (Unix-/Windows-Systeme)

Prozess, der die *UTM-Anwendung* startet. Er startet die *Workprozesse*, die *UTM-System-Prozesse*, *Druckerprozesse*, *Netzprozesse*, *Logging-Prozess* und den *Timerprozess* und überwacht die *UTM-Anwendung*.

Main Routine KDCROOT

siehe *KDCROOT*.

Management Unit

Komponente des *SE Servers*; ermöglicht mit Hilfe des *SE Managers* ein zentrales, web-basiertes Management aller Units eines *SE Servers*.

Mapped Hostname

Abbildung des UTM-Hostnamen der Partner-Anwendung in einen realen Hostnamen oder umgekehrt.

Meldung / UTM-Meldung

Meldungen werden vom Transaktionsmonitor openUTM oder von UTM-Tools (wie z.B. *KDCDEF*) an *Meldungsziele* ausgegeben. Eine Meldung besteht aus einer Meldungsnummer und dem Meldungstext, der ggf. *Inserts* mit aktuellen Werten enthält. Je nach Meldungsziel werden entweder die gesamte Meldung oder nur Teile der Meldung (z.B. nur die *Inserts*) ausgegeben.

Meldungsdefinitionsdatei

Die Meldungsdefinitionsdatei wird mit openUTM ausgeliefert und enthält standardmäßig die UTM-Meldungstexte in deutscher und englischer Sprache und die Definitionen der Meldungseigenschaften. Aufbauend auf diese Datei kann der Anwender auch eigene, individuelle Meldungsmodule erzeugen.

Meldungsziel

Ausgabemedium für eine *Meldung*. Mögliche Meldungsziele von Meldungen des Transaktionsmonitors openUTM sind z.B. *Terminals*, *TS-Anwendungen*, der *Event-Service MSGTAC*, die *System-Protokolldatei SYSLOG* oder *TAC-Queues*, *Asynchron-TACs*, *USER-Queues*, *SYSOUT/SYSLST* bzw. *stderr/stdout*. Meldungsziele von Meldungen der UTM-Tools sind *SYSOUT/SYSLST* bzw. *stderr/stdout*.

Mehrschritt-Transaktion

Transaktion, die aus mehr als einem *Verarbeitungsschritt* besteht.

Mehrschritt-Vorgang (KDCS)

Vorgang, der in mehreren *Dialog-Schritten* ausgeführt wird.

Message Queuing

Message Queuing (MQ) ist eine Form der Kommunikation, bei der die Nachrichten (Messages) nicht unmittelbar, sondern über zwischengeschaltete *Message Queues* ausgetauscht werden. Sender und Empfänger können zeitlich und räumlich entkoppelt ablaufen. Die Übermittlung der Nachricht hängt nicht davon ab, ob gerade eine Netzverbindung besteht oder nicht. Bei openUTM gibt es *UTM-gesteuerte Queues* und *Service-gesteuerte Queues*.

Message Queue

Warteschlange, in der bestimmte Nachrichten transaktionsgesichert bis zur Weiterverarbeitung eingereiht werden. Je nachdem, wer die Weiterverarbeitung kontrolliert, unterscheidet man *Service-gesteuerte Queues* und *UTM-gesteuerte Queues*.

MSGTAC

Spezieller Event-Service, der Meldungen mit dem Meldungsziel MSGTAC per Programm verarbeitet. MSGTAC ist ein Asynchron-Vorgang und wird vom Betreiber der Anwendung erstellt.

Multiplexanschluss (BS2000-Systeme)

Spezielle Möglichkeit, Terminals an eine *UTM-Anwendung* anzuschließen. Ein Multiplexanschluss ermöglicht es, dass sich mehrere Terminals eine *Transportverbindung* teilen.

Nachrichten-Bereich/NB (KDCS)

Bei KDCS-Aufrufen: Puffer-Bereich, in dem Nachrichten oder Daten für openUTM oder für das *Teilprogramm* bereitgestellt werden.

Nachrichten-Verteiler (BS2000-Systeme)

Einrichtung in einem zentralen Rechner oder Kommunikationsrechner zur Verteilung von Eingabe-Nachrichten an unterschiedliche *UTM-Anwendungen*, die auf unterschiedlichen Rechnern liegen können. Der Nachrichten-Verteiler ermöglicht außerdem, mit *Multiplexanschlüssen* zu arbeiten.

Network File System/Service / NFS

Ermöglicht den Zugriff von Unix-Rechnern auf Dateisysteme über das Netzwerk.

Netzprozess (Unix-/Windows-Systeme)

Prozess einer *UTM-Anwendung* zur Netzanbindung.

Netzwerk-Selektor

Der Netzwerk-Selektor identifiziert im lokalen System einen *Dienstzugriffspunkt* zur Vermittlungsschicht des *OSI-Referenzmodells*.

Normale Beendigung einer UTM-Anwendung

Kontrollierte Beendigung einer *UTM-Anwendung*; das bedeutet u.a., dass die Verwaltungsdaten auf der *KDCFILE* aktualisiert werden. Eine normale Beendigung veranlasst der *Administrator* (z.B. mit *KDCSHUT N*). Den Start nach einer normalen Beendigung führt *openUTM* als *Kaltstart* durch.

Object Identifier

Ein Object Identifier ist ein weltweit eindeutiger Bezeichner für Objekte im OSI-Umfeld. Ein Object Identifier besteht aus einer Folge von ganzen Zahlen, die einen Pfad in einer Baumstruktur repräsentiert.

Offener Terminalpool

Terminalpool, der nicht auf *Clients* eines Rechners oder eines bestimmten Typs beschränkt ist. An diesen Terminalpool können sich alle Clients anschließen, für die kein Rechner- oder Typ-spezifischer Terminalpool generiert ist.

Online-Import

Als Online-Import wird in einer *UTM-Cluster-Anwendung* das Importieren von Anwendungsdaten aus einer normal beendeten Knoten-Anwendung in eine laufende Knoten-Anwendung bezeichnet.

Online-Update

Als Online-Update wird in einer *UTM-Cluster-Anwendung* die Änderung der Konfiguration der Anwendung oder des Anwendungsprogramms oder der Einsatz einer neuen UTM-Korrekturstufe bei laufender *UTM-Cluster-Anwendung* bezeichnet.

OpenCPIC

Trägersystem für UTM-Clients, die das *OSI TP* Protokoll verwenden.

OpenCPIC-Client

OSI TP Partner-Anwendungen mit Trägersystem *OpenCPIC*.

openSM2

Die Produktlinie *openSM2* ist eine einheitliche Lösung für das unternehmensweite Performance Management von Server- und Speichersystemen. *openSM2* bietet eine Messdatenerfassung, Online-Überwachung und Offline-Auswertung.

openUTM-Anwendung

siehe *UTM-Anwendung*.

openUTM-Cluster

aus der Sicht von UPIC-Clients, **nicht** aus Server-Sicht:
Zusammenfassung mehrerer Knoten-Anwendungen einer UTM-Cluster-Anwendung zu einer logischen Anwendung, die über einen gemeinsamen Symbolic Destination Name adressiert wird.

openUTM-D

openUTM-D (openUTM-Distributed) ist eine openUTM-Komponente, die *verteilte Verarbeitung* ermöglicht. openUTM-D ist integraler Bestandteil von openUTM.

OSI-LPAP-Bündel

LPAP-Bündel für *OSI TP*-Partner-Anwendungen.

OSI-LPAP-Partner

OSI-LPAP-Partner sind die bei openUTM generierten Adressen der *OSI TP-Partner*. Für die *verteilte Verarbeitung* über das Protokoll *OSI TP* muss in der lokalen Anwendung für jede Partner-Anwendung ein OSI-LPAP-Partner konfiguriert werden. Der OSI-LPAP-Partner spiegelt in der lokalen Anwendung die Partner-Anwendung wider. Bei der Kommunikation wird die Partner-Anwendung nicht über ihren Anwendungsnamen oder ihre Adresse, sondern über den Namen des zugeordneten OSI-LPAP-Partners angesprochen.

OSI-Referenzmodell

Das OSI-Referenzmodell stellt einen Rahmen für die Standardisierung der Kommunikation von offenen Systemen dar. ISO, die Internationale Organisation für Standardisierung, hat dieses Modell im internationalen Standard ISO IS7498 beschrieben. Das OSI-Referenzmodell unterteilt die für die Kommunikation von Systemen notwendigen Funktionen in sieben logische Schichten. Diese Schichten haben jeweils klar definierte Schnittstellen zu den benachbarten Schichten.

OSI TP

Von der ISO definiertes Kommunikationsprotokoll für die verteilte Transaktionsverarbeitung. OSI TP steht für Open System Interconnection Transaction Processing.

OSI TP-Partner

Partner der UTM-Anwendung, der mit der UTM-Anwendung über das OSI TP-Protokoll kommuniziert.

Beispiele für solche Partner sind:

- eine UTM-Anwendung, die über OSI TP kommuniziert
- eine Anwendung im IBM-Umfeld (z.B. CICS), die über openUTM-LU62 angeschlossen ist

- eine Anwendung des Trägersystems OpenCPIC des openUTM-Client
- Anwendungen anderer TP-Monitore, die OSI TP unterstützen

Outbound-Conversation (CPI-C)

siehe *Outgoing-Conversation*.

Outgoing-Conversation (CPI-C)

Eine Conversation, bei der das lokale CPI-C-Programm der *Initiator* ist, heißt Outgoing-Conversation. In der X/Open-Specification wird für Outgoing-Conversation auch das Synonym Outbound-Conversation verwendet.

Pagepool

Teil der *KDCFILE*, in dem Anwenderdaten gespeichert werden.

In einer *stand-alone-Anwendung* sind dies z.B. *Dialog-Nachrichten*, Nachrichten an *Message Queues*, *Sekundärspeicherbereiche*.

In einer *UTM-Cluster-Anwendung* sind dies z.B. Nachrichten an *Message Queues*, *TLS*.

Parameterbereich

Datenstruktur, in der ein *Teilprogramm* bei einem UTM-Aufruf die für diesen Aufruf notwendigen Operanden an openUTM übergibt.

Partner-Anwendung

Partner einer UTM-Anwendung bei *verteilter Verarbeitung*. Für die verteilte Verarbeitung werden höhere Kommunikationsprotokolle verwendet (*LU6.1*, *OSI TP* oder *LU6.2* über das Gateway openUTM-LU62).

Postselection (BS2000-Systeme)

Auswahl der protokollierten UTM-Ereignisse aus der SAT-Protokolldatei, die ausgewertet werden sollen. Die Auswahl erfolgt mit Hilfe des Tools SATUT.

Prepare to commit (PTC)

Bestimmter Zustand einer verteilten Transaktion:

Das Transaktionsende der verteilten Transaktion wurde eingeleitet, es wird jedoch noch auf die Bestätigung des Transaktionsendes durch den Partner gewartet.

Preselection (BS2000-Systeme)

Festlegung der für die *SAT-Beweissicherung* zu protokollierenden UTM-Ereignisse. Die Preselection erfolgt durch die UTM-SAT-Administration. Man unterscheidet Ereignis-spezifische, Benutzer-spezifische und Auftrags-(TAC-)spezifische Preselection.

Presentation-Selektor

Der Presentation-Selektor identifiziert im lokalen System einen *Dienstzugriffspunkt* zur Darstellungsschicht des *OSI-Referenzmodells*.

Primärspeicherbereich

Bereich im Arbeitsspeicher, auf den das *KDCS-Teilprogramm* direkt zugreifen kann, z.B. *Standard Primärer Arbeitsbereich*, *Kommunikationsbereich*.

Printerprozess (Unix-Systeme)

siehe *Druckerprozess*.

Programmschnittstelle zur Administration

UTM-Programmschnittstelle, mit deren Hilfe der Anwender eigene *Administrationsprogramme* erstellen kann. Die Programmschnittstelle zur Administration bietet u.a. Funktionen zur *dynamischen Konfiguration*, zur Modifikation von Eigenschaften und Anwendungsparametern und zur Abfrage von Informationen zur *Konfiguration* und zur aktuellen Auslastung der Anwendung.

Prozess

In den openUTM-Handbüchern wird der Begriff "Prozess" als Oberbegriff für Prozess (Unix-/Windows-Systeme) und Task (BS2000-Systeme) verwendet.

Queue

siehe *Message Queue*

Quick Start Kit

Beispielanwendung, die mit openUTM (Windows-Systeme) ausgeliefert wird.

Quittungs-Auftrag

Bestandteil eines *Auftrags-Komplexes*, worin der Quittungs-Auftrag dem *Basis-Auftrag* zugeordnet ist. Es gibt positive und negative Quittungsaufträge. Bei positivem Ergebnis des *Basis-Auftrags* wird der positive Quittungs-Auftrag wirksam, sonst der negative.

Redelivery

Erneutes Zustellen einer *Asynchron-Nachricht*, nachdem diese nicht ordnungsgemäß verarbeitet werden konnte, z.B. weil die *Transaktion* zurückgesetzt oder der *Asynchron-Vorgang* abnormal beendet wurde. Die Nachricht wird wieder in die Message Queue eingereiht und lässt sich damit erneut lesen und/oder verarbeiten.

Reentrant-fähiges Programm

Programm, dessen Code durch die Ausführung nicht verändert wird. In BS2000-Systemen ist dies Voraussetzung dafür, *Shared Code* zu nutzen.

Request

Anforderung einer *Service-Funktion* durch einen *Client* oder einen anderen Server.

Requestor

In XATMI steht der Begriff Requestor für eine Anwendung, die einen Service aufruft.

Resource Manager

Resource Manager (RMs) verwalten Datenressourcen. Ein Beispiel für RMs sind Datenbank-Systeme. openUTM stellt aber auch selbst Resource Manager zur Verfügung, z.B. für den Zugriff auf *Message Queues*, lokale Speicherbereiche und Logging-Dateien. Anwendungsprogramme greifen auf RMs über RM-spezifische Schnittstellen zu. Für Datenbank-Systeme ist dies meist SQL, für die openUTM-RMs die Schnittstelle KDCS.

RFC1006

Von IETF (Internet Engineering Task Force) definiertes Protokoll der TCP/IP-Familie zur Realisierung der ISO-Transportdienste (Transportklasse 0) auf TCP/IP-Basis.

RSA

Abkürzung für die Erfinder des RSA-Verschlüsselungsverfahrens Rivest, Shamir und Adleman. Bei diesem Verfahren wird ein Schlüsselpaar verwendet, das aus einem öffentlichen und einem privaten Schlüssel besteht. Eine Nachricht wird mit dem öffentlichen Schlüssel verschlüsselt und kann nur mit dem privaten Schlüssel entschlüsselt werden. Das RSA-Schlüsselpaar wird von der UTM-Anwendung erzeugt.

SAT-Beweissicherung (BS2000-Systeme)

Beweissicherung durch die Komponente SAT (Security Audit Trail) des BS2000-Softwareproduktes SECOS.

SE Manager

Web-basierte Benutzeroberfläche (GUI) für Business Server der SE Serie. Der SE Manager läuft auf der *Management Unit* und ermöglicht die zentrale Bedienung und Verwaltung von Server Units (mit /390-Architektur und/oder x86-Architektur), Application Units (x86-Architektur), Net Unit und der Peripherie.

SE Server

Ein Business Server der SE Serie von Fujitsu.

Sekundärspeicherbereich

Transaktionsgesicherter Speicherbereich, auf den das KDCS-Teilprogramm mit speziellen Aufrufen zugreifen kann. Lokale Sekundärspeicherbereiche (LSSB) sind einem *Vorgang* zugeordnet, auf globale Sekundärspeicherbereiche (GSSB) kann von allen Vorgängen einer *UTM-Anwendung* zugegriffen werden. Weitere Sekundärspeicherbereiche sind der *Terminal-spezifische Langzeitspeicher (TLS)* und der *User-spezifische Langzeitspeicher (ULS)*.

Selektor

Ein Selektor identifiziert im lokalen System einen *Zugriffspunkt* auf die Dienste einer Schicht des *OSI-Referenzmodells*. Jeder Selektor ist Bestandteil der Adresse des Zugriffspunktes.

Semaphor (Unix-/Windows-Systeme)

Betriebsmittel auf Unix- und Windows-Systemen, das zur Steuerung und Synchronisation von Prozessen dient.

Server

Ein Server ist eine *Anwendung*, die *Services* zur Verfügung stellt. Oft bezeichnet man auch den Rechner, auf dem Anwendungen laufen, als Server.

Server-Seite einer Conversation (CPI-C)

Begriff ersetzt durch *Akzeptor*.

Server-Server-Kommunikation

siehe *verteilte Verarbeitung*.

Service Access Point

siehe *Dienstzugriffspunkt*.

Service

Services bearbeiten die *Aufträge*, die an eine Server-Anwendung geschickt werden. Ein Service in einer UTM-Anwendung wird auch *Vorgang* genannt und setzt sich aus einer oder mehreren *Transaktionen* zusammen. Ein Service wird über den *Vorgangs-TAC* aufgerufen. Services können von *Clients* oder anderen Services angefordert werden.

Service-gesteuerte Queue

Message Queue, bei der der Abruf und die Weiterverarbeitung der Nachrichten durch *Services* gesteuert werden. Ein Service muss zum Lesen der Nachricht explizit einen KDCS-Aufruf (DGET) absetzen. Service-gesteuerte Queues gibt es bei openUTM in den Varianten *USER-Queue*, *TAC-Queue* und *Temporäre Queue*.

Service Routine

siehe *Teilprogramm*.

Session

Kommunikationsbeziehung zweier adressierbarer Einheiten im Netz über das SNA-Protokoll *LU6.1*.

Session-Selektor

Der Session-Selektor identifiziert im lokalen System einen *Zugriffspunkt* zu den Diensten der Kommunikationssteuerschicht (Session-Layer) des *OSI-Referenzmodells*.

Shared Code (BS2000-Systeme)

Code, der von mehreren Prozessen gemeinsam benutzt werden kann.

Shared Memory

Virtueller Speicherbereich, auf den mehrere Prozesse gleichzeitig zugreifen können.

Shared Objects (Unix-/Windows-Systeme)

Teile des *Anwendungsprogramms* können als Shared Objects erzeugt werden. Diese werden dynamisch zur Anwendung dazugebunden und können im laufenden Betrieb ausgetauscht werden. Shared Objects werden mit der KDCDEF-Anweisung SHARED-OBJECT definiert.

Sicherungspunkt

Ende einer *Transaktion*. Zu diesem Zeitpunkt werden alle in der Transaktion vorgenommenen Änderungen der *Anwendungsinformation* gegen Systemausfall gesichert und für andere sichtbar gemacht. Während der Transaktion gesetzte Sperren werden wieder aufgehoben.

single system image

Unter single system image versteht man die Eigenschaft eines *Clusters*, nach außen hin als ein einziges, in sich geschlossenes System zu erscheinen. Die heterogene Natur des Clusters und die interne Verteilung der Ressourcen im Cluster ist für die Benutzer des Clusters und die Anwendungen, die mit dem Cluster kommunizieren, nicht sichtbar.

SOA

SOA (Service-oriented architecture).

SOA ist ein Konzept für eine Systemarchitektur, in dem Funktionen in Form von wieder verwendbaren, technisch voneinander unabhängigen und fachlich lose gekoppelten *Services* implementiert werden. Services können unabhängig von zugrunde liegenden Implementierungen über Schnittstellen aufgerufen werden,

deren Spezifikationen öffentlich und damit vertrauenswürdig sein können. Service-Interaktion findet über eine dafür vorgesehene Kommunikationsinfrastruktur statt.

SOAP

SOAP (Simple Object Access Protocol) ist ein Protokoll, mit dessen Hilfe Daten zwischen Systemen ausgetauscht und Remote Procedure Calls durchgeführt werden können. SOAP stützt sich auf die Dienste anderer Standards, XML zur Repräsentation der Daten und Internet-Protokolle der Transport- und Anwendungsschicht zur Übertragung der Nachrichten.

Socket-Verbindung

Transportsystem-Verbindung, die die Socket-Schnittstelle verwendet. Die Socket-Schnittstelle ist eine Standard-Programmschnittstelle für die Kommunikation über TCP/IP.

stand-alone Anwendung

siehe *stand-alone UTM-Anwendung*.

stand-alone UTM-Anwendung

Herkömmliche *UTM-Anwendung*, die nicht Bestandteil einer *UTM-Cluster-Anwendung* ist.

Standard Primärer Arbeitsbereich/SPAB (KDCS)

Bereich im Arbeitsspeicher, der jedem KDCS-*Teilprogramm* zur Verfügung steht. Sein Inhalt ist zu Beginn des Teilprogrammablaufs undefiniert oder mit einem Füllzeichen vorbelegt.

Startformat

Format, das openUTM am Terminal ausgibt, wenn sich ein Benutzer erfolgreich bei der *UTM-Anwendung* angemeldet hat (ausgenommen nach *Vorgangswiederanlauf* und beim Anmelden über *Anmelde-Vorgang*).

statische Konfiguration

Festlegen der *Konfiguration* bei der Generierung mit Hilfe des UTM-Tools *KDCDEF*.

SYSLOG-Datei

siehe *System-Protokolldatei*.

System-Protokolldatei

Datei oder Dateigeneration, in die openUTM während des Laufs einer *UTM-Anwendung* alle UTM-Meldungen protokolliert, für die das *Meldungsziel* SYSLOG definiert ist.

TAC

siehe *Transaktionscode*.

TAC-Queue

Message Queue, die explizit per KDCDEF-Anweisung generiert wird. Eine TAC-Queue ist eine *Service-gesteuerte Queue* und kann unter dem generierten Namen von jedem Service aus angesprochen werden.

Teilprogramm

UTM-*Services* werden durch ein oder mehrere Teilprogramme realisiert. Die Teilprogramme sind Bestandteile des *Anwendungsprogramms*. Abhängig vom verwendeten API müssen sie KDCS-, XATMI- oder CPIC-Aufrufe enthalten. Sie sind über *Transaktionscodes* ansprechbar. Einem Teilprogramm können mehrere Transaktionscodes zugeordnet werden.

Temporäre Queue

Message Queue, die dynamisch per Programm erzeugt wird und auch wieder per Programm gelöscht werden kann, vgl. *Service-gesteuerte Queue*.

Terminal-spezifischer Langzeitspeicher/TLS (KDCS)

Sekundärspeicher, der einem LTERM-, LPAP- oder OSI-LPAP-Partner zugeordnet ist und über das Anwendungsende hinaus erhalten bleibt.

Timerprozess (Unix-/Windows-Systeme)

Prozess, der Aufträge zur Zeitüberwachung von *Workprozessen* entgegennimmt, sie in ein Auftragsbuch einordnet und nach einer im Auftragsbuch festgelegten Zeit den Workprozessen zur Bearbeitung wieder zustellt.

TNS (Unix-/Windows-Systeme)

Abkürzung für den Transport Name Service, der einem Anwendungsnamen einen Transport-Selektor und das Transportsystem zuordnet, über das die Anwendung erreichbar ist.

Tomcat

siehe *Apache Tomcat*

Transaktion

Verarbeitungsabschnitt innerhalb eines *Services*, für den die Einhaltung der *ACID-Eigenschaften* garantiert wird. Von den in einer Transaktion beabsichtigten Änderungen der *Anwendungsinformation* werden entweder alle konsistent durchgeführt oder es wird keine durchgeführt (Alles-oder-Nichts Regel). Das Transaktionsende bildet einen *Sicherungspunkt*.

Transaktionscode/TAC

Name, über den ein *Teilprogramm* aufgerufen werden kann. Der Transaktionscode wird dem Teilprogramm bei der *statischen* oder *dynamischen Konfiguration* zugeordnet. Einem Teilprogramm können auch mehrere Transaktionscodes zugeordnet werden.

Transaktionsrate

Anzahl der erfolgreich beendeten *Transaktionen* pro Zeiteinheit.

Transfer-Syntax

Bei *OSI TP* werden die Daten zur Übertragung zwischen zwei Rechnersystemen von der lokalen Darstellung in die Transfer-Syntax umgewandelt. Die Transfer-Syntax beschreibt die Daten in einem neutralen Format, das von allen beteiligten Partnern verstanden wird. Jeder Transfer-Syntax muss ein *Object Identifier* zugeordnet sein.

Transport-Selektor

Der Transport-Selektor identifiziert im lokalen System einen *Dienstzugriffspunkt* zur Transportschicht des *OSI-Referenzmodells*.

Transportsystem-Anwendung

Anwendung, die direkt auf einer Transportsystem-Schnittstelle wie z.B. CMX, DCAM oder Socket aufsetzt. Für den Anschluss von Transportsystem-Anwendungen muss bei der *Konfiguration* als Partnertyp APPLI oder SOCKET angegeben werden. Eine Transportsystem-Anwendung kann nicht in eine *Verteilte Transaktion* eingebunden werden.

TS-Anwendung

siehe *Transportsystem-Anwendung*.

Typisierter Puffer (XATMI)

Puffer für den Austausch von typisierten und strukturierten Daten zwischen Kommunikationspartnern. Durch diese typisierten Puffer ist die Struktur der ausgetauschten Daten den Partnern implizit bekannt.

UPIC

Trägersystem für UTM-Clients. UPIC steht für Universal Programming Interface for Communication.

UPIC-Client

Bezeichnung für UTM-Clients mit Trägersystem UPIC.

UPIC Analyzer

Komponente zur Analyse der mit *UPIC Capture* mitgeschnittenen UPIC-Kommunikation. Dieser Schritt dient dazu, den Mitschnitt für das Abspielen mit *UPIC Replay* aufzubereiten.

UPIC Capture

Mitschneiden der Kommunikation zwischen UPIC-Clients und UTM-Anwendungen, um sie zu einem späteren Zeitpunkt abspielen zu können (*UPIC Replay*).

UPIC Replay

Komponente zum Abspielen der mit *UPIC Capture* mitgeschnittenen und mit *UPIC Analyzer* aufbereiteten UPIC-Kommunikation.

USER-Queue

Message Queue, die openUTM jeder Benutzerkennung zur Verfügung stellt. Eine USER-Queue zählt zu den *Service-gesteuerten Queues* und ist immer der jeweiligen Benutzerkennung zugeordnet. Der Zugriff von fremden UTM-Benutzern auf die eigene USER-Queue kann eingeschränkt werden.

User-spezifischer Langzeitspeicher/ULS

Sekundärspeicher, der einer *Benutzerkennung*, einer *Session* oder einer *Association* zugeordnet ist und über das Anwendungsende hinaus erhalten bleibt.

USLOG-Datei

siehe *Benutzer-Protokolldatei*.

UTM-Anwendung

Eine UTM-Anwendung stellt *Services* zur Verfügung, die Aufträge von *Clients* oder anderen Anwendungen bearbeiten. openUTM übernimmt dabei u.a. die Transaktionssicherung und das Management der Kommunikations- und Systemressourcen. Technisch gesehen ist eine UTM-Anwendung eine Prozessgruppe, die zur Laufzeit eine logische Server-Einheit bildet.

UTM-Cluster-Anwendung

UTM-Anwendung, die für den Einsatz in einem *Cluster* generiert ist und die man logisch als **eine** Anwendung betrachten kann.

Physikalisch gesehen besteht eine UTM-Cluster-Anwendung aus mehreren, identisch generierten UTM-Anwendungen, die auf den einzelnen *Knoten* laufen.

UTM-Cluster-Dateien

Oberbegriff für alle Dateien, die für den Ablauf einer UTM-Cluster-Anwendung benötigt werden. Dazu gehören folgende Dateien:

- *Cluster-Konfigurationsdatei*

- *Cluster-User-Datei*
 - Dateien des *Cluster-Pagepool*
 - *Cluster-GSSB-Datei*
 - *Cluster-ULS-Datei*
 - Dateien des *Cluster-Administrations-Journals**
 - *Cluster-Lock-Datei**
 - Lock-Datei zur Start-Serialisierung* (nur bei Unix- und Windows-Systemen)
- Die mit * gekennzeichneten Dateien werden beim Start der ersten Knoten-Anwendung angelegt, alle anderen Dateien werden bei der Generierung mit KDCDEF erzeugt.

UTM-D

siehe *openUTM-D*.

UTM-Datenstation

Begriff ersetzt durch *LTERM-Partner*.

UTM-F

UTM-Anwendungen können als UTM-F-Anwendungen (UTM-Fast) generiert werden. Bei UTM-F wird zugunsten der Performance auf Platteneingaben/-ausgaben verzichtet, mit denen bei *UTM-S* die Sicherung von Benutzer- und Transaktionsdaten durchgeführt wird. Gesichert werden lediglich Änderungen der Verwaltungsdaten.

In UTM-Cluster-Anwendungen, die als UTM-F-Anwendung generiert sind (APPLIMODE=FAST), werden Cluster-weit gültige Anwenderdaten auch gesichert. Dabei werden GSSB- und ULS-Daten genauso behandelt wie in UTM-Cluster-Anwendungen, die mit UTM-S generiert sind. Vorgangs-Daten von Benutzern mit RESTART=YES werden jedoch nur beim Abmelden des Benutzers anstatt bei jedem Transaktionsende geschrieben.

UTM-gesteuerte Queues

Message Queues, bei denen der Abruf und die Weiterverarbeitung der Nachrichten vollständig durch openUTM gesteuert werden. Siehe auch *Asynchron-Auftrag*, *Hintergrund-Auftrag* und *Asynchron-Nachricht*.

UTM-S

Bei UTM-S-Anwendungen sichert openUTM neben den Verwaltungsdaten auch alle Benutzerdaten über ein Anwendungsende und einen Systemausfall hinaus. Außerdem garantiert UTM-S bei allen Störungen die Sicherheit und Konsistenz der Anwendungsdaten. Im Standardfall werden UTM-Anwendungen als UTM-S-Anwendungen (UTM-Secure) generiert.

UTM-SAT-Administration (BS2000-Systeme)

Durch die UTM-SAT-Administration wird gesteuert, welche sicherheitsrelevanten UTM-Ereignisse, die im Betrieb der *UTM-Anwendung* auftreten, von *SAT* protokolliert werden sollen. Für die UTM-SAT-Administration wird eine besondere Berechtigung benötigt.

UTM-Seite

Ist eine Speichereinheit, die entweder 2K, 4K oder 8K umfasst. In *stand-alone UTM-Anwendungen* kann die Größe einer UTM-Seite bei der Generierung der UTM-Anwendung auf 2K, 4K oder 8K gesetzt werden. In einer *UTM-Cluster-Anwendung* ist die Größe einer UTM-Seite immer 4K oder 8K. *Pagepool* und Wiederanlauf-Bereich der *KDCFILE* sowie *UTM-Cluster-Dateien* werden in Einheiten der Größe einer UTM-Seite unterteilt.

UTM-System-Prozess

UTM-Prozess, der zusätzlich zu den per Startparameter angegebenen Prozessen gestartet wird und nur ausgewählte Aufträge bearbeitet. UTM-System-Prozesse dienen dazu, eine UTM-Anwendung auch bei sehr hoher Last reaktionsfähig zu halten.

utmpfad (Unix-/Windows-Systeme)

Das Dateiverzeichnis unter dem die Komponenten von openUTM installiert sind, wird in diesem Handbuch als *utmpfad* bezeichnet.

Um einen korrekten Ablauf von openUTM zu garantieren, muss die Umgebungsvariable *UTMPATH* auf den Wert von *utmpfad* gesetzt werden. Auf Unix-Systemen müssen Sie *UTMPATH* vor dem Starten einer UTM-Anwendung setzen, auf Windows-Systemen wird *UTMPATH* bei der Installation gesetzt.

Verarbeitungsschritt

Ein Verarbeitungsschritt beginnt mit dem Empfangen einer *Dialog-Nachricht*, die von einem *Client* oder einer anderen Server-Anwendung an die *UTM-Anwendung* gesendet wird. Der Verarbeitungsschritt endet entweder mit dem Senden einer Antwort und beendet damit auch den *Dialog-Schritt* oder er endet mit dem Senden einer Dialog-Nachricht an einen Dritten.

Verbindungs-Benutzerkennung

Benutzerkennung, unter der eine *TS-Anwendung* oder ein *UPIC-Client* direkt nach dem Verbindungsaufbau bei der *UTM-Anwendung* angemeldet wird.

Abhängig von der Generierung des Clients (= *LTERM-Partner*) gilt:

- Die Verbindungs-Benutzerkennung ist gleich dem *USER* der *LTERM-Anweisung* (explizite Verbindungs-Benutzerkennung). Eine explizite Verbindungs-Benutzerkennung muss mit einer *USER-Anweisung* generiert sein und kann nicht als "echte" *Benutzerkennung* verwendet werden.

- Die Verbindungs-Benutzererkennung ist gleich dem LTERM-Partner (implizite Verbindungs-Benutzererkennung), wenn bei der LTERM-Anweisung kein USER angegeben wurde oder wenn ein LTERM-Pool generiert wurde. In einer *UTM-Cluster-Anwendung* ist der Vorgang einer Verbindungs-Benutzererkennung (RESTART=YES bei LTERM oder USER) an die Verbindung gebunden und damit Knoten-lokal. Eine Verbindungs-Benutzererkennung, die mit RESTART=YES generiert ist, kann in jeder *Knoten-Anwendung* einen eigenen Vorgang haben.

Verbindungsbündel

siehe *LTERM-Bündel*.

Verschlüsselungsstufe

Die Verschlüsselungsstufe legt fest, ob und inwieweit ein Client Nachrichten und Passwort verschlüsseln muss.

Verteilte Transaktion

Transaktion, die sich über mehr als eine Anwendung erstreckt und in mehreren (Teil)-Transaktionen in verteilten Systemen ausgeführt wird.

Verteilte Transaktionsverarbeitung

Verteilte Verarbeitung mit verteilten Transaktionen.

Verteilte Verarbeitung

Bearbeitung von *Dialog-Aufträgen* durch mehrere Anwendungen oder Übermittlung von *Hintergrundaufträgen* an eine andere Anwendung. Für die verteilte Verarbeitung werden die höheren Kommunikationsprotokolle *LU6.1* und *OSITP* verwendet. Über openUTM-LU62 ist verteilte Verarbeitung auch mit LU6.2 Partnern möglich. Man unterscheidet verteilte Verarbeitung mit *verteilten Transaktionen* (Anwendungs-übergreifende Transaktionssicherung) und verteilte Verarbeitung ohne verteilte Transaktionen (nur lokale Transaktionssicherung). Die verteilte Verarbeitung wird auch Server-Server-Kommunikation genannt.

Vorgang (KDCS)

Ein Vorgang dient zur Bearbeitung eines *Auftrags* in einer *UTM-Anwendung*. Er setzt sich aus einer oder mehreren *Transaktionen* zusammen. Die erste Transaktion wird über den *Vorgangs-TAC* aufgerufen. Es gibt *Dialog-Vorgänge* und *Asynchron-Vorgänge*. openUTM stellt den Teilprogrammen eines Vorgangs gemeinsame Datenbereiche zur Verfügung. Anstelle des Begriffs Vorgang wird häufig auch der allgemeinere Begriff *Service* gebraucht.

Vorgangs-Kellerung (KDCS)

Ein Terminal-Benutzer kann einen laufenden *Dialog-Vorgang* unterbrechen und einen neuen Dialog-Vorgang einschieben. Nach Beendigung des eingeschobenen *Vorgangs* wird der unterbrochene Vorgang fortgesetzt.

Vorgangs-Kettung (KDCS)

Bei Vorgangs-Kettung wird nach Beendigung eines *Dialog-Vorgangs* ohne Angabe einer *Dialog-Nachricht* ein Folgevorgang gestartet.

Vorgangs-TAC (KDCS)

Transaktionscode, mit dem ein *Vorgang* gestartet wird.

Vorgangs-Wiederanlauf (KDCS)

Wird ein Vorgang unterbrochen, z.B. infolge Abmeldens des Terminal-Benutzers oder Beendigung der *UTM-Anwendung*, führt openUTM einen Vorgangs-Wiederanlauf durch. Ein *Asynchron-Vorgang* wird neu gestartet oder beim zuletzt erreichten *Sicherungspunkt* fortgesetzt, ein *Dialog-Vorgang* wird beim zuletzt erreichten Sicherungspunkt fortgesetzt. Für den Terminal-Benutzer wird der Vorgangs-Wiederanlauf eines Dialog-Vorgangs als *Bildschirm-Wiederanlauf* sichtbar, sofern am letzten Sicherungspunkt eine Dialog-Nachricht an den Terminal-Benutzer gesendet wurde.

Warmstart

Start einer *UTM-S-Anwendung* nach einer vorhergehenden abnormalen Beendigung. Dabei wird die *Anwendungsinformation* auf den zuletzt erreichten konsistenten Zustand gesetzt. Unterbrochene *Dialog-Vorgänge* werden dabei auf den zuletzt erreichten *Sicherungspunkt* zurückgesetzt, so dass die Verarbeitung an dieser Stelle wieder konsistent aufgenommen werden kann (*Vorgangs-Wiederanlauf*). Unterbrochene *Asynchron-Vorgänge* werden zurückgesetzt und neu gestartet oder beim zuletzt erreichten *Sicherungspunkt* fortgesetzt. Bei UTM-F-Anwendungen werden beim Start nach einer vorhergehenden abnormalen Beendigung lediglich die dynamisch geänderten Konfigurationsdaten auf den zuletzt erreichten konsistenten Zustand gesetzt. In UTM-Cluster-Anwendungen werden die globalen Sperren auf GSSB und ULS, die bei der abnormalen Beendigung von dieser Knoten-Anwendung gehalten wurden, aufgehoben. Außerdem werden Benutzer, die zum Zeitpunkt der abnormalen Beendigung an dieser Knoten-Anwendung angemeldet waren, abgemeldet.

Web Service

Anwendung, die auf einem Web-Server läuft und über eine standardisierte und programmatische Schnittstelle (öffentlich) verfügbar ist. Die Web Services-Technologie ermöglicht es, UTM-Teilprogramme für moderne Web-Client-Anwendungen verfügbar zu machen, unabhängig davon, in welcher Programmiersprache sie entwickelt wurden.

WebAdmin

Web-basiertes Tool zur Administration von openUTM-Anwendungen über Web-Browser. WebAdmin enthält neben den kompletten Funktionsumfang der *Programmschnittstelle zur Administration* noch zusätzliche Funktionen.

Wiederanlauf

siehe *Bildschirm-Wiederanlauf*,
siehe *Vorgangs-Wiederanlauf*.

WinAdmin

Java-basiertes Tool zur Administration von openUTM-Anwendungen über eine grafische Oberfläche. WinAdmin enthält neben dem kompletten Funktionsumfang der *Programmschnittstelle zur Administration* noch zusätzliche Funktionen.

Workload Capture & Replay

Programmfamilie zur Simulation von Lastsituationen, bestehend aus den Haupt-Komponenten *UPIC Capture*, *UPIC Analyzer* und *Upic Replay* und auf Unix- und Windows-Systemen) dem Dienstprogramm *kdcsort*. Mit Workload Capture & Replay lassen sich UPIC-Sessions mit UTM-Anwendungen aufzeichnen, analysieren und mit veränderten Lastparametern wieder abspielen.

Workprozess (Unix-/Windows-Systeme)

Prozess, in dem die *Services* der *UTM-Anwendung* ablaufen.

WS4UTM

WS4UTM (**W**eb**S**ervices for open**U**TM) ermöglicht es Ihnen, auf komfortable Weise einen Service einer UTM-Anwendung als Web Service zur Verfügung zu stellen.

XATMI

XATMI (X/Open Application Transaction Manager Interface) ist eine von X/Open standardisierte Programmschnittstelle für die Programm-Programm-Kommunikation in offenen Netzen.

Das in openUTM implementierte XATMI genügt der XATMI CAE Specification von X/Open. Die Schnittstelle steht in COBOL und C zur Verfügung. XATMI in openUTM kann über die Protokolle *OSI TP*, *LU6.1* und *UPIC* kommunizieren.

XHCS (BS2000-Systeme)

XHCS (Extended Host Code Support) ist ein BS2000-Softwareprodukt für die Unterstützung internationaler Zeichensätze.

XML

XML (eXtensible Markup Language) ist eine vom W3C (WWW-Konsortium) genormte Metasprache, in der Austauschformate für Daten und zugehörige Informationen definiert werden können.

Zeitgesteuerter Auftrag

Auftrag, der von openUTM bis zu einem definierten Zeitpunkt in einer *Message Queue* zwischengespeichert und dann an den Empfänger weitergeleitet wird. Empfänger kann sein: ein *Asynchron-Vorgang* der selben Anwendung, eine *TAC-Queue*, eine Partner-Anwendung, ein Terminal oder ein Drucker. Zeitgesteuerte Aufträge können nur von *KDCS-Teilprogrammen* erteilt werden.

Zugangskontrolle

Prüfung durch openUTM, ob eine bestimmte *Benutzerkennung* berechtigt ist, mit der *UTM-Anwendung* zu arbeiten. Die Berechtigungsprüfung entfällt, wenn die UTM-Anwendung ohne Benutzerkennungen generiert wurde.

Zugriffskontrolle

Prüfung durch openUTM, ob der Kommunikationspartner berechtigt ist, auf ein bestimmtes Objekt der Anwendung zuzugreifen. Die Zugriffsrechte werden als Bestandteil der Konfiguration festgelegt.

Zugriffspunkt

siehe *Dienstzugriffspunkt*.

Abkürzungen

ACSE	Association Control Service Element
AEQ	Application Entity Qualifier
AES	Advanced Encryption Standard
AET	Application Entity Title
APT	Application Process Title
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
Axis	Apache eXtensible Interaction System
BCAM	Basic Communication Access Method
BER	Basic Encoding Rules
BLS	Binder-Lader-Starter (BS2000)
CCP	Communication Control Program
CCR	Commitment, Concurrency and Recovery
CCS	Codierter Zeichensatz (Coded Character Set)
CCSN	Name des codierten Zeichensatzes (Coded Character Set Name)
CICS	Customer Information Control System (IBM)
CID	Control Identification
CMX	Communication Manager in Unix-Systemen
COM	Component Object Model
CPI-C	Common Programming Interface for Communication
CRM	Communication Resource Manager
CRTE	Common Runtime Environment (BS2000)
DB	Datenbank
DC	Data Communication
DCAM	Data Communication Access Method

Abkürzungen

DES	Data Encryption Standard
DLM	Distributed Lock Manager (BS2000)
DMS	Data Management System
DNS	Domain Name Service
DSS	Datensichtstation (=Terminal)
DTD	Document Type Definition
DTP	Distributed Transaction Processing
DVS	Datenverwaltungssystem
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EJB	Enterprise JavaBeans TM
FGG	File Generation Group
FHS	Format Handling System
FT	File Transfer
GSSB	Globaler Sekundärer Speicherbereich
HIPLEX [®]	Highly Integrated System Complex (BS2000)
HLL	High-Level Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IFG	Interaktiver Format-Generator
ILCS	Inter Language Communication Services (BS2000)
IMS	Information Management System (IBM)
IPC	Inter-Process-Communication
IRV	Internationale Referenzversion
ISO	International Organization for Standardization
Java EE	Java Platform, Enterprise Edition
JCA	Java EE Connector Architecture
JDK	Java Development Kit
KA	KDCS Application Area
KB	Kommunikationsbereich
KBPROG	KB-Programmbereich
KDCADMI	KDC Administration Interface
KDCS	Kompatible Datenkommunikationsschnittstelle

KTA	KDCS Task Area
LAN	Local Area Network
LCF	Local Configuration File
LLM	Link and Load Module (BS2000)
LSSB	Lokaler Sekundärer Speicherbereich
LU	Logical Unit
MQ	Message Queuing
MSCF	Multiple System Control Facility (BS2000)
NB	Nachrichtenbereich
NEA	Netzwerkarchitektur bei BS2000-Systemen
NFS	Network File System/Service
NLS	Unterstützung der Landessprache (Native Language Support)
OLTP	Online Transaction Processing
OML	Object Modul Library
OSI	Open System Interconnection
OSI TP	Open System Interconnection Transaction Processing
OSS	OSI Session Service
PCMX	Portable Communication Manager
PID	Prozess-Identifikation
PIN	Persönliche Identifikationsnummer
PLU	Primary Logical Unit
PTC	Prepare to commit
RAV	Rechenzentrums-Abrechnungs-Verfahren
RDF	Resource Definition File
RM	Resource Manager
RSA	Encryption-Algorithmus nach Rivest, Shamir, Adleman
RSO	Remote SPOOL Output (BS2000)
RTS	Runtime System (Laufzeitsystem)
SAT	Security Audit Trail (BS2000)
SECOS	Security Control System
SEM	SE Manager
SGML	Standard Generalized Markup Language

Abkürzungen

SLU	Secondary Logical Unit
SM2	Software Monitor 2
SNA	Systems Network Architecture
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SPAB	Standard Primärer Arbeitsbereich
SQL	Structured Query Language
SSB	Sekundärer Speicherbereich
SSO	Single-Sign-On
TAC	Transaktionscode
TCEP	Transport Connection End Point
TCP/IP	Transport Control Protocol / Internet Protocol
TIAM	Terminal Interactive Access Method
TLS	Terminal-spezifischer Langzeitspeicher
TM	Transaction Manager
TNS	Transport Name Service
TP	Transaction Processing (Transaktions-Betrieb)
TPR	Task privileged (privilegierter Funktionszustand des BS2000)
TPSU	Transaction Protocol Service User
TSAP	Transport Service Access Point
TSN	Task Sequence Number
TU	Task user (nicht privilegierter Funktionszustand des BS2000)
TX	Transaction Demarcation (X/Open)
UDDI	Universal Description, Discovery and Integration
UDS	Universelles Datenbanksystem
UDT	Unstructured Data Transfer
ULS	User-spezifischer Langzeitspeicher
UPIC	Universal Programming Interface for Communication
USP	UTM-Socket-Protokoll
UTM	Universeller Transaktionsmonitor
UTM-D	UTM-Funktionen für verteilte Verarbeitung („Distributed“)
UTM-F	Schnelle UTM-Variante („Fast“)
UTM-S	UTM-Sicherheitsvariante

UTM-XML	XML-Schnittstelle von openUTM
VGID	Vorgangs-Identifikation
VTSU	Virtual Terminal Support
VTV	Verteilte Transaktionsverarbeitung
VV	Verteilte Verarbeitung
WAN	Wide Area Network
WS4UTM	WebServices for openUTM
WSDD	Web Service Deployment Descriptor
WSDL	Web Services Description Language
XA	X/Open Access Interface (Schnittstelle von X/Open zum Zugriff auf Resource Manager)
XAP	X/OPEN ACSE/Presentation programming interface
XAP-TP	X/OPEN ACSE/Presentation programming interface Transaction Processing extension
XATMI	X/Open Application Transaction Manager Interface
XCS	Cross Coupled System
XHCS	eXtended Host Code Support
XML	eXtensible Markup Language

Literatur

Die Handbücher finden Sie im Internet unter <http://manuals.ts.fujitsu.com>. Handbücher, die mit einer Bestellnummer angezeigt werden, können Sie auch in gedruckter Form bestellen.



PDF-Dateien von allen openUTM-Handbüchern sind sowohl auf der openUTM Enterprise Edition DVD für die offenen Plattformen als auch für BS2000-Systeme auf der openUTM WinAdmin-DVD enthalten.

Dokumentation zu openUTM

openUTM

Konzepte und Funktionen

Benutzerhandbuch

openUTM

Anwendungen programmieren mit KDCS für COBOL, C und C++

Basishandbuch

openUTM

Anwendungen generieren

Benutzerhandbuch

openUTM

Einsatz von openUTM-Anwendungen unter BS2000-Systemen

Benutzerhandbuch

openUTM

Einsatz von openUTM-Anwendungen unter Unix- und Windows-Systemen

Benutzerhandbuch

openUTM

Anwendungen administrieren

Benutzerhandbuch

openUTM

Meldungen, Test und Diagnose in BS2000-Systemen

Benutzerhandbuch

openUTM

Meldungen, Test und Diagnose in Unix- und Windows-Systemen

Benutzerhandbuch

openUTM

Anwendungen erstellen mit X/Open-Schnittstellen

Benutzerhandbuch

openUTM

XML für openUTM

openUTM-Client (Unix-Systeme)

für Trägersystem OpenCPIC

Client-Server-Kommunikation mit openUTM

Benutzerhandbuch

openUTM-Client

für Trägersystem UPIC

Client-Server-Kommunikation mit openUTM

Benutzerhandbuch

openUTM WinAdmin

Grafischer Administrationsarbeitsplatz für openUTM

Beschreibung und Online-Hilfe

openUTM WebAdmin

Web-Oberfläche zur Administration von openUTM

Beschreibung und Online-Hilfe

openUTM, openUTM-LU62

Verteilte Transaktionsverarbeitung

zwischen openUTM und CICS-, IMS- und LU6.2-Anwendungen

Benutzerhandbuch

openUTM (BS2000)

Anwendungen programmieren mit KDCS für Assembler

Ergänzung zum Basishandbuch

openUTM (BS2000)

Anwendungen programmieren mit KDCS für Fortran

Ergänzung zum Basishandbuch

openUTM (BS2000)

Anwendungen programmieren mit KDCS für Pascal-XT

Ergänzung zum Basishandbuch

openUTM (BS2000)

Anwendungen programmieren mit KDCS für PL/I

Ergänzung zum Basishandbuch

WS4UTM (Unix- und Windows-Systeme)

Web-Services für openUTM

openUTM

Masterindex

Dokumentation zum openSEAS-Produktumfeld

BeanConnect

Benutzerhandbuch

JConnect

Verbindung von Java-Clients zu openUTM

Benutzerdokumentation und Java-Docs

WebTransactions

Konzepte und Funktionen

WebTransactions

Template-Sprache

WebTransactions

Anschluss an openUTM-Anwendungen über UPIC

WebTransactions

Anschluss an MVS-Anwendungen

WebTransactions

Anschluss an OSD-Anwendungen

Dokumentation zum BS2000-Umfeld

AID

Advanced Interactive Debugger

Basishandbuch

Benutzerhandbuch

BCAM

BCAM Band 1/2

Benutzerhandbuch

BINDER

Benutzerhandbuch

BS2000 OSD/BC

Makroaufrufe an den Ablaufteil

Benutzerhandbuch

BS2000

BLSSERV

Bindelader-Starter

Benutzerhandbuch

DCAM

COBOL-Aufrufe

Benutzerhandbuch

DCAM

Makroaufrufe

Benutzerhandbuch

DCAM

Programmschnittstellen

Beschreibung

FHS

Formatierungssystem für openUTM, TIAM, DCAM

Benutzerhandbuch

IFG für FHS

Benutzerhandbuch

HIPLEX AF
Hochverfügbarkeit von Anwendungen in BS2000/OSD
Produkt Handbuch

HIPLEX MSCF
BS2000-Rechner im Verbund
Benutzerhandbuch

IMON
Installationsmonitor
Benutzerhandbuch

LMS
SDF-Format
Benutzerhandbuch

MT9750 (MS Windows)
9750-Emulation unter Windows
Produkt Handbuch

OMNIS/OMNIS-MENU (BS2000)
Funktionen und Kommandos
Benutzerhandbuch

OMNIS/OMNIS-MENU (BS2000)
Administration und Programmierung
Benutzerhandbuch

OSS (BS2000)
OSI Session Service
User Guide

RSO
Remote SPOOL Output
Benutzerhandbuch

SECOS
Security Control System
Benutzerhandbuch

SECOS
Security Control System
Tabellenheft

SESAM/SQL

Datenbankbetrieb

Benutzerhandbuch

openSM2

Software Monitor

Band 1: Verwaltung und Bedienung

TIAM

Benutzerhandbuch

UDS/SQL

Datenbankbetrieb

Benutzerhandbuch

Unicode im BS2000/OSD

Übersichtshandbuch

VTSU

Virtual Terminal Support

Benutzerhandbuch

XHCS

8-bit-Code- und Unicode-Unterstützung im BS2000/OSD

Benutzerhandbuch

Dokumentation zum Umfeld von Unix-Systemen

CMX V6.0 (Unix-Systeme)
Betrieb und Administration
Benutzerhandbuch

CMX V6.0
CMX-Anwendungen programmieren
Programmierhandbuch

OSS (UNIX)
OSI Session Service
User Guide

PRIMECLUSTERTM
Konzept (Solaris, Linux)
Benutzerhandbuch

openSM2
Die Dokumentation zu openSM2 wird in Form von ausführlichen Online-Hilfen bereitgestellt, die mit dem Produkt ausgeliefert werden.

Sonstige Literatur

XCPI-C (X/Open)

Distributed Transaction Processing
X/Open CAE Specification, Version 2
ISBN 1 85912 135 7

Reference Model Version 2 (X/Open)

Distributed Transaction Processing
X/Open Guide
ISBN 1 85912 019 9

TX (Transaction Demarcation) (X/Open)

Distributed Transaction Processing
X/Open CAE Specification
ISBN 1 85912 094 6

XATMI (X/Open)

Distributed Transaction Processing
X/Open CAE Specification
ISBN 1 85912 130 6

XML

Spezifikation des W3C (www – Konsortium)
Webseite: <http://www.w3.org/XML>

Stichwörter

- !
- !!
- KDCDUMP-Anweisung [64](#)
- KDCDUMP-Anweisung [65](#)
- %LANG% [133](#), [152](#)
- \$LANG [126](#)
- A**
- Abmelden mit DEL bzw. CTRL+C [37](#)
- Abnormal beenden
 - der Anwendung [181](#)
 - Vorgang [40](#)
- ACCOUNTING-AREA [93](#)
- adb [45](#)
- ADMI-DIAGAREA [112](#)
 - Dump [92](#)
- ADMI-Trace [55](#)
- ADMI-USERAREA [116](#)
 - Dump [92](#)
- Administrations-Journal [452](#)
- Administrations-Journal-Dateien
 - Meldung K190 [295](#)
- Administrationskommando
 - Testmodus einschalten [41](#)
- Administrationsprogramm
 - Message-Dump-Funktion ausschalten [43](#)
 - Message-Dump-Funktion einschalten [42](#)
- AFIND
 - KDCDUMP-Anweisung [65](#)
- Ändern
 - der Meldungsdatei [141](#)
 - Meldungen [134](#)
 - U-Meldungen [141](#)
- Animator [29](#)
- Anmelden
 - PTERM-Name [37](#)
- Announcements
 - Trace [48](#)
- Anwendung
 - UTM-Meldung an andere [130](#)
- Anwendungsglobaler Systemspeicher (KAA) [82](#)
- Anwendungsglobaler Systemspeicher (XAP-TP) [87](#)
- Area Table [90](#)
- Aufbereiten
 - SYSLOG-Datei [154](#), [155](#)
- Aufbereiteter Dump [81](#)
- Aufbereitungstool
 - KDCPSYSL [155](#)
- Aufrufen
 - KDCDUMP [62](#)
 - KDCIPC [56](#)
 - KDCKAA [57](#)
 - KDCMMOD [144](#)
 - KDCMTXT [136](#)
- Aufrufhierarchie anzeigen lassen [45](#)
- Ausfall Knoten-Anwendung
 - Meldung K190 [302](#)
- Ausgabe-Nachrichten
 - Trace [48](#)
- Auswerten
 - BCAM-Trace [50](#)
 - KTA-Trace [52](#)
 - Meldungstexte durch UTM [132](#)
 - OSS-Trace [54](#)
- AUTOMATIC STORAGES [81](#)
- B**
- BCAM-Parameterblöcke [48](#)

- BCAM-Trace 48
 - auswerten 50
 - ein-/ausschalten 49
- Bedienplatz 128
- Benutzer-spezifische Meldungsziele 128, 131
- Big Endian 105
- Blättern
 - KDCDUMP 65
- BUF-SGMT 120

- C**
- C-String 64
- CF-ENT 120
- CF-HDR 120
- Cluster-Administrations-Journal 452
- Cluster-Konfigurationsdatei
 - Meldung K190 287
- Cluster-Pagepool
 - Meldung K190 296
- Cluster-ULS-Datei
 - Meldung K190 298, 299
- Cluster-User-Datei
 - Meldung K190 289
- CMX-Record
 - Trace 48
- COBOL-Testhilfen 29
- Connection-Letters
 - Trace 48
- CONS_ENTRIES 85
- CONSOLE 130
- CONSOLE (Meldungsziel) 128
- console.txt 130
- CONTEXT-AREA 90
- CTRL + C ignorieren 37

- D**
- Darstellungsmittel 25
- Dateibearbeitung
 - Fehler 383
- Datenverschlüsselung 272
- Datum
 - Meldungen 129
- DB Error Code 111
- DB Operations Code 110

- DB-DIAGAREA 93, 107
- DB-INF-APPL 93
- DB-INF-PROG 93
- DB-USER-AREA 93
- dbx 29, 45
- debug 29
- Debugger 45
 - unter Unix-Systemen 30
 - unter Windows-Systemen 33
- DEL ignorieren 37
- Dezimal-Eingabe in KDCDUMP 64
- Dezimalfelder im Dump 105
- DIAGAREA 97
- Diagnose
 - SYSLOG 153
 - Unterlagen 44
- DIAGNOSTIC-Word 233
- Dialog-Terminalprozess
 - beim Testen 36
- Dialogstation (Meldungsziel) 128
- DMS-Fehler 383
- Dokumentation, Wegweiser 12
- DUMP
 - KDCDUMP-Anweisung 67
- Dump
 - Aufbereitung 62
 - Dateien 60
 - Fehlercodes 181
 - unterdrücken 60
- Dump-Auswertung
 - Tabelle anzeigen 78
- DUMP-CONTENT
 - Startparameter 61
- Dump-Fehlercode
 - Grp 286
- Dynamischer Trace 46

- E**
- ED
 - KDCDUMP-Anweisung 68
- EDITOR 68
- Eigene Meldungstexte ausgeben 141
- Eigener Meldungsmodul 125, 134, 141
- Ein-/Ausschalten

BCAM-Trace 49
Eingabe-Nachrichten
Trace 48
Einträge in der Meldungsdatei 128, 398
ENCRYPTION_LEVEL 272
END 144
KDCDUMP-Anweisung 69
ENDMSG 144
Environment Area 92
errno 383
Erzeugen
Meldungstexte (SYSLOG) 155
ETPND
TPR 88
Event Logging (Windows-Systeme) 416
Exit Table 90

F

Fehler
im INPUT-Exit 40
reproduzieren 45, 182
Fehlercode
Programmschnittstelle 39
UTM-Dump 181
Fehlernummer
DMS-Fehler 383
Fehlertexte in der DIAGAREA 102
Fehlerunterlagen 44
FGG, KDCDUMP-Anweisung 69
FIND
KDCDUMP-Anweisung 72
FORMUSER-BUFFER 91
Funktionseinheit 137, 146

G

gdb 29
GEN 144
Gestaltung der Meldungs Ausgabe 134
GF-ENT 120
GF-HDR 120
Grp
Dump-Fehlercode 286
Gruppe Dump-Fehlercode 181

H

Header 129
HELP
KDCDUMP-Anweisung 73, 75
HLL-USER-AREA 91
HP-UX 11

I

Initialisierung der Meldungsbearbeitung 132
INPUT-Exit 101
Inserts 123
ändern 134
K-Meldungen 386
U-Meldungen 395
Interner Returncode KRCRDC 39, 422
Interprozesskommunikation
Trace 56
IO-BUFFER 91
IPC-ANNOS 92
IPC-APPL 91
IPC-APPL-GLOB 91
IPC-BRSE 91
IPC-ELEMENTS 91
IPC-EXTP 91
IPC-FREE-QUEUE 91
IPC-HEADER 91
IPC-LETTER 92
IPC-SEMA 91
IPC-TIMER-ID 91
IPC-TRACE 92

J

JF-1-ENT 120
JF-1-HDR 120
JF-2-ENT 120
JF-2-HDR 120

K

K-Meldungen 157, 398
K316 61
KAA 82
KAA-Trace 57
KB 117
KB (im Dump) 90

- KCRCCC 39, 419
- KCRCDC 39, 422
- KDCCSYSL
 - aufrufen 154
 - Meldungen 155
 - starten 154
- kdccsysl 153
- KDCDEF
 - Meldungen 342
- KDCDIAG
 - Message-Dump-Funktion ausschalten 43
 - Message-Dump-Funktion einschalten 42
 - Testmodus einschalten 41
- KDCDUMP 62
 - ! 64
 - !! 65
 - AFIND 65
 - Anweisungen 63
 - aufrufen 62
 - beenden 69
 - blättern 65
 - DUMP 67
 - EDT 68
 - FGG 69
 - FIND 72
 - HELP 73, 75
 - LIST 74
 - Meldungen 352
 - SH / SYS 77
 - starten 62
 - SYSLST 77
 - TABLE 78
- KDCIPC 56
- KDCKAA 57
- KDCMMOD 134, 141
 - aufrufen 144
 - Eingaben und Ausgaben 143
 - Meldungen 350
- KDCMSGLT (LTERM-Partner) 131
- KDCMSGUS (Benutzer) 131
- KDCMTXT 135
 - aufrufen 136
 - Meldungen 350
- KDCPSYSL 153
 - Meldungen 155, 349
 - starten 155
- KDCREM nach STXIT=OFF 45
- KDCROOT im UTM-Dump 90
- KDCS_C_DEBUG 47
- KDCS-Returncode 39, 419
- kdcsort 50
- Knoten-Anwendung
 - Meldung K190 302
- Konsole 128, 130
- KTA 88
 - Trace 52
- L**
- Lademodule 95
- LANG 126, 133, 152
- Länge
 - Meldung in Systemzeile 149
- Längenbeschränkungen
 - Meldung 148
- ldd 45
- LF-DLK 120
- LF-ENT 120
- LF-HDR 120
- LIB 90
- Linux-Distribution 11
- LIST
 - KDCDUMP-Anweisung 74
- Little Endian 64, 105
- LOAD-MODULE 90
- M**
- Meldungen 129
 - ändern 134, 141
 - auf Konsole ausgeben 141
- KDCCSYSL 155
- KDCDEF 342
- KDCDUMP 352
- KDCMMOD 350
- KDCMTXT 350
- KDCPSYSL 155, 349
- Maximal-Länge 148
- Sprache auswählen (Windows-Systeme) 133, 152

Transaktionsmonitor 157
 U-Meldungen 355
 XAP-TP-Provider 315
 Meldungsaufbereitung 133
 Meldungsdefinitionsdatei 134
 Meldungskatalog 126
 Sourcedatei 127
 Meldungskopf 129
 Aufbau 156
 Meldungsmodul 124, 125
 eigener 134
 Meldungsnummer 123
 Meldungsschlüssel 123
 Meldungstext 123
 ändern 134, 141
 übersetzen 134
 Meldungstexte
 für SYSLOG erzeugen 155
 Meldungsziel 123, 128
 Benutzer-spezifisch 128, 131
 hinzufügen 134
 wegnehmen 134
 Memory Pool Table 90
 Message Mod Table 90
 Message-Dump 41
 Metasyntax 25
 Modifikation von Standardmeldungen 141
 MODMSG 144
 MPUT-BUFFER 90
 MSCF 454
 MSGTAC
 Meldungsziel 128
 MSGTAC-Programm 131
 MSGTAC-Routine 141
 MSGTACT 131
 mtxtin 136

N
 Netzanbindungs-Fehler 44
 NLOAD-MODULE-Tabelle 95
 NLS-Meldungskatalog 124, 126

O
 openUTM-Anwendung, siehe UTM-Anwendung

OPTION
 KDCMMOD 144
 OSS Area 92
 OSS-Trace 53
 OTRACE (Startparameter) 53

P
 Parameter des INPUT-Exits im Dump 101
 PARTNER (Meldungsziel) 128
 PCMX 16
 PID
 Meldungen 129
 PROGRAM-TABL 90
 Protokollierung in KDCDUMP
 SYSLST 77
 Prozess-spezifischer Systemspeicher (KTA) 88
 Prozess-spezifischer Systemspeicher (XAP-TP) 89
 PTERM-Name
 anmelden 37

R
 Rahmendefinitionen 398
 REASON 181
 Red Hat 11
 Reduzierung der DUMP-Information 61
 RESTART-BUFFER 91
 Returncode
 KCRCCC 419
 KCRCDC 422
 Programmschnittstelle 39
 ROOT-TRACE 91
 ROOTDATA 91

S
 sdb 29, 45
 SH
 KDCDUMP-Anweisung 77
 Shared Objects
 ermitteln 45
 SHMPROT Area 92
 SIGBUS 45
 Signal 45
 Signalbehandlung

- ausschalten 45
 - SIGSEGV 45
 - Socket-Partner
 - UTM-Meldung an 130
 - Solaris 11
 - SPAB 90
 - Sprache
 - für Meldungen (Windows-Systeme) 133, 152
 - Sprache der Meldungen 134
 - Stand-alone UTM-Anwendung 9
 - Standardmeldungen 125
 - Standardmeldungsmodul 125, 141
 - Start-Fehlercodes 169
 - Starten
 - KDCCSYSL 154
 - KDCDUMP 62
 - KDCPSYSL 155
 - mit Debugger unter Unix-Systemen 30
 - mit Debugger unter Windows-Systemen 33
 - Startparameter
 - DUMP-CONTENT 61
 - Fehlerunterlagen 44
 - OTRACE 53
 - STXIT 45
 - TEST 30, 33
 - TESTMODE 56
 - Startprozedur
 - Diagnose 44
 - STATION (Meldungsziel) 128
 - Statische Bibliothek 45, 182
 - Status
 - DB-Transaktion 109
 - STDERR (Meldungsziel) 128
 - STDOUT (Meldungsziel) 128
 - STEP 54
 - Summary-Information 121
 - FGG 69
 - FILE 71
 - SUSE 11
 - SYS
 - KDCDUMP-Anweisung 77
 - SYSLINE (Meldungsziel) 128, 149
 - SYSLOG (Meldungsziel) 128
 - SYSLOG-Datei 153
 - aufbereiten 154, 155
 - SYSLST
 - KDCDUMP-Anweisung 77
 - Meldungsziel 147
 - SYSOUT (Meldungsziel) 147
 - SYSTEM-PEND-ER 102
 - System-Protokolldatei SYSLOG 153
 - Meldungsziel 128
 - Systemkommando
 - KDCDUMP 64
 - Systemspeicher
 - KAA 82
 - KTA 88
 - Systemzeile
 - des Terminals 128
 - Meldung 149
- ## T
- Tabelle anzeigen im Dump 78
 - Tabellenausschnitt von Dump ausgeben 74
 - TABLE
 - KDCDUMP-Anweisung 78
 - Table Descriptors 92
 - TAM 93
 - Task 82
 - Task-spezifischer Systemspeicher (KTA) 88
 - Task-spezifischer Systemspeicher (XAP-TP) 89
 - Task-spezifischer Trace-Bereich 107
 - Taste DEL
 - Wirkung 37
 - Tastenkombination CTRL+C
 - Wirkung 37
 - Terminal
 - im Testbetrieb 36
 - TEST 30, 33
 - Testen
 - UTM-Produktivanwendung (Unix-Systeme) 30
 - UTM-Produktivanwendung (Windows-Systeme) 33
 - Workprozess neu starten 31
 - Testhilfen
 - in Unix-Systemen 29
 - TESTMODE

- Dump unterdrücken 60
- Startparameter 56
- Timer Area 92
- Tool
 - KDCDUMP 62
 - KDCIPC 56
 - KDCKAA 57
 - KDCMMOD 134
 - KDCMTXT 135
 - KDCPSYSL 155
- Trace 46
 - BCAM 48
 - dynamischer (UTMTRAC) 46
 - IPC 56
 - KAA 57
 - KTA 52
 - OSS 53
- Transaktionsmonitor
 - Meldungen 157
- TSKM 93
- U**
- U-Meldungen 355
 - ändern 141
- U244 30, 34
- U500 415
- U5xx-Meldungen 415
- Übertragung
 - KDCUPD 355
- Übertragungsfehler
 - Meldung 355
- UF-ENT 120
- UF-HDR 120
- Uhrzeit
 - Meldungen 129
- UL-ENT 120
- UL-HDR 120
- Umgebungsvariable
 - EDITOR 68
 - KDCS_C_DEBUG 47
 - LANG 126
 - UTMTRAC 46
- Unix-Plattform 11
- Unterbrechen KDCDUMP, SH und SYS 77
- Unterlagen für Diagnose 44
- USER-DEST 128, 131
- USER-ROOT 92
- USP-Header 130
- UTM_MSG_DATE 130
- UTM_MSG_PID 130
- UTM-Cluster-Anwendung 9
 - Cluster-Administrations-Journal 452
- UTM-Cluster-Dateien
 - Meldung K190 301
- UTM-DIAGAREA 90, 97
- UTM-Dump 59
 - Tabellendarstellung 59
- UTM-Meldung 123
 - an andere Anwendung 130
 - an Socket-Anwendung 130
 - Aufbau 156
- UTM-Meldungen 157
- UTM-Meldungsziele 128
- UTM-MSG-DATE 130
- UTM-Produktivanwendung
 - testen (Unix-Systeme) 30
 - testen (Windows-Systeme) 33
- UTM-Protokolldatei SYSLOG 123, 153
- UTM-Returncodes 419
- utmdtp 36
- utmpfad 26
- utmpfad/ex 36
- UTMTRAC 46
- V**
- VGM-AREA 92
- W**
- Windows Event Logging 416
- Windows-System 11
- X**
- X-String 64
- XA Area 92
- XAP-TP
 - Systemspeicher 87, 89
- XAP-TP-Provider
 - Meldungen 315

Z

Ziele der K-Meldungen [128](#)

Ziele der UTM-Meldungen [398](#)

Ziele einzelner Meldungen ändern [141](#)