FUJITSU Software

openUTM V6.3

Messages, Debugging and Diagnostics in BS2000 systems

User Guide

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your
opinion on this manual. Your feedback helps us to optimize our
documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation
according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and
user-friendliness, this documentation was created to
meet the regulations of a quality management system which
complies with the requirements of the standard
DIN EN ISO 9001:2008.

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

Copyright and Trademarks

Contents

1 Preface

Modern enterprise-wide IT environments are subjected to many challenges of rapidly increasing importance. This is the result of:

- heterogeneous system landscapes

- different hardware platforms

- different networks and different types of network access (TCP/IP, SNA, ...)

- the applications used by companies

Consequently, problems arise – whether as a result of mergers, joint ventures or labor-saving measures. Companies are demanding flexible, scalable applications, as well as transaction processing capability for processes and data, while business processes are becoming more and more complex. The growth of globalization means, of course, that applications are expected to run 24 hours a day, seven days a week, and must offer high availability in order to enable Internet access to existing applications across time zones.

openUTM is a high-end platform for transaction processing that offers a runtime environment that meets all these requirements of modern, business-critical applications, because openUTM combines all the standards and advantages of transaction monitor middleware platforms and message queuing systems:

- consistency of data and processing

- high availability of the applications (not just the hardware)

- high throughput even when there are large numbers of users (i.e. highly scalable)

- flexibility as regards changes to and adaptation of the IT system

An UTM application can be run as a standalone UTM application or sumultanously on several different computers as a UTM cluster application.

openUTM forms part of the comprehensive **openSEAS** offering. In conjunction with the Oracle Fusion middleware, openSEAS delivers all the functions required for application innovation and modern application development. Innovative products use the sophisticated technology of openUTM in the context of the **openSEAS** product offering:

● BeanConnect is an adapter that conforms to the Java EE Connector Architecture (JCA) and supports standardized connection of UTM applications to Java EE application servers. This makes it possible to integrate tried-and-tested legacy applications in new business processes.

● The WebTransactions member of the openSEAS family is a product that allows tried-and-tested host applications to be used flexibly in new business processes and modern application scenarios. Existing UTM applications can be migrated to the Web without modification.

1.1   Summary of contents and target group

The openUTM manual "Messages, Debugging and Diagnostics in BS2000 Systems" is intended for users, administrators and programmers of UTM applications.

It describes how to debug a UTM application under BS2000 systems, the structure of the UTM dump, behavior in the event of an error, and all the UTM messages and return codes output by openUTM.

Chapter 4 also describes the UTM message system and the options for outputting messages in different languages or to modify the messages supplied with openUTM for specific applications or to redirect messages to different destinations.

This manual assumes a knowledge of openUTM and the BS2000 operating system.

A separate openUTM manual "Messages, Debugging and Diagnostics in Unix Systems and Windows Systems", is available for the Unix operating systems and the Windows operating systems.

1.2 Summary of contents of the openUTM documentation

This section provides an overview of the manuals in the openUTM suite and of the various related products.

1.2.1 openUTM documentation

The openUTM documentation consists of manuals, the online help systems for the graphical administration workstation openUTM WinAdmin and the graphical administration tool WebAdmin, and a release note for each platform on which openUTM is released.

Some manuals are valid for all platforms, and others apply specifically to BS2000 systems, Unix systems or Windows systems.

All the manuals are available as PDF files on the internet at

http://manuals.ts.fujitsu.com

On this site, enter the search term "openUTM V6.3" in the **Search by product** field to display all openUTM manuals of version 6.3.

The manuals are included on the Enterprise DVD with open platforms and are available on the WinAdmin DVD for BS2000 systems.

The following sections provide a task-oriented overview of the openUTM V6.3 documentation. You will find a complete list of documentation for openUTM in the chapter on related publications at the back of the manual on .

Introduction and overview

The **Concepts and Functions** manual gives a coherent overview of the essential functions, features and areas of application of openUTM. It contains all the information required to plan a UTM operation and to design an UTM application. The manual explains what openUTM is, how it is used, and how it is integrated in the BS2000, Unix based and Windows based platforms.

Programming

- You will require the **Programming Applications with KDCS for COBOL, C and C++** manual to create server applications via the KDCS interface. This manual describes the KDCS interface as used for COBOL, C and C++. This interface provides the basic functions of the universal transaction monitor, as well as the calls for distributed processing. The manual also describes interaction with databases.

- You will require the **Creating Applications with X/Open Interfaces** manual if you want to use the X/Open interface. This manual contains descriptions of the UTM-specific extensions to the X/Open program interfaces TX, CPI-C and XATMI as well as notes on configuring and operating UTM applications which use X/Open interfaces. In addition, you will require the X/Open-CAE specification for the corresponding X/Open interface.

- If you want to interchange data on the basis of XML, you will need the document entitled openUTM **XML for openUTM**. This describes the C and COBOL calls required to work with XML documents.

- For BS2000 systems there is supplementary documentation on the programming languages Assembler, Fortran, Pascal-XT and PL/1.

Configuration

The **Generating Applications** manual is available to you for defining configurations. This describes for both standalone UTM applications and UTM cluster applications how to use the UTM tool KDCDEF to

- define the configuration

- generate the KDCFILE

- and generate the UTM cluster files for UTM cluster applications

In addition, it also shows you how to transfer important administration and user data to a new KDCFILE using the KDCUPD tool. You do this, for example, when moving to a new openUTM version or after changes have been made to the configuration. In the case of UTM cluster applications, it also indicates how you you can use the KDCUPD tool to transfer this data to the new UTM cluster files.

Linking, starting and using UTM applications

In order to be able to use UTM applications, you will need the **Using openUTM Applica-tions** manual for the relevant operating system (BS2000 or Unix systems/Windows systems). This describes how to link and start a UTM application program, how to sign on and off to and from a UTM application and how to replace application programs dynamically and in a structured manner. It also contains the UTM commands that are available to the terminal user. Additionally, those issues are described in detail that need to be considered when operating UTM cluster applications.

Administering applications and changing configurations dynamically

● The **Administering Applications** manual describes the program interface for admin-istration and the UTM administration commands. It provides information on how to create your own administration programs for operating a standalone UTM application or a UTM cluster application and on the facilities for administering several different applications centrally. It also describes how to administer message queues and printers using the KDCS calls DADM and PADM.

● If you are using the graphical administration workstation **openUTM WinAdmin** or the Web application **openUTM WebAdmin**, which provides comparable functionality, then the following documentation is available to you:

– A **description of WinAdmin** and **description of WebAdmin**, which provide a comprehensive overview of the functional scope and handling of WinAdmin/WebAdmin. These documents are shipped with the associated software and are also available online as a PDF file.

– The respective **online help systems**, which provide context-sensitive help infor-mation on all dialog boxes and associated parameters offered by the graphical user interface. In addition, it also tells you how to configure WinAdmin or WebAdmin in order to administer standalone UTM applications and UTM cluster applications.

i For detailed information on the integration of openUTM WebAdmin in SE Server's SE Manager, see the SE Server manual **Operation and Administration**.

Testing and diagnosing errors

You will also require the **Messages, Debugging and Diagnostics** manuals (there are separate manuals for Unix systems / Windows systems and for BS2000 systems) to carry out the tasks mentioned above. These manuals describe how to debug a UTM application, the contents and evaluation of a UTM dump, the behavior in the event of an error, and the openUTM message system, and also lists all messages and return codes output by openUTM.

Creating openUTM clients

The following manuals are available to you if you want to create client applications for communication with UTM applications:

● The **openUTM-Client for the UPIC Carrier System** describes the creation and operation of client applications based on UPIC. In addition to the description of the CPI-C and XATMI interfaces, you will find information on how you can use the C++ classes to create programs quickly and easily.

● The **openUTM-Client for the OpenCPIC Carrier System** manual describes how to install and configure OpenCPIC and configure an OpenCPIC application. It describes how to install OpenCPIC and how to configure an OpenCPIC application. It indicates what needs to be taken into account when programming a CPI-C application and what restrictions apply compared with the X/Open CPI-C interface.

● The documentation for the **JUpic-Java classes** shipped with **BeanConnect** is supplied with the software. This documentation consists of Word and PDF files that describe its introduction and installation and of Java documentation with a description of the Java classes.

● The **BizXML2Cobol** manual describes how you can extend existing COBOL programs of a UTM application in such a way that they can be used as an XML-based standard Web service. How to work with the graphical user interface is described in the **online help system**.

● If you want to provide UTM services on the Web quickly and easily then you need the manual **WebServices for openUTM**. The manual describes how to use the software product WS4UTM (WebServices for openUTM) to make the services of UTM applications available as Web services. The use of the graphical user interface is described in the corresponding **online help system**.

Communicating with the IBM world

If you want to communicate with IBM transaction systems, then you will also require the manual **Distributed Transaction Processing between openUTM and CICS, IMS and LU6.2 Applications**. This describes the CICS commands, IMS macros and UTM calls that are required to link UTM applications to CICS and IMS applications. The link capabilities are described using detailed configuration and generation examples. The manual also describes communication via openUTM-LU62 as well as its installation, generation and administration.

PCMX documentation

The communications program PCMX is supplied with openUTM on Unix and Windows systems. The functions of PCMX are described in the following documents:

● CMX manual "Betrieb und Administration" (Unix-Systeme) (only available in German)

● PCMX online help system for Windows systems

1.2.2   Documentation for the openSEAS product environment

The **Concepts and Functions** manual briefly describes how openUTM is connected to the
openSEAS product environment. The following sections indicate which openSEAS
documentation is relevant to openUTM.

Integrating Java EE application servers and UTM applications

The BeanConnect adapter forms part of the openSEAS product suite. The BeanConnect
adapter implements the connection between conventional transaction monitors and
Java EE application servers and thus permits the efficient integration of legacy applications
in Java applications.

● The manual **BeanConnect** describes the product BeanConnect, that provides a JCA
1.5- and JCA 1.6-compliant adapter which connects UTM applications with applications
based on Java EE, e.g. the Oracle application server.
The manuals for the Oracle application server can be obtained from Oracle.

Connecting to the web and application integration

You require the WebTransactions manuals to connect new and existing UTM applications
to the Web using the product **WebTransactions**.

The manuals will also be supplemented by JavaDocs.

1.2.3   Readme files

Information on any functional changes and additions to the current product version described in this manual can be found in the product-specific Readme files.

Readme files are available to you online in addition to the product manuals under the various products at *http://manuals.ts.fujitsu.com*. For the BS2000 platform, you will also find the Readme files on the Softbook DVD.

Information under BS2000 systems

When a Readme file exists for a product version, you will find the following file on the BS2000 system:

SYSRME.<product>.<version>.<lang>

This file contains brief information on the Readme file in English or German (<lang>=E/D). You can view this information on screen using the /SHOW-FILE command or an editor. The /SHOW-INSTALLATION-PATH INSTALLATION-UNIT=<product> command shows the user ID under which the product's files are stored.

Additional product information

Current information, version and hardware dependencies, and instructions for installing and using a product version are contained in the associated Release Notice. These Release Notices are available online at *http://manuals.ts.fujitsu.com*.

1.3   Innovations in openUTM V6.3

The following sections provide more detail on the innovations in the individual areas.

1.3.1   New server functions

Additional UTM system processes for internal tasks

In addition to the processes specified by means of the start parameters, UTM starts up to three additional processes that are reserved for internal openUTM tasks or privileged jobs issued by the administrator.

To permit this, both generation and administration interfaces have been extended:

● Generation, KDCDEF statement MAX

– New operand PRIVILEGED-LTERM, used to identify a specific LTERM as privileged. When a user signs on with administration authorizations, all the user's jobs are considered to be privileged jobs.

– TASKS operand: The maximum value has been reduced to 240 due to the additional system processes.

● KDCADMI administration interface

– Data structure *kc_max_par_str*: New field *privileged_lterm* for the generated privileged LTERM.

– Data structure *kc_tasks_par_str*: New fields *gen_system_tasks* and *curr_system_tasks* for the system processes.

– Data structure *kc_curr_par_str*: New field *curr_system_tasks* for the system processes.

Higher resolution for output of used CPU time

The used CPU time is now output in microseconds for TACs and in milliseconds for USERs. The following interfaces have been changed to support this:

● KDCADMI

– Data structure *kc_tac_str*: New field *taccpu_micro_sec* for the average used CPU time in microseconds.

– Data structures *kc_user_str* and *kc_user_dyn1_str*: New field *cputime_msec* for the used CPU time in milliseconds.

- KDCADM command interface
    - KDCINF type=TAC: TACCPU outputs the average used CPU time in microseconds.
    - KDCINF type=USER: CPUTIME outputs the used CPU time in milliseconds.
- KDCEVAL lists
    - Some times are now output in microseconds in the KDCEVAL lists.

New trace functions

Additional traces can be enabled and disabled during live operation.

- ADMI trace, i.e. trace of the administration program interface (KDCADMI)
- X/Open traces (CPI-C, TX, XATMI)

The following interfaces have been extended to support this:

- Start parameters:

    New start parameters ADMI-TRACE, CPIC-TRACE, TX-TRACE and XATMI-TRACE for enabling traces.

- KDCADMI

    Data structure *kc_diag_and_account_par_str*: New fields *admi_trace*, *cpic_trace*, *tx_trace* and *xatmi_trace* for enabling and disabling traces.

KDCDEF input/output via LMS library elements

In BS2000 systems, it is possible to read KDCDEF statements from LMS library elements and, in the case of inverse KDCDEF, output them to LMS library elements. The following interfaces have been extended to support this:

- Generation
    - KDCDEF statement OPTION: New operand value LIBRARY-ELEMENT(...) in the DATA operand.
    - KDCDEF statement CREATE-CONTROL-STATEMENTS: New operand value LIBRARY-ELEMENT(...) in the TO-FILE operand.
- KDCADMI

    Data structure *kc_create_statements_str*: New fields *lib_name*, *elem_name*, *vers*, *type*, *stmt_type* and *file_error_code*.

● Messages

New messages K234, K519 and K520 when reading KDCDEF statements from LMS library elements and outputting KDCDEF statements to LMS library elements.

Performance enhancements

● UTM cache

The UTM cache has been optimized in order to improve performance during intensive use of the UTM cache (e.g. in the case of extremely extensive service data).

● UTM lock algorithm

The Compare&Swap functionality offered by the operating system is used throughout on open platforms for concurrent access to internal UTM administration data.

● UTM network access

The network access on open platforms has been improved so that delays no longer occur when sending data to UTM partner applications, in particular in low-load situations.

Other changes

● Messages

– The message area for system messages has been increased and now comprises the range from K001 to K399 (previously up to K249). As a result, the following message areas have been moved:

– The message numbers for messages exclusively output by KDCUPD now occupy the range K800 to K899 instead of K250 to K322.

Messages output by KDCUPD and by online import are considered to be system messages and remain unchanged.

– The message numbers for KDCCSYSL and KDCPSYSL messages now occupy the range K600 to K649 instead of K550 to K599.

– New message K235 if name resolution for a computer takes too long.

– The default message destinations for messages K162 and K163 have been changed.

- KDCADMI
    - The fields *auto_connect* in *kc_lpap_str* and *auto_connect_number* in *kc_osi_lpap_str* have the property GPD instead of PD, changes to these fields always have a global effect throughout the application. Any administrative change to the properties "automatic establishment of connection" in the case of LPAP and "number of connections" for OSI-LPAP remains effective beyond the end of the application.
    - New field *max_btrace_lth* in *kc_diag_and_account_par_str* for the maximum length of the recorded data when the BCAM trace function is activated.
- In the case of platforms on which UTM can run in 64-bit mode, KDCUPD makes it possible to migrate from a 32-bit application environment to a 64-bit application environment. At present, UTM only supports 64-bit mode on Unix platforms.
- The Oracle User ID can also be entered in lowercase in the KDCDEF statements DATABASE and RMXA.
- The InstallAware installation procedure is used on Windows systems. As a result, openUTM is supplied in the form of MSI files for Windows systems.
- New sample program ADJTCLT (ADJust Tac-CLass Table)

    Using the C program unit ADJTCLT, users can control how the processes are distributed to the TAC classes in the light of the current total number of processes and the current number of asynchronous processes. To do this, the user creates a table containing the desired settings. The settings must be chosen in such a way that there is always at least one process free to perform other tasks, such as end-of-transaction processing for distributed transactions for example.

1.3.2 Load simulation with "Workload Capture & Replay"

Thanks to the new Workload Capture & Replay function, it is possible to record UTM application communications with UPIC clients and then replay these in combination with adjustable load profiles. In this way, it is possible to test the behavior of the UTM application at high loads under real-life conditions.

Workload Capture & Replay consists of the following components:

- *UPIC Capture*: Records communication with the UPIC client.

    The trace function BTRACE (BCAM trace), which is present on all the server platforms, is used to record a UPIC session.

- *UPIC Analyzer*: Used to analyze the recorded communication.
- *UPIC Replay*: Used to replay the recorded UPIC session with different load parameters (speed, number of clients).

UPIC Analyzer and *UPIC Replay* are only available on 64-bit Linux systems and are supplied with openUTM Client (UPIC).

openUTM for Unix and Windows systems also comes with the utility program *kdcsort*. You can use *kdcsort* to sort the communication recorded by BTRACE over time if the UTM application ran with more than one process during the recording period and multiple process-specific files have therefore been generated.

1.3.3 New client function

On Windows systems, UPIC Client is available in both a 32-bit and a 64-bit variant.

1.3.4 New and modified functions for openUTM WinAdmin

- WinAdmin supports all the new features of UTM V6.3 relating to the administration program interface. These include, for example, the new trace functions, the writing of KDCDEF statements to library elements on Inverse KDCDEF runs in BS2000 or the display of a user's used CPU time in milliseconds.

- Introduction of a lifetime for statistical values in order to limit the number of statistical values stored in the configuration database.

1.3.5 New functions for openUTM WebAdmin

Additional functions

WebAdmin now provides additional functions that go beyond the functionality available in the KDCADMI administration interface and which were previously available only in WinAdmin:

- Display of message queues (DADM functionality)

- Administration of statistics collectors and tabular display of the associated values (including the new "Lifetime for statistical values" function).

- Depiction of statistics in graphical form (graphs)

- Execution of threshold actions for statistics collectors

Support for new features in openUTM V6.3

WebAdmin supports all the new features of UTM V6.3 relating to the administration program interface. These include, for example, the new trace functions, the writing of KDCDEF statements to library elements on Inverse KDCDEF runs in BS2000 or the display of a user's used CPU time in milliseconds.

Integration in SE Server

WebAdmin can be installed as an add-on in the management unit (SE Manager) of an SE Server. It then provides much the same range of functions as when operated outside of the SE Manager.

1.4 Notational conventions

Metasyntax

The table below lists the metasyntax and notational conventions used throughout this manual:

Representation	Meaning	Example
UPPERCASE LETTERS	Uppercase letters denote constants (names of calls, statements, field names, commands and operands etc.) that are to be entered in this format.	`LOAD-MODE=STARTUP`
lowercase letters	In syntax diagrams and operand descriptions, lowercase letters are used to denote place-holders for the operand values.	`KDCFILE=filebase`
lowercase letters in italics	In running text, variables and the names of data structures and fields are indicated by lowercase letters in italics.	*utmpath* is the UTM installation directory
`Typewriter font`	Typewriter font (Courier) is used in running text to identify commands, file names, messages and examples that must be entered in exactly this form or which always have exactly this name or form.	The call tpcall
{ } and		Curly brackets contain alternative entries, of which you must choose one. The individual alternatives are separated within the curly brackets by pipe characters.	`STATUS={ ON	OFF }`
[ ]	Square brackets contain optional entries that can also be omiited.	`KDCFILE=( filebase` `[, { SINGLE| DOUBLE} ] )`
( )	Where a list of parameters can be specified for an operand, the individual parameters are to be listed in parentheses and separated by commas. If only one parameter is actually specified, you can omit the parentheses.	`KEYS=(key1,key2,...key`*n*`)`
Underscoring	Underscoring denotes the default value.	`CONNECT= { A/YES	`<u>`NO`</u>` }`

Representation	Meaning	Example
abbreviated form	The standard abbreviated form of statements, operands and operand values is emphasized in boldface type. The abbreviated form can be entered in place of the full designation.	TRANSPORT**-SEL**ECTOR=c'C'
...	An ellipsis indicates that a syntactical unit can be repeated. It can also be used to indicate sections of a program or syntax description etc.	Start KDCDEF : : OPTION DATA=statement_file : END

Other symbols

Indicates notes that are of particular importance.

Indicates warnings.

2 Debugging and error diagnosis

This chapter tells you

– how to debug a UTM application
– how to use return codes and messages for diagnostic purposes
– which traces you can use for diagnostic purposes.

2.1 Debugging UTM applications

You can debug a UTM application interactively in the dialog (BS2000 system, TIAM timesharing mode).
This enables you to use the operating system debugging aids and the corresponding compiler (in particular, see the language-specific user guides "AID (Advanced Interactive Debugger)" and "BS2000 Interactive Debugging Aid").

Comprehensive interactive debugging functionality is available to you:

– all UTM function levels can be debugged, including functions for distributed processing

– all partner connections can be included for debugging

– debugging can be done in single task mode as well as in multitasking mode

– printer outputs can be debugged

– dialog I/Os for the BS2000 debugger and I/Os for the UTM application are entered and produced in different windows (or on different terminals) and do not interfere with each other; the user dialog will then run just like in live operation

– you can update your application to a new UTM version and test it in parallel to the "old" version. In parallel operation, you simply need to make sure that all the transport system endpoints in your application (MAX APPLINAME, BCAMAPPL, T-SEL of ACCESS-POINT) have different names.

– the debugged application can be released for live operation without any additional changes.

– in a dialog task, you can only debug applications that share the same UTM version, i.e. you cannot start by debugging a UTM application with an older UTM version and then debug a version V6.3 application or vice versa.

2.1.1   Preparations for debugging in the dialog

Applications started in the dialog are intended exclusively for debugging purposes. The differences to applications started with ENTER-JOB are that tasks are not automatically started afterwards and that UTM-STXIT handling can be switched off.

Symbolic debugging

If you want to carry out symbolic debugging, then you must compile the program such that the compiler creates a list for symbolic debugging (LSD). In COBOL, for example, you do this by specifying COMOPT SYMTEST=ALL. The LSD information does not have to be linked statically and can be loaded from your module library when needed.

You should not link the LSD information statically because you can load it dynamically as and when required. You can then put the debugged programs into live operation immediately.

Exchanging program units

Specifying LOAD-MODE={STARTUP | ONCALL} in the LOAD-MODULE statement can be advantageous when debugging your program units and event exits. You do not need to link the application after making a program change and recompiling.
You only have to relink the LLM that contains the modified program unit. If the LLM only consists of this program unit, then you do not have to relink. You activate the modified program unit by restarting the application or exchanging the LLM while the program is running (KDCPROG LOAD-MODULE= ..., VERS=... command or via the program administration interface (opcode=KC_MODIFY_OBJECT) or via WinAdmin/WebAdmin.

Parallel operation for upgrading to a new version

If you make use of parallel operation, when upgrading to a new version of UTM you can minimize the effort involved as follows:

– Install the new version of UTM in addition to your current version.

– Link your application program (with a different name) to the new version (see the openUTM manual "Using openUTM Applications under BS2000 Systems".

– Carry out debugging in parallel with live operation.

By conducting I/O for the dialog task and the application in two different windows (or on two different terminals), the I/O from the dialog task and the application do not interfere with each other. The user's dialog with the application runs just like in live operation and debugging statements for the application task are input and evaluated in the corresponding window.

You do not need a special generation for debugging, i.e. you can also use the debugged application program in live operation without having to recompile and relink it.

Number of tasks during debugging

For debugging purposes, it is normally sufficient to start the application with one task only. If you want to start an application interactively with two or more tasks, please note the following points:

– It is not possible to start tasks in both interactive **and** batch mode.

– Continuation tasks cannot be started with an administration command, but only "manually" at a terminal or in a separate window.

– The AID commands must be repeated for every task because you cannot determine which task of the application will process a user requirement. The required AID commands can be inserted in the start procedure to prevent having to repeatedly enter the commands when debugging in multitasking mode.

– If you debug and shut down an OSI TP application in multiple dialog tasks then the application should not be started again in the same dialog tasks since this can result in start-up errors or the abnormal termination of the application.

A dialog task only works for the application as long as the program remains loaded. Following program termination, you can continue to work with the task in BS2000 timesharing mode as long as you want. If you are working with only one task, each program termination also causes the application to terminate.

2.1.2   **Starting the application and entering debugging commands**

You must carry out the following steps if you want to start an application in the dialog for debugging purposes:

1. Open a $DIALOG session and start a dialog task for your UTM application. Do this just like for live operation but do not start the application, just load it with LOAD-EXECUTABLE-PROGRAM. The corresponding window will be called the *task window* in the following. LOAD-EXECUTABLE-PROGRAM loads all statically linked programs. If, for example, you want to test a program unit for a Start-Exit, you can now enter the corresponding test commands.

2. Start the program with %RESUME. openUTM now loads all the programs in load modules that were generated with LOAD-MODE=STARTUP. This also includes shareable components (LOAD-MODE=(POOL/POOL,STARTUP).

3. Press the K2 key so that the program is aborted and you can enter debugging commands.
    The K2 key does not take effect until the UTM process is in the non-privileged state of BS2000 (TU). This may not be the case until you have executed step 4.

4. Connect to your UTM application via a UTM client and, if necessary, sign on at the UTM application via a user ID. Next start a program unit or make your first user input.
    Now you can enter debugging commands for the application program in the *task window*. Please note that load modules generated with LOAD-MODE=ONCALL are not loaded until they are called in a task. Only then can you enter debugging commands for these programs.

5. If you want to debug using symbols and the LSD information is not statically linked to the program, then you must assign the libraries that contain the LSD information with
    %SYMLIB *symlib1, symlib2 ...*
    If you use variable names with uppercase and lowercase letters in your program (C/C++ program units) and you want to refer to these names, then you must change the predefined settings for AID using the %AID LOW command.

    Now you can enter your debugging commands in the *task window*, for example with
    %INSERT S=TA4DB@.mput_call to set a breakpoint.

    You resume execution of the UTM application program with RESUME-PROGRAM.

6. After the dialog step has been processed, the UTM client receives the response to your input just like in live operation.

If you want to start an additional dialog task, then you need to repeat steps 1 and 2, then enter something from the UTM client and then carry out the actions in step 4 again. If you have specified "TASKS=1" in the start parameters, then you will need to increase the number of tasks via administrative measures first, otherwise start error 31 will occur. Please

note that if you are testing in dialog mode then the second task is a UTM system task and that it may therefore be necessary to start three work processes if, for example, programs are to be tested with PGWT.

Please note that you **cannot** start any batch tasks afterwards.

The required AID commands can be inserted in the start procedure to prevent having to repeatedly enter the commands (the breakpoints must be set for each task) when debugging in multitasking mode.

You can connect additional UTM clients to your UTM application at any time, regardless of how many tasks the application runs with.

Notes

The following special points arise when an application is started in the dialog:

● The KDCS call PEND ER does not terminate the program. Thus, the program is not reloaded either. The user exit SHUT is **not** called. If you want to continue debugging with a newly loaded program, you have to terminate and restart the program yourself. If you are working with one task only, program termination also causes the application to terminate.

● BS2000 system does not give openUTM dialog tasks priority over other timesharing tasks. The tasks do not run as TP tasks, but as normal dialog tasks. The cache memory is not made resident (see the openUTM manual "Generating Applications").

● It is not possible to exchange the application program via the administration. However, LLM's can be exchanged (see "Exchanging program units" on page 26).

● The TASKS= start parameter only sets the upper limit of the number of tasks, but does not cause the follow-up task to be started. Every task started allocates a timesharing process in BS2000 system and is therefore also normally a BS2000 session.

● The command KDCAPPL TASKS=n only causes the value "n" to be entered as a new maximum value for the number of tasks (if "n" is greater than the number of tasks started). Further information is contained in the openUTM manual "Generating Applications".

● You should analyze or save (recatalog) a UTM dump before the next start since the dump file can otherwise be overwritten.

2.1.3 STXIT response when debugging in the dialog

You have the option of switching off UTM STXIT handling during debugging using the start parameter STXIT=OFF.

If you have specified STXIT=OFF, your program will be aborted when an error occurs at the location where the error has occurred and you can diagnose the error with the appropriate debugging statements.

⚠ **CAUTION**!
If you have specified STXIT=OFF, no transactions will be rolled back when an error occurs. This can lead to inconsistencies in the data. You should therefore never set this parameter when accessing databases used in live operation!

2.1.4 Testing for errors when accessing the address zero

openUTM provides a function that allows you to test for errors when the program units of the application access the address zero. The function causes openUTM to set up a local task memory pool at the address zero. It can therefore only be used in applications that do not themselves work with a memory pool at the address zero.

Activating/deactivating the test function

To activate the test method, you set up a FILE-LINK to the name KDCWRPR in the start procedure or in the ENTER job:

```
/SET-FILE-LINK LINK-NAME=KDCWRPR
```

If an error occurs when setting up the memory pool because, for example, there is already a memory pool working at the address zero in the application, openUTM outputs message K078 at startup:

```
K078 ENAMP xxxxxxxx. In KDCRTST.
```

In this case, the method cannot be used.

You can deactivate the test method again by restarting the application without the above SET-FILE-LINK command.

Accessing the address zero

If the address zero is accessed while the application is running, the program unit is terminated by an STXIT event (STXIT with IW = 48 or 5C).

The program unit can then be modified.

2.1.5 Example

The following trace listing shows the beginning of a debugging session in the dialog.

i	This example, in particular with regard to the FILEBASE start parameter, only applies to standalone UTM applications. A node application in a UTM cluster application can also be tested in the dialog.

The application is started in the dialog with the START-APPLI command procedure.
The TAC DB4TP1 is called in the user dialog. The TAC is implemented in the C program TA4DB.
The symbolic information on this program is contained in the PLAM library OML.TA4DB.

The DB4TP1 TAC outputs the message received together with general information such as the service start or follow-up TAC in line mode. If the message contains the string "@end", then the service is terminated.

Step1: Start the application in the task window

```
/CALL-PROC NAME=START-APPLI,LOGGING=*YES
 1  1 /SET-PROCEDURE-OPTIONS IMPLICIT-DECLARATION=*NO,-
       DATA-ESCAPE-CHAR=*STD,JV-REPLACEMENT=*AFTER-BUILTIN-FUNCTION
 7  1 /WRITE-TEXT TEXT='** Procedure starts openUTM - application DBAPP **'
** Procedure starts openUTM - application DBAPP **
 9  1 /ASSIGN-SYSDTA TO-FILE=*SYSCMD
11  1 /SET-FILE-LINK FILE=DBAPP.SYSLOG(+1),LINK=SYSLOG,SHARED-UPDATE=*YES
14  1 /RELEASE-SUBSYSTEM-SPACE
%  ESM0254 COMMAND 'RELEASE-SUBSYSTEM-SPACE' COMPLETELY PROCESSED
16  1 /START-EXECUTABLE-PROGRAM -
          FROM-FILE=*LIB-ELEM(LIB=plam-lib,ELEM=start-llm)
%  BLS0523 ELEMENT 'DBAPPT', VERSION '630' FROM LIBRARY
   ':4OSQ:$userid.LIB.DBAPP.TD' IN PROCESSING
%  BLS0524 LLM 'DBAPPT', VERSION '630' OF '2015-01-11:13:21:55' LOADED
22  1 /%R
K044 Please enter start parameters
.UTM START FILEBASE=DBAPP
.UTM START TABLIB=NEWV5,ROOTNAME=DBAPPT
.UTM START TASKS=1,ASYNTASKS=0,STXIT=ON
.UTM END
K136 (First) SYSLOG file is 4OSQ:$userid.DBAPP.SYSLOG(*0001)
K044 Please enter start parameters
.FHS MAPLIB=MAPLIB=$userid.SYSLIB.UTM.063.EXAMPLE
.FHS ISTD=RUNP
END
FC01 FHS Version 08.3A00 ready
K051 Successful cold start for application DBAPP under V06.3A00/BS2000 (390)
/32 Bi
K156 UTM-UPIC encryption function available: N
K052 Startup completed - task 6HVK activated for application DBAPP version 0
;System-Task: N
```

Step 2: The user signs on to the UTM application (user dialog window)

Step 3:   **Press the K2 key** in the task window

User dialog window

```
K008  Sign-on accepted - input please
DB4TP1 ** message 1 **
```

Step 4:   Enter commands

Task window

```
%  CMD0170 DO YOU WANT TO INSERT COMMANDS? REPLY (Y=YES; N=NO)?
Y
%  SSM2014 TASK IN ESCAPE MODE AT LEVEL NUMBER '1'
%SYMLIB OML.TA4DB
%AID LOW
%IN S=TA4DB@.TA4DB
%R
STOPPED AT SRC_REF: 148, SOURCE: TA4DB@, PROC: TA4DB
%R
```

User dialog window

```
_____
Service start with TAC = DB4TP1


** message 1 **


Follow-up TAC= DB4TP1
_____
```

Step 5:   Stop at the breakpoint

User dialog window

```
** message 2 * @end **
```

Task window

```
STOPPED AT SRC_REF: 148, SOURCE: TA4DB@, PROC: TA4DB
%t 10
148                         EXT.PROC START    , BLOCK START, ASSIGN
156                 ASSIGN
157                 ASSIGN
159                 ASSIGN
163                  CALL
164                 CALL
165                 ASSIGN
166                 ASSIGN
168                 CALL
170                  IF
STOPPED AT SRC_REF: 170 , SOURCE: TA4DB@ , PROC: TA4DB , END OF TRACE
%d ca->ca_head.kcpr_tac
ca_area.ca_head.kcpr_tac( 0: 7)
( 0) |D|  ( 1) |B|  ( 2) |4|  ( 3) |T|  ( 4) |P|  ( 5) |1|  ( 6) | |  ( 7) |
|
%d ca->ca_head.kcpr_tac %x1
CURRENT PC: 0106E396 CSECT: TA4DB@ ****************************************
V'0101B3E0' = ca_area.ca_head.kcpr_tac + #'00000000'
0101B3E0 (00000000) C4C2F4E3 D7F14040               DB4TP1
%R
```

Step 6: End debugging

User dialog window

```
--------------------------------------------------------------------------

End of service with TAC = DB4TP1
** message 2 * @end **


--------------------------------------------------------------------------
KDCSHUT NORMAL
```

Task window

```
K056 Task 6HVK terminated
K138 SYSLOG file :40SQ:$userid.DBAPP.SYSLOG(*0001) closed
K057 Application run terminated
```

2.2 Error diagnosis

This section tells you

– which return codes the program interface supplies
– how openUTM signals errors by means of messages
– which documentation has to be produced in the event of errors
– which traces you can use for diagnostic purposes

For a description of a UTM dump and how to evaluate it, please refer to chapter "The UTM dump" on page 53.

2.2.1 Return codes at the program interface

Following each KDCS call (except for PEND), openUTM returns the following return codes and IDs in the return field of the communication area:

– the KDCS return code and

– the internal return code.

KDCS return code in the KCRCCC field

Please note the following points:

– If several errors occur at the same time, the ones in the highest category are displayed. The lowest return code is not always specified within a category (different to DIN 66 265).

– The exact meaning of the KDCS return codes is described in the openUTM manual „Programming Applications with KDCS" for each KDCS call. A summary of all KCRCCC return codes begins on page 423.

Internal openUTM return code in KCRCDC

The internal return code contains a more accurate specification of the error than the KDCS return code in KCRCCC. Usually, this return code is set with the KDCS return codes 40Z or 70Z (system or generation error). You can find the precise meaning on page 426ff.

The internal openUTM return code is not part of the standardized KDCS interface.

2.2.2 UTM messages in response to program errors

Program errors are formatting errors, errors in the programming of the KDCS interface (see KDCS return code) or errors that lead to STXIT events. openUTM then generates messages that are output by default to SYSOUT, SYSLST and generally also to SYSLOG (see f).

Formatting errors

Whenever there are formatting errors, openUTM issues message K087 and - if the formatting system requests it - message K153. If there are formatting errors when an asynchronous message is output, openUTM also issues message K063. If there are formatting errors in the dialog, openUTM issues message K015 in cases where the formatting error does not lead to abnormal termination of the service.

When errors occur in the FORMAT event exit, openUTM enters the return code FRxx in the KCRCDC field. The FORMAT exit defines the meaning of these return codes.

If serious formatting errors occur while formatting with FHS, openUTM reports these with an internal return code KCRCDC = FHxx, e.g. 'FH01'. In this case, the message contains the 4-byte, printable, secondary FHS return code described in the manual "FHS Format Handling System for openUTM, TIAM, DCAM".

These messages contain the following entries:

● `K015 Formatting error &RCDC &RCF2A – input please`

   contains the entries

```
&RCDC  :    internal return code in the KCRCDC field
&RCF2A :    secondary FHS return code or VTSU return code
```

● `K063 Formatting error &RCF1A &RCF2A for format &FMTN asynchronous message`
  `for LTERM = &LTRM`

   contains the entries

```
&RCF1A :  internal return code in the KCRCDC field
&RCF2A :  secondary FHS return code
&FMTN :   format name
&LTRM :    name of the LTERM to which the message was sent
```

● `K087 Error in the interoperation of UTM and format handling system:`
  `opcode = &OPCD3, return code = &ERCD4 &ERCD5`

   contains the entries

```
&OPCD3 :    operation code at the interface IUTMFORM
&ERCD4 :    internal return code FHxx in the KCRCDC field
&ERCD  :    secondary FHS return code
```

Example

Message

```
K015   Formatting error FH01 080C — input please
```

means that the format is not available (see also sample dump starting on). In this way, an MSGTAC service which is intended to react to a formatting error can access the secondary return code simply and conveniently.

Abnormal termination of a service

If a dialog service is terminated abnormally, openUTM issues message K017. If an asynchronous service is terminated abnormally, it issues message K055.

These messages contain return codes that indicate the cause of the error:

● `K017 Service &TCVG terminated by UTM (&RCCC/&RCDC &RCF2A) — input please`

The entries in the message have the following meanings:

```
&TCVG  :   TAC with which the service was started
&RCCC  :   KDCS return code in the KCRCCC field
&RCDC  :   internal return code in the KCRCDC field
&RCF2A :   secondary FHS return code
```

● `K055 Asynchronous service &ATAC1 terminated by UTM; KCRCCC= &RCCC ; KCRCDC=`
  `&RCDC ; USER= &USER ; LTERM= &LTRM`

The entries in the message have the following meanings:

```
&ATAC1 :   TAC with which the asynchronous service was started
&RCCC  :   KDCS return code in the KCRCCC field
&RCDC  :   internal return code in the KCRCDC field
&USER  :   user ID that created the asynchronous service
&LTRM  :   LTERM partner from which the asynchronous service was generated
```

Errors in the INPUT exit

● When there are errors in the INPUT exit, openUTM generally issues message K098 to the terminal. You will find an explanation of the error codes contained in K098 on .

STXIT events

● When STXIT events occur, openUTM issues messages K102 and K103 and terminates the service abnormally. The messages are described on .

2.2.3 Addressing errors

openUTM overwrites the first 4 KB of the KDCROOT table module and sets up write protection for the first page of KDCROOT to support the search for addressing errors in application programs. This reveals impermissible accesses to the table module (address error, PEND ER). Such accesses occur, for example, as the result of incorrectly transferred addresses (KB, SPAB, etc.) or incorrect register specifications.

Errored accesses to the address zero

If you have activated the test function described on then the program unit is aborted by an STXIT event (STXIT with IW = 48 or 5C) if address 0 is accessed during the operation of the application.

The program unit can then be modified.

2.2.4 Diagnostic dump with defined messages/events

You can cause a diagnostic dump, known as a message dump, to be generated when a certain event occurs. The dump ID depends on the event type.

A message dump is created only by the task in which the event occurs. The UTM application is not terminated. In order to take a message dump like this, you must switch on test mode for the application and set the event at which the message dump is to be taken. You can do both of these things by means of start parameters or using the administration functions.
You can specify the following events:

– the output of a specific K message
– the occurrence of a specific KDCS return code (CC or DC) in a program unit run
– the occurrence of a specific SIGN status when a user signs on

Activating test mode

● using the relevant start parameter

   `.UTM TESTMODE = ON`

● using the relevant administration command

   `KDCDIAG TESTMODE = ON`

● via the administration program interface (e.g. via WinAdmin or WebAdmin)

   In the object type KC_DIAG_AND_ACCOUNT, specify:

   `testmode='Y'` (**data structure** *kc_diag_and_account_par_str*)

Activating and resetting the message dump function

You can activate and reset the message dump function by means of a start parameter or using the administration functions. The function is deactivated by default at application startup.

Activating the message dump function

● using the relevant start parameter

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

   This causes a message dump to be created as soon as the event occurs.

   You can only specify one event when you activate the function using a start parameter. It is not possible to specify inserts for a message.

● using the relevant administration command

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

   In the same way, you can use the parameters `DUMP-MESSAGEx` (where x = 1, 2, 3) to specify up to three different events for generating a message dump. In this case, `DUMP-MESSAGE` is synonymous with `DUMP-MESSAGE1`.

   You can specify up to three inserts as additional constraints for the event "output of a specific K message" (parameter `INSERTx` where x = 1,2,3). For further details, see `KDCDIAG` in the openUTM manual "Administering Applications".

● via the administration program interface (e.g. via WinAdmin or WebAdmin)

   In object type KC_DIAG_AND_ACCOUNT, specify the event and the event type in the *event* and *event_type* fields (data structure *kc_diag_and_account_par_str* with the corresponding substructures). You can also specify up to three inserts as a condition. At the program interface, you can specify one event per call. For further details, see the description of KC_DIAG_AND_ACCOUNT in the openUTM manual "Administering Applications".

event-type specifies the event type and *event* specifies a particular event for which the message dump is to be generated. You can specify the following events:

● Output of a specific K message (*event-type* = `MSG`)
  Specify the UTM message number *Knnn* as the *event*.
  A dump is generated each time the message number occurs until such time as you reset the message number. Only one dump is generated for the message numbers K043, K061, K062. The message number is then automatically reset.

● Occurrence of a particular compatible KDCS return code (*event-type* = `RCCC`)

Specify the number of the compatible KDCS return code (KCRCCC) as the *event* e.g. 14Z. If the return code occurs during a KDCS call, only one dump is generated and the parameter `DUMP-MESSAGE[x]` is reset to `*NONE`.

● Occurrence of a particular incompatible KDCS return code (*event-type* = `RCDC`)
  Specify an incompatible KDCS return code (KCRCDC) as the *event*, e.g. KD10.
  If the return code occurs during a KDCS call, only one dump is generated and the parameter `DUMP-MESSAGE[x]` is reset to `*NONE`.

  Note
  In the case of all KDCS return codes ≥ 70Z and the associated incompatible KDCS return codes, when no PENDER dump is written (e.g. 70Z/K316), no message dump is created either.

● Occurrence of a particular sign-on status (*event-type* = `SIGN`)
  Specify the SIGNON status code in the form *xyy* as the *event* (e.g. U05):
  – *x* corresponds to the value in KCRSIGN1, with U, I, A or R being possible values.
  – *yy* corresponds to the value in KCRSIGN2

If the status code occurs during a SIGN call, only one dump is generated and the parameter `DUMP-MESSAGE[x]` is reset to `*NONE`. This happens regardless of whether or not a sign-on service is generated in the application.

Deactivating the message dump function

● using the relevant start parameter

  `.UTM START DUMP-MESSAGE = *NONE`

● using the relevant administration command

  `KDCDIAG DUMP-MESSAGE = *NONE`

  The relevant parameter must be set to *NONE for events activated using the `DUMP-MESSAGEx` parameters.

● via the administration program interface (e.g. via WinAdmin/WebAdmin)

  In the object type KC_DIAG_AND_ACCOUNT, reset all the events by specifying *event_type=NONE* for each event.

2.2.5 Producing error documentation

In the event of errors, the following documentation is required for diagnostic purposes:

- A detailed description of the error situation and an indication as to whether and how the error can be reproduced.

- The BS2000 operating system version number with correction level.

- openUTM-specific documentation:
  - UTM dumps; you must take care to ensure that *all* dumps from *all* tasks which were active at the time the error occurred are supplied
  - all available traces. When reproducing errors, TESTMODE=ON should be activated, where TRACEREC should be set to at least 3000 (when openUTM-D is used: at least 10000).
  - in the case of warm start errors, errors in the KDCUPD and if the application aborts with PMIO22, the file(s) of the KDCFILE are required
  - openUTM version number with correction status
  - list of control statements for KDCDEF
  - linkage editor listing for the application program
  - compilation listings for all affected program units
  - the SYSLOG file(s)
  - user dumps for errors such as XT48, XT58..., for example
  - SYSLST and SYSOUT logs

- In the case of errors in UTM cluster applications, then the following documents are also required::
  - All files that are global to the cluster, log files (and DUMPs) for all node applications
  - the cluster configuration file and, in the case of administrative problems, all the administration journal files with the suffix JKAA, JRN1, JRN2.
  - in the case of problems caused by interactions between the node applications, the log files of all the other node applications
  - The start procedure and the procedures specified as EMERGENCY-CMD and FAILURE-CMD during generation
  - in the case of user problems (e.g. sign-on problems), also the cluster user file (i.e. the file with the suffix UTM-C.USER)

- Plus, for errors associated with FHS:
  - specification of the FHS version used, with correction status
  - format printout with IFG
  - ready-to-use format module in file form
  - user dumps and UTM dump, if available

- For errors associated with databases: Please refer to the release notices for the database systems in question.

2.2.6 Traces

Alongside the traces in the UTM dump, you can also utilize the following traces for diagnostic purposes for openUTM:
– BCAM trace in openUTM
– OSS trace for OSI TP
– KTA trace in file
– ADMI trace, i.e. trace of the administration program interface (KDCADMI)

2.2.6.1 Logging of KDCS calls from a C/C++ program unit

This function can be used to log all KDCS calls from C program units. For this to be possible, the C program unit must use the macros from the header file kcmac.h.

Enabling logging

Set the job variable link *KDCSCDB.

The job variable link is evaluated each time a UTM task is started. By default, logging is output to SYSOUT. If logging is to be written to a file, you must set the constant KDCS_DEBUG_FP to the value of a global FILE variable before including kcmac.h. You enter the value for this variable in the Start-Exit.

Default: The calls are not logged.

2.2.6.2 BCAM trace in openUTM

The BCAM trace function of openUTM allows logging of all connection-related activities within a UTM application.

Contents of the BCAM trace

The following types of trace record are written:
– Parameter block: The BCAM parameter blocks of the calls REQCON, ACCON, REJCON, DISCON and also the BCAM parameter blocks of the calls RECLET and SENDLET which supplied a return code or the RSO parameter blocks of the calls of the RSO interface
– Announcement: All connection-related announcements
– Connection letter
– Communication with TS application of socket type: Parameter blocks for connection request, connection response and disconnect request functions at the socket interface
– Message: all input/output messages

Each trace record contains the following entries in the specified sequence:

1. Time stamp

2. BCAMAPPL name

3. PTERM/MUX/CON name

4. PROCESSOR name

5. LTERM/LPAP name

6. USER name

7. Type of the trace records (see above):

    –   announcement
    –   parameter block
    –   connection letter
    –   TCP/IP record
    –   message (input/output messages)

8. Up to 32680 bytes of data (depending on the record type and the value of the *length* operand of the BTRACE start parameter)). With longer data elements (such as input/output messages) the behavoir depends on the maximum recording length set for the data.

    –   If a value smaller than the maximum of 32680 is set then the trace record contains the first length/2 bytesand the last length/2 bytes of the data element.

    –   If the maximum value of 32680 is set then the function attempts to compress the data during writing. If the data cannot be compressed to a length of 32680 bytes then the written trace record consists of only the first 16340 bytes and the last 16340 bytes of data.

Installation notes

In the UTM start procedure, a trace file must be created for each task and the link name KDCBTRC assigned to this file by means of the SET-FILE-LINK command.

Each task in a UTM application writes the trace records to a separate file. Therefore, the trace files must be created in such a way in the start procedure that they are given different file names. To this end, the TSN of the task can be appended to the file name for example.

Using SDF-P, this can be done in the following manner:

```
/CREATE-FILE FILE-NAME = &(FILEBASE//'.BTRC.'//TSN()) -
/ , SUPPORT=*PUBLIC-DISK(SPACE=*RELATIVE -
/ (PRIMARY-ALLOCATION=<integer_1..2147483647> -
/ ,SECONDARY-ALLOCATION=<*STD or integer_0..32767>))
/ADD-FILE-LINK FILE-NAME = &(FILEBASE//'.BTRC.'//TSN()) -
/ ,LINK-NAME = KDCBTRC [ , BUFFER-LEN = *STD(SIZE=<1..16>)
```

Here, the procedure parameter FILEBASE should contain the base name of the UTM application.

As of a maximum trace record length of 1960, you must also specify the BUFFER-LENGTH parameter with the corresponding number of blocks. When doing so, note that the primary allocation must be at least twice as large as the block size and that the secondary allocation must be either zero or at least as large as the block size.

Maximum trace length	BUFFER-LENGTH=
> 1960	STD(2)
> 4008	STD(4)
> 8104	STD(8)
> 16296	STD(16)

If SDF-P is not available, the same result may be achieved by using job variables or EDT procedures.

Activating/deactivating the BCAM trace

The BCAM trace can be activated and deactivated by means of a start parameter or an administration command. By default, the function is inactive on starting the application.

● Activation/deactivation by start parameter

$$.\text{UTM START,BTRACE} = \begin{Bmatrix} \text{ON/OFF} \\ (\text{ ON / OFF, length}) \end{Bmatrix}$$

In this way, the trace function is activated (ON) or left inactive (OFF) on starting the application.

You can also specify the maximum length of the data to be recorded.
Minimum: 32
Maximum: 32680
Default value: 256

This maximum length can only be defined via start parameters.

If you use the BCAM trace for the UPIC Capture function (see openUTM manual "Using openUTM Applications under BS2000 Systems") then it is advisable to use the maximum value.

● Activation/deactivation by administration command

```
KDCDIAG BTRACE=ON/OFF [,LTERM=lterm-name/LPAP=lpap-name
                MUX=(mux-name,processor-name,bcamappl) / USER=user-name]
```

In this way, the trace function is activated/deactivated while the application is running.
If an LTERM, LPAP or MUX name is specified, only the events associated with this
connection are recorded.
If a USER name is specified, only the events associated with this user ID are listed.

● Activation/deactivation via administration program interface (e.g. via WinAdmin or
WebAdmin)

In the object type KC_DIAG_AND_ACCOUNT, specify:

```
bcam_trace='Y' or 'N' (
```
data structure *kc_diag_and_account_par_str*)

After activation of the BCAM trace function, the trace files (SAM file type) assigned through
the SET-FILE-LINK commands are opened with EXTEND.
If the BCAM trace is activated with the KDCDIAG command without assigning trace files
with SET-FILE-LINK commands, then the command will be executed successfully, but the
application task will encounter a DMS error while writing the trace information, and the
BCAM trace is then deactivated as a result.

After deactivation of the BCAM trace function, the trace files are closed and can subse-
quently be evaluated. Since closure of the files must be performed by all the application's
tasks, it may take a little while until all trace files have been closed. The SHOW-FILE-
ATTRIBUTES command can be used to see which files have been opened.
If a DMS error occurs for any of the trace files, then openUTM terminates the trace function.

Evaluating the BCAM trace

The trace files can be evaluated after deactivation of the BCAM trace function or after termi-
nation of the UTM application. The tool KDCBTRC is provided for this purpose. The tool
KDCBTRC can only be used to evaluate trace files of the same UTM version.

Before the KDCBTRC tool is called, the trace files for the individual tasks must be merged
into a single file.

There is a sample procedure named BTRACE for merging the individual trace files and for
evaluating the merged file with the KDCBTRC tool. The SDF-P BTRACE procedure is
shipped in the SYSPRC.UTM.063 library.

As a general rule, KDCBTRC evaluates the file having the link name KDCBTRC. The
results are written to the file having the link name LISTFILE. The control statement is read
from SYSDTA.

The control statement has the following format
[argument-1, ..., argument-n]

and permits entry of optional arguments to control editing. The arguments are described in the following section. The entire trace file is edited if you enter an „empty" control statement (blank).

Arguments for the editing program

LT=ltn / (ltn1, ...,ltn5)

One to five LTERM names can be specified with this operand. Only those trace records which contain one of the specified LTERM/LPAP names are edited.

Default: All trace records are edited.

PT=ptn / (ptn1, ...,ptn5)

One to five PTERM names, CON names or MUX names can be specified with this operand. Only those trace records which contain one of the specified PTERM/MUX/CON names are edited.

Default: All trace records are edited.

BC=bcn / (bnc1, ...,bnc5)

One to five BCAMAPPL names or ACCESS-POINT names can be specified with this operand. Only those trace records which contain one of the specified BCAMAPPL/ACCESS-POINT names are edited.

Default: All trace records are edited.

PR=prn / (prn1, ...,prn5)

One to five processor names can be specified with this operand. Only those trace records that contain one of the specified PROCESSOR names are edited.

Default: All trace records are edited.

US=usr / (usr1, ...,usr5)

One to five user IDs (USER) can be specified with this operand. Only those trace records which contain the name of one of the specified user IDs are edited.

Default: All trace records are edited.

AN=Y/N   When AN=Y is specified, those trace records which contain announce-ments are edited. If AN=N is specified, then editing is suppressed.

Default: AN=N

PB=Y/N   When PB=Y is specified, those trace records which contain BCAM or RSO parameter blocks are edited. If PB=N is specified, then editing is suppressed.

Default: PB=N

CL=Y/N When CL=Y is specified, those trace records which contain connection letters are edited. If CL=N is specified, then editing is suppressed.

Default: CL=Y

IN=Y/N When IN=Y is specified, those trace records which contain input messages are specified. If IN=N is specified, then editing is suppressed.

Default: IN=Y

OUT=Y/N When OUT=Y is specified, those trace records which contain output messages are specified. If OUT=N is specified, then editing is suppressed.

Default: OUT=Y

DVC=T/P/B The DVC operand serves to determine the format of the output file.

DVC=T: Max. line length 79 characters, no printer control characters
DVC=B: Max. line length 79 characters, with printer control characters
DVC=P: Max. line length 132 characters, with printer control characters

Default: DVC=T

SOCKET=Y/N

When SOCKET=Y is specified, those trace records that contain traces for communication with the TCP/IP protocol are edited. If SOCKET=N is specified, editing is suppressed.

Default: SOCKET=Y

2.2.6.3   **KTA trace in file**

In certain special error situations the trace entries to be found in the UTM dump will not be sufficient to ascertain the cause of an error. The KTA trace is available for such situations; this writes the KTA trace entries to a file whenever the trace area overflows.

The name of this file is made up of the base name *filebase* and the TSN of the tasks in question, i.e. the following file is created for each task:

filebase`.KTATRC.tsn`

Activating/deactivating the KTA trace

When the application starts up, the function is deactivated by default. The following start parameter specifies that KTA trace entries are written to a file as of application startup:

`.UTM START TESTMODE=FILE`

The administration command:

`KDCDIAG TESTMODE = FILE`

creates and opens the UTM trace file during the application run. However, the trace entries are only written to the file when the KTA trace area is full or when this task is terminated normally.

This function is deactivated with the administration command

`KDCDIAG TESTMODE = OFF`

Thereby the content of the KTA trace area is written to the file, which is then closed.

If the function was active when the application was terminated, then remaining traces are written to a file.

Evaluating the KTA trace

The SHOW-FILE command or the utility routine DPAGE can be used to view the content of this file and print it if required.

2.2.6.4   OSS trace

If problems are experienced with OSI TP connections, the trace function of OSS is available as a diagnostic tool. OSS writes the trace information to the following files:

filebase.OSST.no.tsn

where *filebase* is the base name of the KDCFILE in the MAX statement, *no* is the sequential number of the trace (0 - 9) and *tsn* is the task sequence number of the task.
openUTM monitors the size of the OSS trace file and switches to the next file when needed. The numbers 0 through 9 are assigned cyclically, and trace file 0 is overwritten after trace file 9 is closed.

Contents of the OSS trace

The OSS trace comprises a number of different types of trace records. Logging can be activated for each type individually or for all types.

The following types of trace records exist:

SPI   The XAP-TP System Programming Interface is logged.

INT   Internal execution in the XAP-TP module is logged.

OSS   The OSS calls are logged.

SERV   The OSS-internal trace records of the type O_TR_SERV are logged.

PROT   The OSS-internal trace records of the type O_TR_PROT are logged.

Activating/deactivating the OSS trace

This trace can be activated and deactivated either by using a start parameter, i.e. on starting the application, or by means of an administration command.

● Activation/deactivation by start parameter:

```
[.UTM]    START     [,OTRACE=ON ]
                    [,OTRACE=( type1 [, type2 ], ... )
                    [,OTRACE=OFF]
```

When OTRACE=ON is set, the OSS trace is activated on starting the application and all types of trace records are logged.

With OTRACE=(type1 [, type2], ...), the OSS trace is activated for the specified types after the application is started (the types can be specified in any sequence).

With OTRACE=OFF (default) the trace is inactive on starting the application.

● Activation/deactivation by administration command:

```
KDCDIAG  OT[RACE]=ON | ( type1 [, type2 ], ... ) | OFF
```

OTRACE = ON activates the OSS trace for all types of trace records, OTRACE=(type1 [, type2], ...) activates the OSS trace for all the specified types, OTRACE=OFF deactivates the OSS trace.

At the OSS interface, the parameters are set to the following values when the trace is activated:

```
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
```

These parameters are described in detail in the OSS manual.

Evaluating the OSS trace

OSS provides the program STEP for evaluation of the OSS trace files which are produced. Use of this evaluation program is described in detail in the "OSS (BS2000/OSD)" manual along with the associated README file.

2.2.6.5 ADMI trace

The ADMI trace logs all calls of the KDCADMI program interface.

The following data is written prior to the call:

–   Content of the data area

The following data is written after the call:

–   Addresses of the parameter area, identification area, selection area and data area

–   Contents of the fields in the parameter area, e.g. operation code (opcode), object type, length of the data area, return code

–   Content of the data area

The ADMI trace can be enabled via the start parameter ADMI-TRACE and enabled or disabled via WinAdmin, WebAdmin or the administration program interface (KDCADMI).

By default, the trace is written to the following file:

KDC.TRC.ADMI.*appliname*.*hostname*.*tsn*

Where *appliname* is the name of the UTM application (MAX APPLINAME), *hostname* is the name of the computer on which the application is running and *tsn* is the number of the task. In the UTM start procedure, you can also set up a different trace file for each task and use the SET-FILE-LINK command to assign it the link name KDCADMI.

3 The UTM dump

In the case of serious errors openUTM creates a dump (UTM dump) of all the relevant data. A dump can be caused by the following:

– a program unit has issued a PEND ER call
– a KDCS return code KCRCCC $\geq$ 70Z occurred in the application program due to a serious error with a KDCS call or an error in interoperation with a database or formatting system or because an STXIT event occurred
– a diagnostic dump was requested (e.g. KDCDIAG)
– an error in the UTM system code or in other software components such as BS2000 system, FHS, internal formatting routines or VTSU aborted the application run.

In the last instance, the UTM dump contains data from all tasks of the application and in all other cases, only data for the task concerned. All UTM dumps are written in compressed form.

3.1 The files of the UTM dump

openUTM writes the dumps in a file generation group (FGG) or in a normal BS2000 file (dump file). By default, the names of the dump files or of the FGG possess the following structure:

```
DUMP.UTM.rrrrrr.ttttff.aaaaaaaa
```

If the dump was written while the application was running and the start parameter DUMP-PREFIX has been specified then the names of the dump files or of the FGG possess the following structure:

```
dump-prefix.rrrrrr.ttttff
```

The letters mean:

`dump-prefix` prefix that was specified in the start parameter DUMP-PREFIX.

`rrrrrr` ID identifying the cause of the memory dump.

`tttt` is the task sequence number of the task which caused the dump. In the event of the application being aborted, `tttt` is the TSN of the task that initiated abortion.

tttt = UTIL if the dump is generated in a UTM tool (KDCDEF or KDCUPD).

`ff` sequence number of the dumps generated by a task in an application if the dump files are not created as a file generation group (FGG); or hexadecimal value of the counter for the number of dumps a process has written since the application started if the dump files were created as FGG files.

`aaaaaaaa` name of the application for which the dump is produced.

The dump files are either normal BS2000 files or files of a file generation group (FGG).

A file generation group is created if the dump is caused by the application being aborted. The name of the FGG depends on the task that initiated abortion of the application. For the other tasks, openUTM writes the dump information to other files of the FGG. This may not be possible in some cases. openUTM then writes the dump information of follow-up tasks to individual files and the 'ff' count is maintained on a task-specific basis. Additionally, the suffix .iii is appended to this one file (iii = file generation number that the task would have normally written). If this file generation number is not available, e.g. at the start or end of a task, then .000 is appended.

You can specify the user ID to which openUTM writes a UTM dump by means of the start parameter DUMP-USERID, see the openUTM manual "Using openUTM Applications". Syntax of the start parameter:

```
.UTM START DUMP-USERID={ STANDARD | SYSUSER } )
```

When STANDARD is specified (default setting), the dump files are written under the user's own user ID (i.e. the user ID under which the UTM application is running), and under the $SYSUSER user ID when SYSUSER is specified.

Notes

– In some cases, openUTM may set tttt=0000, ff=00 and aaaaaaaa=NONAME; this means that the relevant data is not available, e.g. at the start or end of a task.

– The dumps contain areas which are of significance for the application.

– If more than one task is active for the application, one of the dumps contains the return code indicating the reason for abortion (REASON). The other dumps contain a return code indicating that these are follow-up dumps.

If the UTM application was started with TESTMODE=OFF, and if a PEND ER occurs with one of the following KCRCDC codes, the UTM dump is suppressed:

```
FH01, FH02,
FR01, FR02,
K301, K302, K345, K601, K602, K603, K608,
KM01, KM02, KM03, KM04, KM05, KM07, KM08,
KR01, KR02,
KT01, KT02, KT04,
KU14,
XT80
```

In these instances loading will also be avoided. The exception is when PGWT calls are permissible for the current TAC, and the call where the error occurred was not a PEND call.

In the case of the KCRCDC code K316, no UTM dump is written regardless of whether or not test mode is active. If PGWT calls are permitted for the current TAC, and if the call where the error occurred was not a PEND call, the program will be loaded.

3.1.1 Reducing the volume of dump information via the DUMP-CONTENT parameter

The start parameter DUMP-CONTENT allows you to specify whether openUTM is to reduce the volume of dump information or not. In this case, reduction means that task-independent KAA memory areas (common memory pools) are only included in the dump of the task which caused the application to abort. Reducing the dump information means that the diagnostic documentation in the event of abortion of an application requires far less space. Costs for data storage and file transfer costs when forwarding the diagnostic documentation are far lower. For this reason, reduction of the dump information is activated by default. The start parameter DUMP-CONTENT can be used to deactivate or reactivate reduction of the dump information as required.

Syntax of the start parameter:

`.UTM START DUMP-CONTENT={ ST`AND`ARD | EXT`ENDED `}`

STAN**D**ARD   When openUTM creates a dump file generation, task-independent memory areas are only contained in the dump for the first task (which caused abortion). This is generally sufficient for diagnostic purposes and is set by default.

EXTENDED   Task-independent memory areas are contained in all the dumps of a dump file generation. You should only set this value when required to explicitly by the system service staff.

3.2 **The KDCDUMP utility routine**

The dump files created by openUTM are PAM files. These files must be edited using the utility routine KDCDUMP. KDCDUMP offers two possible methods of editing:

● Interactive evaluation:
  KDCDUMP loads a complete dump or a part of it into memory, allowing you to process it interactively at the
  terminal. Interactive evaluation allows you, for example, to perform specific editing and search operations on particular tables in an extensive dump.

● Editing of entire dump files in list form:
  KDCDUMP produces complete, edited dump files or file generation groups (FGGs) which you can output to a printer or view on the terminal by means of /SHOW-FILE, for example.

Editing of the dumps is controlled by means of statements which are entered after starting KDCDUMP. The KDCDUMP utility can also be controlled by specifying link names.

KDCDUMP is terminated by the control statement END.

If KDCDUMP runs into an error during the evaluation of the dump, the link name KDCDUMP is not released automatically. In this case, you should release the link name manually before KDCDUMP starts up next time and you should try to prepare the dump without summary information (INFO=DUMP).

If KDCDUMP and the dump file do not belong to the same version of openUTM, then the dump is not output. KDCDUMP then outputs message K719 from which you can determine which version KDCDUMP and the UTM dump each belong to.

Please note that the dumps from all the tasks of an application are required as when you are putting together diagnostic material.

3.2.1 Starting KDCDUMP

The KDCDUMP utility can be started for the interactive analysis of a dump in the dialog or to create the dump file in the form of a list. The following sections describe how you use START-EXECUTABLE-PROGRAM to start KDCDUMP with and without a link name.

Alternatively, you can also call KDCDUMP via the SDF command START-KDCDUMP. This command is located in the SDF UTM application area. For more detailed information, see openUTM manual "Using openUTM Applications under BS2000 Systems" section "Calling UTM tools".

Starting without link names to evaluate the dump

```
/START-EXECUTABLE-PROGRAM FROM-FILE= -
/        *LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL,ELEM=KDCDUMP)[,TEST-OPT=*AID]
```

The TEST-OPTION=*AID only needs to be specified if a table is to be edited symbolically; see the KDCDUMP statement TABLE. After the utility has started, the KDCDUMP statements are read from SYSDTA.

If you wish to edit dumps interactively, the link name KDCDUMP must not be assigned to a file.

Starting with link names to create the entire dump file in the form of a list

If you are using a link name, KDCDUMP can be started as follows:

```
/SET-FILE-LINK LINK-NAME=KDCDUMP,FILE-NAME=dumpfile
/SET-FILE-LINK LINK-NAME=LISTFILE,FILE-NAME=listfile
/START-EXECUTABLE-PROGRAM FROM-FILE= -
/        *LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL,ELEM=KDCDUMP)
```

dumpfile   name of the dump file

listfile   name of the edited file

It is also possible to specify an FGG file as a dump file in the /SET-FILE-LINK command.

After starting, KDCDUMP executes the following statements:

```
FILE *KDCDUMP
END
```

The FILE statement is described on .

If no file is assigned the link name LISTFILE, KDCDUMP by default generates a list file under the following name (see page 54 for the meaning of the letters):

```
LST.UTMDMP.rrrrrr.ttttff.aaaaaaaa.
```

You can subsequently print the file created:

```
/PRINT-DOCUMENT FILE-NAME=listfile,LINE-SPACING=*BY-EBCDIC-CONTROL
```

3.2.2 KDCDUMP statements

The following table gives an overview of all the KDCDUMP statements and their meanings:

Statement	Meaning
!	Abort KDCDUMP and execute system command
!!	Execute most recently executed system command again
AFIND	Find address in UTM dump
scrolling statements	Position cursor in work area
DUMP	Read complete UTM dump or part of it into memory
EDT	Call editor
END	Terminate KDCDUMP
FIND	Find and show table entry
FILE	Edit single dump file
FGG	Edit all files of an FGG (file generation group)
HELP	Display help for KDCDUMP statements
LIST	Edit table section
REPFILE	Information about REP file
SFIND	Search for a pattern in the dump
SH	SYS	Interrupt KDCDUMP and return to operating system level
SYSLST	Activate/deactivate logging
TABLE	Show table

If an error occurs during the execution of KDCDUMP, such as an input error or an error in the program itself, then job switch 3 will be set to "ON" if the error is not an address error or when the dump is incomplete. The list of KDCDUMP messages can be found in section "KDCDUMP messages" on page 385.

Job switch 3 is set to "ON" by KDCDUMP at the beginning of the program run. If no error occurs during the program run, then the switch is set to "OFF". Switch 3 may also not be manipulated during the program run by the user. In particular, then user may not set the switch to "OFF" using the ! statement or after the SH or SYS statement.

Entering KDCDUMP statements

All KDCDUMP statements are read from SYSDTA. A statement may comprise up to 256 characters; longer inputs result in error message K759.

Where statement names can be abbreviated, this is indicated in bold type. Thus, for example **AF**IND, indicates that you can abbreviate the statement name to AF.

With certain operands, one of the three input formats "C-string", "X-string" or "decimal" must be observed:

Designation	Input format / Example
C-string	[C]'This is a C-string'
X-string (hexadecimal)	X'AAAF' or X'aaaf' or X'AaAf'
Decimal	12345

Only C-strings are case-sensitive. Case is not significant for the other input formats.

If indices or displacements/offsets are specified, then these are always interpreted as being positive.

The sections which follow describe the KDCDUMP statements in alphabetical order.

! Enter system command

This statement allows you to execute a BS2000 command during a KDCDUMP session.

```
!␣       command
```

command   BS2000 command. Note that some BS2000 commands cause the
            KDCDUMP program to terminate.

!! Repeat most recently executed system command

This statement allows you to repeat the most recent BS2000 command (executed with ! or !!) during a KDCDUMP session.

```
!!
```

AFIND Find address in dump

This statement serves to find a memory address in the UTM dump which was a valid address for a UTM table area while the dump was in progress.

The output displayed on the terminal is the associated table entry, either from the beginning of the entry or from the desired location.

If the command syntax is faulty or if no dump was loaded into memory, then a corresponding error message is output (K710 or K711 respectively). Should the specified address not represent a memory address or not even be assignable to precisely one table entry, then also an error message is output (K712 or K713 respectively).

```
AFIND␣      address
```

address   Desired address in memory. The address must be specified in hexadecimal
              form.
              Addresses in the dynamic table XAP-DYNM-AREA and the table group
              XAP-LOC-BUFF are not found.

Scrolling statements for interactive evaluation

If the required information cannot be displayed in one screen, the following statements can be used to scroll the display:

Statement	Meaning
+ ␣ Return only (blank entry)	Scroll forward one screen page.
+n	– With table elements which cannot be displayed in their entirety in one screen page, n is added to the start address of the information currently displayed. The result of this addition is the address starting at which the next information is displayed. – With tables where one element of the table can be displayed in its entirety in one screen page, n is added to the index of the first table element currently displayed. The result of this addition is the index of the element at which the continuation of the table display commences. If the end of the table is exceeded, the last table element is displayed; with trace tables, division into pages is performed automatically. n can be specified in decimal or hexadecimal form. n is ignored for the statements HELP TABLE-NAMES and REPFILE. Scrolling continues normally.

Statement	Meaning
++	The end of the table or of the table entry is displayed. The trace tables are exceptions, where scrolling is performed to the chronological end. With the statements HELP TABLE-NAMES and REPFILE a normal "+" statement is executed.
- -n --	For backward scrolling, the same applies - with the opposite sign - by analogy as with forward scrolling. Scrolling is performed in the reverse direction no further than to the beginning of the table. When displaying the REP file, backward scrolling is not possible.
<	Only for the TRCA table: "<" can be used to scroll back to the beginning of processing of the current (or previous) ANNOUNCEMENT. In the case of the SFIND command, "<" takes you to the previous hit, even when the current hit is in a TRCA table.
<<	In the case of the SFIND command, you can use "<<" to scroll to the beginning of the hit list.
>	Only for the table TRCA: ">" can be used to scroll forward to the beginning of processing of the next ANNOUNCEMENT. In the case of the SFIND command, ">" takes you to the next hit, even when the current hit is in a TRCA table.
>>	In the case of the SFIND command, you can use ">>" to scroll to the end of the hit list.
X	The table display is aborted. Alternatively, the END command may also be entered, for example.

Screen support is offered in the form of a display indicating which statements are permitted in each case, for example: "+/X" with the statement REPFILE, "+/-/X" for all tables except TRCA "+/</>/-/X" for the table TRCA

DUMP Read UTM dump into memory

You can use this command to decompress a UTM dump file (not an FGG file) and to read it into working memory either completely or area-by-area. You can specify the UTM dump file via either its file name or its link name.

You must submit this statement before you can edit a UTM dump interactively at the terminal (e.g. using the TABLE, LIST statements).

Particularly in the event of large UTM dumps, where the main memory or virtual memory available is insufficient to accommodate the entire uncompressed dump, it makes sense to read a number of individual dump areas into main memory one after another (using several DUMP commands) and to edit them.

If the UTM dump has not been written in full then completed dump areas can be read in and evaluated in turn.

You use the DOMAIN operand to specify the area of the UTM dump you wish to edit. If you then wish to edit other areas of the UTM dump, you must issue additional DUMP statements. If you specify FILE=*SAME in the place of the file name or link name, openUTM reads the new area from the UTM dump file that has already been uncompressed (suffix `.T`). This considerably accelerates the reading process.

Every time a DUMP command is issued, the area that had been read into main memory previously is removed and the new area is read in.

```
DUMP␣      { FILE = { filename | *SAME } | LINK = linkname }

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

FILE=

    filename   Name of the UTM dump file

    *SAME   Specify FILE=*SAME in place of the file name or link name if you wish to read a new area of the UTM dump which has already been uncompressed using DUMP into main memory (see DOMAIN operand).

LINK=linkname

                Link name of a UTM dump file. The link name may consist of alphanumeric characters only.

DOMAIN=		This allows you to specify which part of the UTM dump is to be read into the work memory.
	ALL	The whole UTM dump is read into memory.
		This is the default.
	KAA	**K**DC **A**pplication **A**rea (application-global system memories) is read into memory.
	KTA	**K**DC **T**ask **A**rea (task-specific system memories) is read into memory.
	CACHE	UTM cache is read into memory.
	MPGP	Memory page pool is read into memory.
	SLOT	Slots (dynamically created tables) are read into memory.
	ROOT	KDC**ROOT** areas and DIAGAREAs are read into memory.
	XAPTP	Areas of the XAPTP module are read into memory.
	STACK	STACK area is read into memory.

USERFILE
    Read cluster user file into memory.

JOURNALFILE-1,JOURNALFILE-2
    Read cluster administration journal file(s) into memory.

BUFSEGMENTS
    Read segments of buffer management at local node level into memory.

GSSBFILE
    Read cluster GSSB file into memory.

LOCKFILE
    Read cluster lock file into memory.

CFGFILE
    Read cluster configuration file into memory.

ULSFILE
    Read cluster ULS file into memory.

EDT Call editor

This statement serves to call the editor EDT as a subroutine, i.e. you can use it to read a file into the work area of EDT.

```
EDT␣       [ filename ]
```

filename   Name of the file to be read in

END Terminate KDCDUMP

This statement serves to terminate KDCDUMP normally.

```
END
```

FGG Edit all files of an FGG

The FGG statement serves to edit all the files of an FGG jointly in a single output file. The name of the output file is specified with the OUTFILE operand. In command mode, the user is free to issue any number of FGG commands.

```
FGG␣               { fgg-name | *linkprefix }

                   [,EDIT= { PRINTER | TERMINAL }]

                   [,INFO= { LONG | DUMP | SHORT }]

                   [,OUTFILE = filename ]
```

fgg-name   Name of a file generation group (FGG) which contains the UTM dump. openUTM creates a separate FGG for each task in the application.

*linkprefix   Link name prefix of the files which originate from a dump FGG, but exist only as individual files. The prefix may consist of a maximum of three alphanumeric characters only. The actual link name has the format

```
linkprefix.number,
```

where `number` is a decimal number comprising a maximum of four digits, without leading zeros, commencing with '1', '2',... etc..

On successful execution of a command, all link names are deleted from the TFT.

EDIT=   This operand controls editing:

With EDIT=PRINTER (default), an output file is created which is intended for printing on a high-speed printer. The output contains feed control characters, page headers and a maximum line length of 132 characters. The table of contents at the end of the output relates to print pages and line numbers (for the unprinted dump). The file can be printed out by means of the command:

```
/PRINT-DOCUMENT FILE-NAME=listfile,
   LINE-SPACING=*BY-EBCDIC-CONTROL
```

With EDIT=TERMINAL, the output file is edited in such a way that it can be evaluated on screen using an editor. The maximum line length is 80 characters. The table of contents at the end of the output relates to line numbers.

INFO= This operand controls output of the summary information. This is an extract of the complete dump information and contains the data frequently required for diagnosis.

INFO=LONG: editing with summary information (default).
INFO=DUMP: editing without summary information.
INFO=SHORT: KDCDUMP outputs only summary information.

OUTFILE= This operand allows you to specify that output is to be written to a file with the name *filename*.
If you do not specify OUTFILE, KDCDUMP checks whether the link name LISTFILE has been assigned or assigns the default name
`LST.UTMDMP.rrrrrr.tttttff.aaaaaaaa` (See page 54 for a description).

Notes

- When the FGG control statement is entered, a UTM dump file which was read in by means of the DUMP command is removed in its entirety from memory. Consequently, the UTM dump under examination prior to the FGG command will no longer be available for diagnostic purposes.

- The individual file generations of the FGG are read consecutively and removed from memory again after processing, so that after execution of this command none of the files remain in memory.

- If the dump files are read in by means of link names (*linkprefix is specified), then after each successful editing operation performed on a dump file the associated link name is removed from the TFT.

Example

Let us assume we have four files of a former UTM dump FGG, each of which is assigned a link name having the prefix UD1.

```
SET-FILE-LINK LINK-NAME=UD1.1,FILE-NAME=DUMP.UTM.PCMM01.1
SET-FILE-LINK LINK-NAME=UD1.2,FILE-NAME=DUMP.UTM.WAITT2.2
SET-FILE-LINK LINK-NAME=UD1.3,FILE-NAME=DUMP.UTM.WAITT2.3
SET-FILE-LINK LINK-NAME=UD1.4,FILE-NAME=DUMP.UTM.WAITT2.4
```

These four files are edited by means of a single KDCDUMP command:

```
FGG *UD1
```

FILE Edit single dump file

This control statement edits a single dump file or a single generation of a UTM dump FGG.
The result of editing is written to an output file. The user can define the name of the output
file by means of the OUTFILE operand or a SET-FILE-LINK command with the link name
LISTFILE. If no file is specified, KDCDUMP creates a list file with the default name.

```
FILE␣            [{ dumpfile | *linkname }]

                 [,EDIT= { PRINTER | TERMINAL }]

                 [,INFO= { LONG | DUMP | SHORT }]

                 [,OUTFILE = filename ]
```

dumpfile   Name of the UTM dump file. This file may also belong to an FGG.
              If the positional operand is omitted, it is assumed that a UTM dump file has
              already been read in with the DUMP statement. The FILE statement is then
              applied to this file. In this instance, the UTM dump file is not removed from
              memory.

*linkname   Link name of the UTM dump file. The link name may consist of three alpha-
              numeric characters only.
              On successful execution of the command, the link name is deleted from the
              TFT.

EDIT=   This operand controls editing:

              With EDIT=PRINTER (default), an output file is created which is intended
              for printing on a high-speed printer. The output contains feed control
              characters, page headers and a maximum line length of 132 characters.
              The table of contents at the end of the output relates to print pages and line
              numbers (for the unprinted dump). The file can be printed out by means of
              the command:

              ```
              /PRINT-DOCUMENT FILE-NAME=listfile,
                 LINE-SPACING=*BY-EBCDIC-CONTROL
              ```

              With EDIT=TERMINAL, the output file is edited in such a way that it can be
              evaluated on screen using an editor. The maximum line length is 80
              characters. The table of contents at the end of the output relates to line
              numbers.

INFO=   This operand controls output of the summary information. This is an extract
              of the complete dump information and contains the data frequently required
              for diagnosis.

              INFO=LONG: editing with summary information (default).
              INFO=DUMP: editing without summary information.
              INFO=SHORT: KDCDUMP outputs only summary information.

OUTFILE=   This operand allows you to specify that output is to be written to a file with the name *filename*.

If you do not specify OUTFILE, KDCDUMP checks whether the link name LISTFILE has been assigned or assigns the default name `LST.UTMDMP.rrrrrr.ttttff.aaaaaaaa` (see for a description).

Notes

- If a link name KDCDUMP exists when KDCDUMP is started, KDCDUMP assumes that the associated file is a UTM dump. This is then edited. KDCDUMP subsequently terminates itself automatically.

- It should be noted that a UTM dump file which was read in by means of the DUMP command is deleted in its entirety from memory when the FILE control statement is used with the positional operand.

- In lists, hyphens in the table names (e.g.. CPTRT-NSR) are replaced by underscores (e.g. CPTRT_NSR).

FIND Find and show table entry

FIND serves to display the table entry for a UTM resource. The type of the display is the same as for the TABLE command.
FIND can only be executed if the current dump contains a KAA portion. Dumps with REASON PENDER or FMTERR, for example, do not contain a KAA portion.

```
FIND␣          { *ANY | type }

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

*ANY   All UTM types specified for *type* in the KDCDEF statement are searched.

type   Type of UTM resource. The following types can be specified:

type	Meaning	type	Meaning
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

If the resource to be sought is a secondary storage area, the corresponding entry from the table DICN-NSR or DICN-SR is displayed. Otherwise, the entry in either the relevant NSR table or an SR table is output.

*ANY   All KDCDEF names are output, which were found for the specified *type*.

kdcdef-name Name from the KDCDEF generation. This name must be entered as a C-string.

If KDCDUMP is unable to unambiguously identify the resource because, for example, "kdcdef-name" has not been specified, KDCDUMP then offers individual suggestions for selection. These must be acknowledged with "+" or ">":

+   displays the associated entry. Afterwards, further scrolling is possible with ">".

>   continues the search

A new command can also be entered instead of "+" or ">".

HELP Help on KDCDUMP

The HELP command provides information about the operation of the KDCDUMP program. This information is output to SYSOUT.

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

HELP without operands simply outputs a list of all KDCDUMP statements. The operands have the following meaning:

ALL   outputs an overview of the KDCDUMP statements. Default value.

command-name

Name of a KDCDUMP statement about which brief information is output.

TABLE-NAMES

outputs all valid table names as well as additional brief information as to whether the table in question is capable of being edited symbolically.
If the table can be edited symbolically, a "D" is displayed, (if not, an "N" is displayed). The dynamic table XAP-DYNM-AREA and the table group XAP-LOC-BUFF cannot be displayed with the TABLE command.

Table groups are indicated by an asterisk (*) prefixed before the name. The individual tables of the table group are listed according to the name of the table group and indicated by a plus sign (+) prefixed before the name.

If a dump is in memory, then the number of table entries is output (in hexadecimal form). This does not apply to the SLOT tables: In this case, the highest table entry index is output - the actual number of table elements is usually smaller. As a consequence of negative results obtained during address validation when creating a dump it may happen that UTM areas or tables are not contained in the dump. Since no table entries are present in such situations, the table are flagged as follows:

UA_ERROR: It was not possible to dump the UTM area containing the table.
TA_ERROR: The table could not be dumped.

REPFILE is similarly listed as a table. The REP file, however, must be output via the REPFILE statement (not via TABLE). REPFILE is followed by the number of different problem message numbers.

LIST Edit table section

This control statement can be used to have a table section written to a SAM file or to SYSLST.

LIST␣	{ table-name | **R**EP**F**ILE } , { listfile | ***SYSL**ST }
	[, **S**TART-INDEX = { <u>F</u>IRST | start }]
	[, **E**ND-INDEX = { <u>L</u>AST | end }]
	[, **E**DIT = { <u>TERMINAL</u> | **P**RINTER }]

table-name	Name of the table from which a part is to be output. The valid names can be ascertained by means of HELP TABLE-NAMES. You can also specify a table group name as the table name (these are indicated by an asterisk in the output from HELP TABLE-NAMES). In this case, all the tables belonging to the group are output (indicated by a prefixed plus sign (+)).
REPFILE	The REP file is to be output. The specification of any further operands then has no effect.
listfile	The table section is output to the file "listfile". An existing file will be overwritten without any warning.
*SYSLST	The table section is written to SYSLST. *listfile* and SYSLST can be printed out by means of the following command: `/PRINT-DOCUMENT <file>, LINE-SPACING=*BY-EBCDIC-CONTROL`
START-INDEX=	Entry in the table at which the output is to commence:
FIRST	First entry in the table, default value.
start	Table entry index at which the output is to commence. This value can be entered in either decimal or hexadecimal form.
END-INDEX =	Last table entry which is to be output:
LAST	Output continues to the end of the table, default value.
end	Table entry index at which the output is to be terminated. This value can be entered in either decimal or hexadecimal form.
EDIT=	Editing for printer or terminal; see description of the FGG statement on page 67.

Notes

● If two or more table areas are to be written into a single file, the operand value *SYSLST is available for the second positional operand as the BS2000 command

```
ASSIGN-SYSLST TO-FILE=< output-file >, OUTPUT=EXTEND
```

causes the table areas to be written consecutively to an output file.

● If you wish to have output a section of an existing SLOT table or of the RSBF table which is contained in the UTM dump and that section is not present, no message is output.

Example

You enter the following statement:

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

If the table SLOT-SCB-STD exists, but neither entry 2 nor entry 3 does, then the file V.LIST will be written containing only the header.

REPFILE Information about REP file

This statement is used to output information about the REP file read in by the UTM dump creator on the terminal (dumps created by a UTM-T application do not contain a REP file).

```
REPFILE␣         [ { *INFO | *ALL *ONLYPMNO | pm-no [, E ] } ]
```

*INFO	If the UTM-REP file was read in without errors, the information output comprises the name of the REP file and the number of active corrections. Otherwise, in addition to the name of the UTM-REP file that was incorporated as a REP file when the openUTM subsystem was started, the DMS return code generated on reading this file is output.
	Default value.
*ALL	The complete REP file is displayed.

*ONLYPMNO

All the problem message numbers from the UTM-REP file are output in the form ppppppppsjjj, where:

pppppppp =		Number of the problem message
s	=	Status ("-" means active, "*" means deactivated)
jjj	=	Julian calendar REP date.

pm-no	Eight-character problem message number (same as pppppppp) in the form of a C-string. KDCDUMP outputs the associated REP records from the REP file (if present) on the terminal.
E	This causes all REP records to be output, starting with that belonging to "pm-no".
	This operand must always be specified in conjunction with a problem message number, otherwise it will be ignored.

SFIND Search for a string

This statement allows you to search for a string in the UTM dump and output it on the terminal. However, you can only search for strings that are part of the user information in the dump. Information created by KCSDUMP in order to prepare the dump is not compared with the search string.
As far as the stacks are concerned, only the AUTOMATIC area is searched.
In addition, the strings are only searched for within a table entry. In other words, if the string you are searching for begins in entry 1 in the USRT-NSR table and ends in entry 2, for example, this does not count as a hit.

In some tables (MPV, PROGRAM-TABL, ...), the information found is output in an edited form. In other words, if the table contains the string you are searching for, the table is displayed in a specially prepared way. Consequently, the string that has been found cannot always be seen immediately.

```
SFIND␣      search
            [, ALIGN = {1 | 2 |4 | 8}]

            [, HITS = {ALL | nr_max_hits}]

            [, DOMAIN = {ALL | KAA | KTA | AUTO | SLOT | ROOT |
                        CACHE | MPGP | XAPTP-LOCAL | XAPTP-GLOBAL |
                        USERFILE | JOURNALFILE-1 | JOURNALFILE-2 } ]
                        BUFSEGMENTS | GSSBFILE | LOCKFILE |
                        CFGFILE | ULSFILE } ]
```

search	The positional operand stands for the string you are searching for, which cannot be longer than 190 bytes. It can be specified as a C-string or an X-string.
ALIGN=	You can use this to specify the alignment limit of the search string.
1	Single-byte alignment
	This is the default.
2	2-byte alignment
4	4-byte alignment
8	8-byte alignment
HITS=	You can use HITS to specify the number of hits after which the search is terminated.
ALL	The entire dump is searched.
	This is the default.

nr_max_hits

>
> The search is terminated after *nr_max_hits*. *nr_max_hits* can be any value from 0 to 32767. 0 has the same effect as ALL.

DOMAIN   This allows you to specify which parts of the UTM dump are to be searched.

ALL   The whole dump is searched.

>
> This is the default.

KAA   The **K**DC **A**pplication **A**rea is searched (global application system memories).

KTA   The **K**DC **T**ask **A**rea is searched (task-specific system memories).

AUTO   **AUTO**MATIC STORAGES are searched (working memory areas of the UTM modules interrupted in the state TPR).

SLOT   Slots are searched (dynamically created tables).

ROOT   KDC**ROOT** areas and DIAGAREAs are searched.

CACHE   CACHE memory areas are searched.

MPGP   The memory page pool is searched.

XAPTP-LOCAL

>
> Local processor areas of the XAPTP module are searched.

XAPTP-GLOBAL

>
> Global areas of the XAPTP module are searched.

USERFILE

>
> Search through cluster user file

JOURNALFILE-1,JOURNALFILE-2

>
> Search through cluster administration journal file(s).

BUFSEGMENTS

>
> Search through segments of buffer management at local node level.

GSSBFILE

>
> Search through the cluster GSSB file.

LOCKFILE

>
> Search through the cluster lock file.

CFGFILE

>
> Search through the cluster configuration file.

ULSFILE

>
> Search through the cluster ULS file.

SH and SYS Interrupt KDCDUMP

The SH or SYS statement interrupts KDCDUMP and allows you to enter BS2000 commands on operating system level.

```
{ SH␣ | SYS␣ }
```

You can return to the KDCDUMP program with the BS2000 command RESUME-PROGRAM.

Please note that some BS2000 commands terminate any loaded program, making it impossible to return to KDCDUMP.

SYSLST Activate/deactivate logging

This statement serves to output the results of the statements AFIND, FIND, HELP TABLE-NAMES, REPFILE, TABLE via SYSLST.

The maximum length of an output line is 80 characters. No messages are written to SYSLST.

```
SYSLST␣           { ON | OFF }
```

ON   The output is directed to SYSLST.

OFF   No output on SYSLST. "SYSLST OFF" applies when KDCDUMP starts.

Output to the terminal is never suppressed.

TABLE Show table

The TABLE statement serves to output part or all of a table from the currently processed
UTM dump. The output can be in dump format or in symbolic form.

```
TABLE␣           table-name

                 [, start-index

                    [, { END-INDEX = { SAME | LAST }       |

                         DISPL = displacement              |

                         SDI = { *YES | struct-par }
                               [, OUTFILE = { filename | *SYSLST} ]
                               [, DB = db-no ]
                       }
                 ]  ]
```

table-name   Name of the table to be output, with a maximum length of 13 characters.

              You can use the HELP TABLE-NAMES command to display a list of all the
              table names.

              The dynamic table XAP-DYNM-AREA and the table group XAP-LOC-BUF
              cannot be displayed with the TABLE statement. They can only be displayed
              with the LIST statement.

              The table names can be abbreviated as long as they remain unique. The
              following rules apply:

              –   There must be at least one character at the beginning and after a
                   hyphen. This character must be the first character of the name.

              –   The individual name parts which begin with a hyphen and are delimited
                   by the next hyphen or by the end can be omitted if the following name
                   part - where one was present - is also removed.

              Example

              The table name SLOT-VGT-DYN can be abbreviated as
              SL-V or S-VGT or S-V-D etc.

start-index This positional operand specifies the number of the entry at which the output is to commence.

The default value is 0.

If the operand is given the value 0, the entire table is output and all other operands have no effect. The SLOT tables and the STACK area are exceptions here:

– The second positional operand is *not* optional for SLOT tables, but must be assignable to a valid SLOT.

– For the STACK area, in the case of operand value 0 (default value), all interrupt addresses (absolute and relative - module name + internal module address) are displayed. If the operand value n is greater than 0, then the stack entry belonging to the interrupt address n is displayed in its entirety.

With both these exceptions, following operands have no validity and, if specified, have no effect.

The input can be in either hexadecimal or decimal form.

Notes

– In the case of the trace tables, with a null specification the current entry is displayed as the last item on the screen page.

– If, in the case of the tables PTRM-NSR, LTRM-NSR, USRT-NSR, TACT-NSR, POOL-NSR, EDIT-NSR and LTAC-NSR, a particular entry is to be displayed, then the generation name is also included in the output. For the first three of these tables, in the event of a connection existing at the time of the dump, cross-references to the other tables are given.

– For bit tables and tables whose entries are always two bytes in length, it is generally not possible to position precisely to the desired entry. But the entry is then contained in the first line output with table information.

END-INDEX= This operand can be used to specify whether, apart from the entry specified under "start-index", any other table entries are to be displayed.

SAME Only the entry specified with "start-index" is displayed.

LAST The entire table is displayed starting with the entry "start-index".

DISPL=displacement

> This facility can be utilized in order to output the table entry specified in the second positional operand with a displacement from the beginning of the table entry. This capability is supported only for tables in normal dump format, i.e. in hexadecimal representation.
>
> The input format is decimal or hexadecimal (see example).

SDI =   The operand SDI (**S**ymbolic **D**ump **I**nformation) serves to output the table entries which have been edited using AID.

> *Not all* tables can be edited symbolically. Those tables which can be edited symbolically are flagged by a "D" following the table name in the output produced by the command HELP TABLE-NAMES. If symbolic editing is not possible, this is flagged by means of an "N".
>
> This facility requires that AID Version 2.0 or higher be available.
>
> Moreover, the KDCDUMP utility must have been started with the operand TEST-OPTION=AID.

   *YES   The table element is output with symbolic editing.

   struct-par   Fully-qualified name of a structure element without a structure name (see examples) which is to be edited symbolically.

> The maximum length of "structure-parameter" is 121 characters. The name must be specified as a C-string.
>
> No structure elements may be specified for ILBF and OLBF since here all headers may only be dumped as a whole.

OUTFILE=   When symbolic editing is employed, this operand can be used to direct the output to a file.

   filename   Name of the file to which the edited information is to be written. In this case, the AID link name F0 of the AID command %OUTFILE is destroyed.

> If the specified file already exists, it will be extended by the output.

   *SYSLST   The output is directed to SYSLST.

DB=db-no   This operand is evaluated only in the case of the ROOT tables TAM and TSKM. For a UTM application in which two or more databases are generated it displays the relevant area of the tables listed above. The value 0 represents the general area.
Default: 0
Minimum value: 0
Maximum value: 2

The default value applies for the optional operands END-INDEX, DISPL and SDI:
END-INDEX = SAME.

Notes

- If the selected table information does not fit in one screen page, scrolling statements (see relevant section) can be used to make further data visible (this does not apply to symbolic editing).

- If a substructure element is incorrectly specified for symbolic editing, AID will issue an error message. In this case, message K746 is not output and job switch 3 is not set.

Examples

1. `T KAA,1,SDI='TABDESC_AREA'`

   This command can be used to output the substructure KAA.TABDESC_AREA with the symbolic debugging information.

2. `T KAA,1,D=100`

   Outputs the KAA structure in hexadecimal and print-edited form starting from offset 100.

3. `T U-N,5,S='VTV_STATE.RTA'`

   The symbolic information about the CLASS variable USRT_NSR.VTV_STATE.RTA in the fifth table entry is output.

4. `T ROOT,1,S=*YES`

   The structure ROOTDATA is output in fully symbolically edited form.

3.2.3 Messages of KDCDUMP

KDCDUMP issues messages in the format K7nn. You will find the message texts and additional information on the messages later on in this document.

3.3 Contents of the UTM dump

The following describes the structure of a UTM dump. A UTM dump edited with KDCDUMP can contain the following information:

– The global application system storage (KAA, XAP-TP, cache, slots)

– The task-specific system storage (KTA, XAP-TP)

– The work pools of the UTM modules interrupted in status TPR (AUTOMATIC STORAGES)

– The KDCROOT area with the KDCROOT tables and the DIAGAREAs

– Areas for UTM cluster applications (user file, journal files, ...).

– A summary (SUMMARY)

– A table of contents.

If the dump is created due to a VTSU error message, then only the data from KTA and KDCROOT are written to the dump, but the application is not aborted.

If the dump is created due to a PEND ER (either it was programmed in the application by the user or it was created internally after KCRCCC $\geq$ 70Z), then the dump only contains the KDCROOT area.

To make finding the table easier, the prepared dump contains a list of tables with page numbers at the end of the dump. Each of these prepared tables has a header that identifies the prepared table. The headers may differ for dumps prepared for the terminal or for printing.
The header starts with the corresponding storage area dumped (KAA TABLE / KTA TABLE or KDCROOT) and the name of the table. Only the names of the prepared tables are listed in the following.
Example: (KAA TABLE) TRAN_SYN_NSR / Transfer Syntax NSR.

The part of the name after the slash has been omitted in the following if only the hyphens have been replaced by underscores or spaces in that part of the name.

3.3.1 Global application system storage (KAA)

The KAA storage area (KDCS application control area) is application-specific, i.e.
openUTM creates a KAA system storage area for every UTM application running.
This storage area is only contained in the dump of the first task of the application (cause of
the dump) by default. See the DUMP-CONTENT start parameter.
It consists of the following areas and tables (the suffix NSR stands for "not security-related",
SR for "security-related" and DSR for "dynamically security-related" in the following):

KAA	An area with a fixed structure. It mainly contains the parameters for the generation statement MAX and other non-security related parameters that openUTM requires to control the application run.
MPV	Memory page vector that contains the KAA-relative addresses for all UTM pages of the KAA.
PGPT-NSR	Page pool bit list.
PGBT-NSR	Page pool block table (bit list).
PGCT-NSR	Page pool block page counter table.
CATRT-NSR	Cache translation table.
CPTRT-NSR	Cache translation table for cluster page pool (NSR).
HASH-NSR	Hash table for 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	Task table.
PTRM-NSR	Physical terminal table.
PTRM-DYN-NSR	Pool-relative addresses for dynamic physical terminal tables.
DCB-DYN-NSR	Pool-relative addresses for dynamic UTM-D control blocks.
LTRM-NSR	Logical terminal table.
TACT-NSR	Transaction code table.
USRT-NSR	User table.
QUEU-NSR	Table for temporary queues.
VGT-DYN-NSR	Pool-relative addresses for dynamic service tables.
POOL-NSR	Terminal pool table.
FPMM-NSR	Table for the administration of asynchronous jobs.

TASEQ-NSR	Table for the order of transactions.
EDIT-NSR	Table with edit profiles.
SCBSTD-NSR	Pool-relative addresses of SCB's (control block for VTSU) for standard terminals.
SCB9763-NSR	Pool-relative addresses of SCB's (control block for VTSU) for model 9763 terminals.
SCB3270-NSR	Pool-relative addresses of SCB's (control block for VTSU) for IBM model 3270 terminals.
LOG-CB-NSR	Logging-Exit Control-Block
CARD-NSR	Internal administration table (NSR).
CDCONT-NSR	Pool-relative addresses of ID card information read in.
BCAT-NSR	BCAM application table.
ACPNT-NSR	OSI TP access point table.
CON-NSR-EXT	OSI TP extensions for physical terminal tables.
PRGT-NSR	Program table.
LMOD-NSR	Load module table.
LTAC-NSR	Local TAC table for UTM-D.
KSET-NSR	Table with key sets.
LPAP-NSR-EXT	Logical partner application table extension.
CACT-NSR	Cache control table.
CONS-ENTRIES	Consistency entries.
CL-CONS-ENTR	Consistency entries of the UTM cluster files (NSR)
SLOTMP-NSR	Slot memory pool control table.
MMOD-NSR	Message module table.
LIBRARY-TABLE	Table of libraries for KDCROOT (NSR).
AREA-TABLE	Area table for KDCROOT (NSR).
DB-NSR	Table with generation information for database systems
MPOOL-TABLE	Memory pool table for KDCROOT (NSR).
EXIT-TABLE	Table of exits for KDCROOT (NSR).
CONTEXT-TABLE	Context table for KDCROOT (NSR).

NODE-NSR	Node table for OSI TP.
INST-NSR	Table of entities for OSI TP.
TRAN-SYN-NSR	Transfer syntax table for OSI TP.
AB-SYN-NSR	Abstract syntax table for OSI TP.
APL-CNTX-NSR	Application context table for OSI TP.
PRCP-NSR	Internal administration table for principal entries (NSR)
CLND-NSR	Address information for the nodes of a cluster
SLOG-NSR	SYSLOG buffer area (NSR).
CLFB-NSR	Data of the cluster configuration file at the time of the last read access
KAA-SR	Area for global runtime parameters.
KAA-SR-MAP	Bit list for valid KAA pages of the KDCFILE.
HASH-SR	Hash table for DICN_SR.
HASHQU-SR	Hash table for QUEU_SR.
DICN-SR	Dictionary of names.
LTRM-SR	Logical terminal table.
TACT-SR	Table of transaction codes.
USRT-SR	User table.
QUEU-SR	Table for temporary queues.
VGT-SR	Service table.
POOL-SR	Terminal pool table.
FPMM-SR	Table for the administration of asynchronous jobs.
RVGT-SR	Remote service table.
CHNT-SR	Chain table for the administration of the pagepool of the KDCFILE.
NODE-SR	Node table for OSI TP.
KAA-DSR	Area for global runtime parameters.
KAA-DSR-MAP	Bit list for valid KAA pages of the KDCFILE.
TSKT-DSR	Task table.
HASHPT-DSR	Hash table for PTRM_DSR.
HASHLT-DSR	Hash table for LTRM_DSR.

HASHTC-DSR	Hash table for TACT_DSR.
HASHUS-DSR	Hash table for USRT_DSR.
HASHPR-DSR	Hash table for PRGT_DSR.
HASHKS-DSR	Hash table for KSET_DSR.
HASHLC-DSR	Hash table for LTAC_DSR.
PTERM-DSR	PTERM table.
LTERM-DSR	LTERM table.
TACT-DSR	TAC table.
USRT-DSR	USER table.
PRGT-DSR	Program table.
LMOD-DSR	Load module table.
CARD-DSR	Table with comparison strings defined in KDCDEF with USER...,CARD=(pos, string) (DSR).
KSET-DSR	Key set table.
LTAC-DSR	Local TAC table for UTM-D.
PRCP-DSR	Table for Principal entries (DSR).
RSAKEY-DSR	Table for RSA keys (DSR).
CPCH-DSR	Chain table for cluster page pool (DSR)

3.3.1.1 The CONS_ENTRIES table

The CONS_ENTRIES table provides space for 10 entries and is written cyclically, i.e. the 11th entry overwrites the 1st entry.

Entries are written in the table when the following events occur:

– when a KDCDEF generation is executed
– when KDCUPD is called
– each time the UTM application is started

A table entry has the following format:

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

type   'DEF ' for KDCDEF run
       'UPD ' for KDCUPD run
       'STRT' for application start.

r1 and r2 designate the correction status of KDCDEF, KDCUPD and UTM system code (e.g. '10' for V6.3A10).

r1,r2 'nm' two digits for the source correction character
      ' ' space for the first release of a version.

f1, f2 Filler.

For $type$ 'STRT' and 'DEF': i1,i2 and i3 contain additional information.
For $type$ = STRT

i1   'C' for UTM cold start
        'W' for UTM warm start

i2   'P' for production

i3   'B' for batch task
        'D' for dialog task

For $type$ = DEF

i2   '-' when the KDCFILE was created successfully
        'W' when the KDCFILE was created with errors.

yymmdd
        Date (year, month, day).

hhmmss
        Time (hours, minutes, seconds).

tttt   TSN of the UTM task for $type$ = STRT.

3.3.1.2 **CACHE buffer**

CACHE-BUFFER   Cache memory

3.3.1.3 **UTM SLOT POOLS**

Dynamically created tables (slots):

SLOT-SCB-STD	SCBs for connected DSS standard
SLOT-SCB-9763	SCBs for connected DSS type 9763
SLOT-SCB-3270	SCBs for connected DSS type 3270
SLOT-CARD	Contents of read ID cards
SLOT-PTRM-DYN	Dynamic, physical terminal tables
SLOT-DCB-DYN	Dynamic UTM-D control tables
SLOT-VGT-DYN	Dynamic service tables
SLOT-LOG-CB	Dynamic control table for logging exit

3.3.2 **Global application system memory of XAP-TP**

The storage area of XAP-TP contains global application tables and areas that XAP-TP sets up to control the application run. This area exists only if the UTM application is generated with OSI TP communication. It contains only information that is not relevant for logging.

By default, the storage area is contained only in the dump of the first task of the application (initiator), see .

It consists of the following areas and tables.

XAP-SHAREMEM	Fixed part of the global storage area of the XAP-TP component; includes, for example, generation values for OSI TP communication.
XAP-INST	Instance table for XAP-TP.
XAP-TRAN-SYN	Transfer syntax table for XAP-TP.
XAP-AB-SYN	Abstract syntax table for XAP-TP.
XAP-ACCPT	OSI TP access point table for XAP-TP.
XAP-PARTNER	Partner table for XAP-TP.
XAP-ASSOC	Association table for XAP-TP.

XAP-CAF	Channel auxiliary facility table for XAP-TP.
XAP-DIAL	Dialog table for XAP-TP.
XAP-LOG-DAM	Log damage record table for XAP-TP.
XAP-RCH-GROUP	Recovery context handle group table for XAP-TP.
XAP-DYNM-AREA	Dynamic global area for XAP-TP; the size of this area is defined by the OSI-SCRATCH-AREA parameter of the MAX statement.

3.3.3   The task-specific system memory (KTA)

The system memory KTA (KDCS task-specific area) is displayed for each task of the application. This memory consists of the following areas and tables:

KTA   Area with task-specific runtime parameters and working areas.

TACB   Transaction control block.

PIBF   Processing item buffer.

FILT   File table.

BCAT   BCAM application IDs.

ADMP   Addresses of memory pool.

ILBF   Input letter buffer.

OLBF   Output letter buffer.

OSBF   OSI TP MSG buffer.

PPLT   PAM parameter list 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   Message buffer.

MUXI   Input buffer for MUX protocol header.

MUXO   Output buffer for MUX protocol header.

LTERM-BUNDLE   Master LTERM bundle table.

LPAP-BUNDLE   Master LPAP bundle table.

FMGT   FMGT file management table

LCBT   Lock Control Block for CLUSTER (global locks)

ETPNDS-TPR   List of the components linked into the UTM subsystem with their ETPNDs.

TRCA   Trace area.

USBF   In UTM cluster applications: user file buffer

JFBF   In UTM cluster applications: journal file buffer

CRBF   Confirmatory record buffer.

REST   Restart control area (only for dumps during the warm start).

RSBF   Restart buffer area (only for dumps during the warm start).

3.3.4 Task-specific system memory of XAP-TP

The storage area exists only if the UTM application is generated with OSI TP communication.

It consists of the following areas and tables.

XAP-LOCALMEM   Area with task-specific runtime parameters and working areas.

XAP-TRACE   Trace area of XAP-TP.

XAP-LOC-ACCPNT   Task-specific table for OSI TP access points.

XAP-LOC-BUFF   Dynamic local buffer containing a group of tables (XAPL-CONO, XAPL-CTRE, XAPL-CTAC, XAPL-SCRA, XAPL-OSSB, XAPL-UDAT). These areas cannot be edited interactively with the TABLE command.

3.3.5 Stack area (automatic storage)

The stack area of all UTM modules that were active at the time of the dump are printed.

The point of interruption is also output as a virtual address and as a module name plus displacement.
This makes the calling sequence of the UTM system modules and the corresponding local storage areas apparent.

3.3.6 The KDCROOT area

Tables and areas are printed from KDCROOT. In many cases, these can be used to diagnose user errors. In the case of PEND ER dumps, this is the only data that is available.

More precisely, the prepared dump contains the following individual KDCROOT areas:

CONTEXT-AREA / Context Area
> The current register set at the time of the STXIT is output here with PC=Program-Counter, IW=Interrupt-Weight and the register contents of registers R0 to R15, ILC , PM and CC.

PROGRAM-TABL / Program Table
> Program table. An entry contains, among other things, the program name and the start address of a program unit.

LOAD-MODULE / Load Module Table
> Area with information on load modules.

LOAD-CONTEXT / Load Context Table
> Load context table.

AREA/Area Table
> Area containing information on the generated areas.

EXIT / Exit Table
> Area containing information on the exits (contains indices in PRGT).

LIB / Library Table
> Library table.

MEMORY-POOL / Memory Pool Table
> Area containing information on the memory pools. In the event of a generation without load modules, this area only contains information from the generation. Other information (e.g. the address of the memory pool) can be found in the User Root area.

MSG-MODULE / Message Mod Table
> Area containing information on the message modules.

UTM-DIAGAREA / UTM Diagarea
> Area with diagmostic information for all KDCS and FHS calls

KB / Communication area KB
> Communication area, consisting of a KB header, KB return information and KBPROG in the generated length.

SPAB / Standard primary working area SPAB
> Standard primary working area.

MPUT-BUFFER / MPUT Buffer
              Intermediate storage for MPUT messages.

FORMUSER-BUF / FORMUSER Buffer
              Buffer area with logical I/O messages.

RESTART-BUFF / RESTART Buffer
              Restart area for screen formatting.

IO-BUFFER / IO Buffer
              Buffer area with physical I/O messages.

ROOTDATA / ROOTDATA
              Communication areas between KDCROOT and the UTM system modules.

NROOT-TRACE / Root Trace
              Area containing trace records on root execution.

FORMUSER-AR
              Output of the IUTMFORM interface parameters

HLL-USER-ARE / HLL User Area
              Parameter list of IUTMHLL.

VGM-AREA / VGM Area
              Area for the service memory of a connected database.

USER-ROOT / Root gen by user
              Area containing data on the user's root module.

XA-AREA / XA Area
              Database area

TCB-ENTRY / TCB-ENTRY (COB1)
              Area for TCB entries, if generated.

TABDESC-AREA / UTABDESC-AREA / Table Descriptors
              Area containing data on the edited root tables (contains name, address,
              number of entries and length).

ADMI-DIAGAREA / Administration DIAGAREA
              Area containing trace records for all calls to the administration program
              interface from the program units.

ADMI-USERAREA / Administration USERAREA
              Area containing a trace record for the data passed from the program unit
              through the administration program interface.

LOGEXITBUFFER/LOG-EXIT-MESSAGE-BUFFER
              Area for logging exit

ETPNDS-TU   Part list of the components linked in KDCROOT with their ETPNDs.

STRT-PAR   Area containing the start parameters that were specified on application start.

ACCOUNTING-A / Accounting Area
              Area for accounting data.

TAM / TAM
              Transaction storage for a connected database.

TSKM / TSKM
              Task-specific storage for database communication.

DB-DIAGAREA / DB Diagarea
              Area containing diagnostic information for all database calls (only if a database has been generated)

DB-USER-AREA / DB User Area
              IUTMDB parameter list (only if DB generated)

DB-INF-PROG / Db Info Program
              Area containing data on the current program unit.

DB-INF-APPL / Db Info Application
              Area containing data on the application.

DB-SUMMARY
              Area containing general information on the generated databases (e.g. number etc.)

DB-TABLE   Table of generated databases

In addition, the contents of the REP file $*userid*.SYSREP.UTM.063 are output provided the file existed and was accessible during the application run.

3.3.6.1 PROGRAM table

The entries have the following structure and meaning:

Entry	Bytes (hex.)	Meaning
1	-	Index of the entry in the ROOT program table
2	24	Program name
3	49	Language type of program as specified in statement PROGRAM... COMP= . Possible values[1]: X'00' = ASSEMB X'06' = C X'01' = COB1 X'08' = FOR1 X'02' = SPL4 X'0B'= ILCS X'04' = PLI/1 X'05' = Pascal-XT
4	48	Load mode of program as specified in statement PROGRAM .. LOAD= . If the application is generated with load modules, this output corresponds to the value specified in the LOAD-MODULE statement (LOAD-MODE parameter). Meaning of values[1]: STATIC = X'00': Program linked statically to application STARTUP = X'01': Program is dynamically loaded on starting application ONCALL = X'02': Program is loaded when first called (call through TAC) POOL = X'03': Program is loaded into a memory pool
5	4C	Exchange mode of program, dependent on load mode. Meaning of values[1]: NOTCH = X'00': Program not exchangeable LLMSI = X'01': Program in an exchangeable load module LLMGR = X'02': (not currently used) APPLI = X'03': Program in a load module which is loaded in a local pool
6	44	Hexadecimal index of the load module (LMOD) into which this program is linked as per generation.
7	4E	Hexadecimal index of the next program in the same load module
8	46	Hexadecimal index of the library from which the program or the load module was dynamically loaded
9	4D	Load status of program. Meaning of values[1]: NOT_LOADABLE = X'00': Program cannot be loaded LOADED = X'01': Program loaded LOAD_ONCALL = X'02': Program is loaded on being called RELOAD_ONCALL = X'03': (not currently used)
10	X'58'	Program address

[1] Hexadecimal values are relevant for a CDUMP

3.3.6.2   LOAD-MODULE table

The entries have the following meaning:

Bytes (hex.)	Meaning
-	Index of this entry in the ROOT load module table (LMOD)
24-43	Load module name
44	Load mode of load module, possible values: X'00' = Load module linked statically to application X'01' = Load module is dynamically loaded on starting application X'02' = Load module is loaded when first called X'03' = Load module is loaded into a memory pool
46	Exchange mode of load module, possible values: X'00' = Load module not exchangeable X'01' = Load module individually exchangeable X'02' = (not currently used) X'03' = Load module exchangeable only with entire application because loaded into local pool
47-5F	Version number of the load module to be loaded (on exchanging)
60-61	Index of the context to which this load module belongs
62-63	Index of the memory pool (MPOOL) into which this load module is loaded
64-65	Index of the library (LIB) from which this load module was loaded
6C-6D	Index of the first program of this load module
6E-85	Version number of the current (=last loaded) load module
88-89	Index of the next load module in the same context
8A-8B	Index of the first AREA in this load module
8C	Load status of load module: X'00' = Load module not loaded X'01' = Load module loaded X'02' = Load module when a program which is linked into this load module is loaded. X'03' = (not currently used)

If the new version number is the same as the old version number, this means that this load module has not been exchanged since the last KDCDEF run.

3.3.6.3   **UTM-DIAGAREA**

The UTM DIAGAREA is a task-specific trace area in which all events are logged. This area thus contains all the events that occurred immediately before a service or application aborted.

The UTM DIAGAREA is written cyclically. Two cycles are separated by a dividing line comprising '=' characters and blanks. The most recent entry is above the dividing line and the oldest entry is below the dividing line. Each entry is 136 bytes long. The total number of entries that can be accommodated by the UTM DIAGAREA depends on the generation parameter `MAX TRACEREC`.

The following types of entries are written to the UTM DIAGAREA:

● UTM records (type `KDCS`)

   UTM records are written if the following events occur:

   – a KDCS call to a program unit or
   – an internal call to the UTM system code or
   – an internal PEND ER call by openUTM (system PEND ER) due to a severe error

   In the event of a system PEND ER, the entry in bytes 22 - 57 contains an error text.

   In the case of administration interface calls (KDCS-Opcode=ADMI), additional trace information is written (ADMI-DIAGAREA).

● UTM records to identify the service (type `VGID`)

   A VGID record is written to the UTM DIAGAREA every time a program unit is started and when a PGWT call returns.

● FORM records (type `FHCL`)

   A FORM record is written on a call to a formatting system.

Header of the records in the UTM-DIAGAREA

Each record starts with a header which contains the following information:

Byte	Meaning
0 - 1	Counter for the current entry in the DIAGAREAs (UTM and DB)
2 - 5	Type identification (KDCS, VGID, FHCL, ITRC)
6 - 7	Not currently used (preset to '= =')
8 - 15	Timestamp

Header structure

The content of the records as of byte 16 is dependent on the record type.

DIAGAREA for a KDCS call from a program unit (type KDCS)

Byte	Field name and meaning		
16-19	KCOP[1]:	user operation code: INIT, MGET, MPUT, etc.	
		internal operation code: see table on page 101	
20-21	KCOM: operation modification		
22-23	KCLA: area length or queue level (in the case of QCRE) or KCLKBPRG: length of KB program area with INIT		
24-25	KCLM: message length or KCLPAB: length of standard primary work area with INIT KCWTIME: wait time in seconds in the case of DGET		
26-33	KCRN: reference name		
			MCOM call
34-41	KCMF: format name or edit profile name KCLT: LTERM name of the LTERM partner or KCUS: user ID or KCPA: name of the partner application (with APRO call)		KCPOS: destination of positive follow-up message

Structure of an entry of the UTM-DIAGAREA for a KDCS call (type KDCS)

Byte	Field name and meaning					KCNEG: destination of negative follow-up message
42-43	KCDF: screen function, see table on page 101					
	DPUT/DADM	DGET	QCRE	PADM	APRO	
44	KCMOD: mode ("A"/"R"/"_")	KCQTYP: destination type ("U"/"Q"/"T")	KCQMODE: mode ("S"/ "W"/ bin. zero)	KCACT: action (ON/ OFF/CON/DIS)	KCPI: service ID	
45-47	KCTAG: days	empty	empty			
48-49	KCSTD: hours			KCADRLT: new LTERM name of the printer		
50-51	KCMIN: minutes					KCCOMID: complex ID
52-53	KCSEC: seconds				KCOF: OSI TP functions	
54	KCQTYP: destination type ("U"/"Q"/bin. zero)					
55-57	empty			empty		
58-59	KCRFELD in KCKBC: KB return information area (see the table on page 101)					
92-95	Return address to the program unit (the address points to after the KDCS call in the program unit)					
96-99	Address of the user data area (2nd parameter of the KDCS call)					
100-103	Service index					
104-111	KCLOGTER in KCKBC: LTERM name					
112-119	KCBENID in KCKBC: name of the current user ID					

Structure of an entry of the UTM-DIAGAREA for a KDCS call (type KDCS)

[1] In the case of KCOP=INFO (bytes 16-19), the message area of the logged call is written, without incrementing the counter, in length KCPAC to the next entry in the UTM-DIAGAREA. In the case of an operation modification KCOM=CK (bytes 20-21), this information is of interest for diagnostic purposes since it logs the call to be verified.

The following entries are possible in the KCDF field (bytes 42-43; screen function):

Contents KCDF, hexadecimal	Symbolic name	Meaning
0000	-----	no screen function
0001	KCREPL	clear screen and output format
0001	KCRESTRT	screen restart after PEND RS
0002	KCERAS	erase unprotected fields
0004	KCALARM	BEL function
0008	KCREPR	local printer output
2000	KCEXTEND	extended line mode
4000	KCCARD	next input from the card reader

Opcode with internal call to openUTM

Contents KCOP	Situation when this record is written	Contents of fields following KCOP
STRT	Start of the UTM application program; begin start handling in UTM system code	- - -
WAIT	UTM task appended to UTM work bourse	- - -
CONT	Continuation in UTM system code after execution of formatting or a DB action via KDCROOT or call to INPUT exit	KCRCCC, KCRCKZ, KCRCDC; for INPUT exit: parameters, see table page 102
NOOP	Buffer area of MESSAREA must be emptied (only possible when event monitor is switched on)	- - -
ADMI	UTM administration action	internal UTM interface
SWCH	Transition between protected (T.C.B.) and unprotected part of the coding (toggle the ISRKEY)	- - -

Operation codes for an internal UTM call

Structure of KCRFELD

Byte	Field name and meaning		
58-59	KCRDF: screen function return KCRWVG: number of waiting services with DGET		
60-61	KCRLM: actual length of the message		
	INFO CK call	MGET call	SIGN ON call
62	KCRINFCC: KDCS return code of the KDCS call checked	KCVGST: service status	KCRSIGN1: primary code
63		KCTAST: transaction status	KCRSIGN2: secondary code
64		empty	
65	empty	KCRMGT: message type	
66-68	KCRCCC: KCDS return code		
69	KCRCKZ: identifier: P (productive or live operation application)		
70-73	KCRCDC: internal return code		
74-81	KCRMF: message format ID or edit profile return		

| 82-89 | KCRPI: service identification return
KCRUS: user ID in the case of SIGN ST or creator of the message with DGET
KCRQN: name assigned by openUTM to the temporary queue in the case of QCRE
NN |

DIAGAREA following a call to the INPUT exit

Following an INPUT exit call, openUTM enters the most important parameters received or set by the exit in the KDCS record (KCPAC):

Bytes	Field name and meaning
16-19	KCOP: "CONT" is entered here
20-25	empty
26-33	KCIFCH: first 8 characters of input
34-35	KCICVST: service status: "ES"/"ET"/"RS"/"EC"
36-37	KCIFKEY: value of F key: 1,...,24
38-39	KCIKKEY: value of K key: 1,...,14
40-41	KCICFINF: information about control fields: "UN"/"NO"/"ON"/"MO"
42-49	KCINTAC/KCINCMD: next TAC to be started or next user command
50-51	KCICCD: code for effect of input: "ER"/"CC"/"SC"/"ST"/"CD"
52	KCICUT: truncate TAC: "Y"/"N"
53	- - -
54-57	KCIERRCD: error info for data display terminal
58-...	The following information is not relevant for diagnostic purposes

Structure of a UTM-DIAGAREA entry following a call to the INPUT exit.

Error texts in DIAGAREA with SYSTEM PEND ER

To permit faster diagnosis in the event of a SYSTEM-PEND-ER, a printable error text is entered in the corresponding line in bytes 22-57 of UTM DIAGAREA.

The following table lists all the possible error texts together with the causes of the error, possible actions and error type (system error or user error).

Error text	Cause	Error type
APPL. PROGRAM DOES NOT EXIST	No indicator (KCRARA) for a program unit existed when program unit started (70Z with KR01)	System error
APPL. PROGRAM WITHOUT PEND	Program unit run was not terminated with PEND. (84Z)	User error
ASYNC. PROGRAM NOT FOUND	Asynchronous program no longer available (possible program exchange). PEND ER dump follows with 70Z and KR02.	User error
ERROR IN "START-TP" OF LGCON	The Language Connection Module delivered an incorrect return code when program unit was started.	System error
KB END LABEL OVERWRITTEN	The KB in the application program is greater than specified in the generation (70Z with KR04). Action: Change MAX KB parameter in the generation or adjust program unit.	User error
SPAB END LABEL OVERWRITTEN	The SPAB in the program unit is greater than specified in generation (70Z with KR05). Action: Change MAX SPAB parameter in the generation or adjust program unit.	User error
KCRCCC >= 70Z AFTER SVC UTM	After SVC UTM the return code KCRCCC in KB head was set to $\geq$ 70Z.	User/ system error
ERR IN STXIT-ROUTINE OF LGCON	During STXIT handling, an faulty return code was supplied by the Language Connection Module.	User/ system error
STXIT WITH XT.. ENTERED	A STXIT with weight .. occurred.	System error

Error texts in UTM-DIAGAREA with SYSTEM PEND ER

Error text	Cause	Error type
DB ERRORCODE = TA_CHAIN_RSET	The database connection module delivered the return code 'TA_CHAIN_RSET'. If this return code was triggered by a K210, K211, K216 with XAER_DUPID then it is possible that an abnormal end of a previous application run has left this transaction in the database and that a new KDCFILE was generated in this status. Possible action: shut down and restart the database	User error
KDCS CALL IN VORGANG EXIT	An illegal KDCS call was made in the VORGANG EXIT program	User error
VORGANG-EXIT-PROGRAM NOT LOADED	The program for VORGANG exit is not loaded.	User error
KDCS CALL IN FORMAT EXIT	An illegal KDCS call was made in the FORMAT EXIT program.	User error
DATABASE DOWN AT USER DB CALL	The database system was no longer connected at the time of the program unit call to the DB system.	User/ system error
ILLEGAL RET CODE FROM DBCON	The database connection module supplied an illegal return code.	System error
NO DB CALL ALLOWED IN SIGN-ON	An illegal DB USER CALL was made in a sign-on service.	User error
DYNAM. PROGRAM LOADING FAILED	An error occurred when loading a program. PEND ER dump with 70Z and KR09	User/ system error

Error texts in UTM-DIAGAREA with SYSTEM PEND ER

Service identification entry (type VGID)

Byte	Meaning
16	Service identifier
17	Session counter
18-19	Transaction counter in service
20-23	Service counter
24-27	Total of USED and ERROR for the current TAC
28-31	Length of Global Transaction Identifier (GTRID) for the XID
32-35	Length of the Branch Qualifier (BQUAL) for the XID
36-67	The first 32 bytes of the Global Transaction Identifier (GTRID)
68-115	The first 48 bytes of the Branch Qualifier (BQUAL)
116-117	Index for program table
118-119	Index of service exit in program table
120-127	Name of the transaction code which started the service
128-135	Name of the current transaction code

Structure of the entry for service identification

FORM-Record in the UTM-DIAGAREA

Each time a call is issued to the format handling system via the
IUTMFORM interface, a record is written to this area.

Bytes	Meaning
16	IUTMFORM operation code
17	IUTMFORM return code
18-21	IUTMFORM info return code
22-24	Status information following IUTMFORM call
25	Format type
26-27	Blanks (reserved for extensions)
28-29	Device feature prior to IUTMFORM call (see KCDF)
30-31	Printer acknowledge bytes
32-39	Terminal characteristics (see TSTAT macro)
40-41	Maximum length of (last) passed message
42-43	Actual length of (last) message supplied following call
44-47	Language ID and territory ID
48-55	Name of character set for USER or LTERM
56-63	Name of character set for format
64-71	Name of (last) transferred format
72-79	Name of (last) supplied format
80-83	Address of (last) transferred message area
84-87	Address of (last) transferred cursor

Structure of FORM-DIAGAREA

3.3.6.4   **DB-DIAGAREA**

Each time a USER CALL is issued to the DB system, a DB record is written to the DB-DIAGAREA. In the same way as UTM-DIAGAREA, this area is cyclically written with trace records.

Two cycles are separated by a dividing line comprising '=' characters. The record above the line is the most recent and the one below the line the oldest.

Bytes	Meaning
0-1	Counter for the current entries in the DIAGAREAs (UTM and DB)
2-5	"DBCL"= ID for DB record
6-7	Not used at present (preset to '= =')
8-15	Timestamp
16-19	Status of DB transaction prior to DB call [1], see table page 108
20-23	Status of DB transaction following DB call [1], see table page 108
24	DB operation code, see table page 109
25	DB secondary opcode
26	DB error code, see table page 110
27	DB type(hex): 01=UDS, 02=SESAM, 03=LEASY, 04=PRISMA, 06=CIS, 07=neutral DB system(type=DB), 08=ORACLE
28-31	DB trace information (for contents and meaning see DB system) In the case of XA, in BS2000: DB trace information of the last XA call: 28=RM number, 29-30=XA function code, 31=XA return code)
32-63	Secondary DB trace information (see DB system)
64-67	Combined status info TAM (further diagnostic information, relevant only when a multi-DB function is used)
72-73	Transaction counter within service
74	Number of UTM application run (starts at '1' after KDCDEF generation)
75	Identifier for table index, =' T '
76-77	UTM table index which resulted in the DB call
78-79	Action index in this table
80-83	Service counter (unique in the UTM application run)
84-87	Address used internally
88-91	Program unit return address. The address points to after the CALL DB system call.
92	* Eye catch for the recognition of the end of the diagnostic record

Structure of DB-DIAGAREA

[1] In this case, the status of the service memory is displayed for operation codes that refer to DB services.

If the DB system is linked to openUTM via the XA interface and a so-called "dynamic xa switch" id=s associated (e.g., in Oracle, the "oraswd" module), then the following diagnostic record is written when a shared transaction is signed on or off.

Byte	Meaning
16-23	Printable "ax_reg" on transaction sign-on Printable "ax_unreg" on transaction sign-off
24-27	Number of the RM instance
28-31	Unused
32-47	Printable version of call return code
48	Internal status of the transaction
49	Internal status of sign-on/sign-off
50 -51	Unused
52-54	Does the call occur inside an openUTM transaction (printable: "yes" or "no")
55	Unused
56	* Eye catch for the recognition of the end of the diagnostic record

Diagnostic record

Status of the DB transaction (content of bytes 0 to 3 or 4 to 7 of the DB record)

Contents (hex.)	Meaning of the DB status
04	The DB transaction was placed in temporary transaction end (PTC).
08	Updates made within a DB transaction.
10	DB transaction has been reset.
20	DB transaction was closed.
40	The program unit run issued a call CLOSE DB transaction; DB transaction is terminated from viewpoint of the program unit run.
80	DB transaction is open.

Status of the DB transaction. Combinations of these values also occur, e.g "88"

DB operation codes (byte 8 of the DB record)

Contents (hex.)	DB op. codes	Meaning
00	STPA	DB connection module is to check DB-specific start parameters. Call takes place at start of UTM application program.
04	CONC	Connection setup between UTM task and connection module of DB system. Call takes place at start of UTM application program.
08	DCON	Connection cleardown between UTM task and DB system. Call takes place in end handling of UTM application program.
10	USRC	Execute DB call of application program (user call).
14	FITA	Terminate DB transaction. UTM issues this call at the end of a joint DB/DC transaction.
18	CATA	Terminate DB transaction abnormally, i.e. the DB transaction is reset to the last synchronization point.
1C	BKTA	The call is used with PEND KP or PEND PA/PR with TASK change (due to TAC classes); the UTM task breaks off from DB/DC transaction processing.
20	COTA	The call takes place whenever a multi-step transaction is continued, i.e. following a PEND KP or PEND PA/PR with TASK change (due to TAC classes).
24	STAT	Display status of a DB transaction or delete all DB statuses. The DB system maintains status information on DB transaction as a means for coordinated restarts in openUTM and the DB system.
28	PETA	Request preliminary end of DB transaction.
2C	EDVG	End of UTM service; DB service still open.
30	BKVG	Interrupt of an open DB service.
34	COVG	Continuation of an interrupted DB service following BKVG.
38	RSVG	openUTM indicates a UTM conversation restart to the DB system if a DB service is open.
3C	CNFPTC	The DB connection module confirms the status of the precommitted transactions to the DB system via the XA interface and deletes these from the internal list.
40	STRT	The DB connection module notifies the DB system of the start of a DB transaction via the XA interface.
44	PEND	The DB connection module notifies the DB system of the temporary end of the transaction via the XA interface and initiates the first commitment phase.

DB operation codes

DB error codes

Contents (hex.)	Meaning of the DB error codes
00	Job performed.
04	The DB transaction had to be reset. openUTM then also resets the UTM transaction. With a DB call (user call), control is restored to the application program unit.
08	Causes service to abort with PEND ER when openUTM is used.
0C	DB system is not available, not activated.
10	DB system not available for another reason.
14	Job not executed. Try again later.
18	A (possibly) recoverable system error occurred. openUTM creates a task dump and tries to terminate the open DB transaction.
1C	An unrecoverable error occurred. Any further work with the DB system is pointless in this DB session.
20	DB system detected a user error, e.g. during checking of the DB start parameters.
24	DB system detected an interface error.

DB error codes

3.3.6.5 ADMI-DIAGAREA

The ADMI-DIAGAREA is a task-specific trace area in KDCROOT. Trace records are written to this area cyclically in the same way as to UTM-DIAGAREA. A record is written to this area each time the administration program interface is called.

The area is large enough for 71 records, each record being 112 bytes in length. The area is written cyclically. A boundary consisting of "=" characters separates two cycles. The entry above the boundary is the most recent and the one below it the oldest. One record is written for every administration call.

Records have the following structure:

Bytes	Meaning
0-1	Counter for the current entry in the ADMI-DIAGAREA
2-3	Counter for the corresponding entry (opcode "ADMI") in the UTM-DIAGAREA
4-5	Printable abbreviation for opcode (see table below)
6-7	Printable abbreviation for object type or subopcode1, depending on the opcode specified (see table below)
8-11	Address of `parameter area`
12-15	Address of `identification area`
16-19	Address of `selection area`
20-23	Address of `data area`
24-79	`Parameter area`
80-111	Name of the administered object in the length specified for the object type (2-32) from the `identification area` or from the `data area`

Structure of ADMI-DIAGAREA

The entry counter and the addresses of the areas are also logged before the UTM system code is called. The remaining data is logged after control has returned from the UTM system code and before branching back to the program unit. For this reason, the `parameter area` also contains return values - including the return code. The contents of the `identification area` are only logged if the area was used during the administration call. The name from the `data area` is only logged for KC_CREATE_OBJECT.

If the administration call is terminated with PEND ER because the address of the `parameter area` cannot be accessed or is not aligned on a word boundary, this log shows the address specified.

The following abbreviations are used:

Abb. Opcode	Opcode	Abb. subopcode1/ object type	Subopcode1 or object type
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
CR DL GT MD	KC_CREATE_OBJECT KC_DELETE_OBJECT KC_GET_OBJECT KC_MODIFY_OBJECT	AB AC AP BC CL CN CC CO CP DA DB DP ED GB	KC_ABSTRACT_SYNTAX KC_ACCESS_POINT KC_APPLICATION_CONTEXT KC_BCAMAPPL KC_TACCLASS KC_CLUSTER_NODE KC_CLUSTER_PAR KC_CON KC_CURR_PAR KC_DIAG_AND_ACCOUNT_PAR KC_DB_INFO KC_DYN_PAR KC_EDIT KC_GSSB

Abbreviation for opcode and subcode1 or object type

Abb. Opcode	Opcode	Abb. subopcode1/ object type	Subopcode1 or object type
(cont.) CR DL GT MD	(cont.) KC_CREATE_OBJECT KC_DELETE_OBJECT KC_GET_OBJECT KC_MODIFY_OBJECT	KS LC LM LP LS LT MC MM MP MX OA OC OL PE PO PR PT QP QU SI SF SP TA TC TI TR UP US UF U1 U2	KC_KSET KC_LTAC KC_LOAD_MODULE KC_LPAP KC_LSES KC_LTERM KC_MSG_DEST_PAR KC_MESSAGE_MODULE KC_MAX_PAR KC_MUX KC_OSI_ASSOCIATION KC_OSI_CON KC_OSI_LPAP KC_PTC KC_TPOOL 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

Abbreviation for opcode and subcode1 or object type

Abb. Opcode	Opcode	Abb. subopcode1/ object type	Subopcode1 or object type
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

Abbreviation for opcode and subcode1 or object type

3.3.6.6   **ADMI-USERAREA**

The ADMI-USERAREA is a task-specific trace area in KDCROOT.
This area is used to log the data passed from the program unit to openUTM at the program interface.

Since a large volume of data can be passed, only the data for one call is stored in the area. The area thus comprises a single record and covers 4140 bytes. The area is only written for those calls which pass data to openUTM. The contents of the `data area` or the `selection area` are logged, depending on the area used for the call. If this area is needed for diagnostic purposes, you must make sure that the call is the last call for which data is logged, otherwise the area will be overwritten by a subsequent call.

If a password for a user is passed in the `data area`, it is not logged, but is overwritten with binary zeros.

A record has the following structure:

Bytes	Meaning
0-1	Counter for the corresponding entry in ADMI-DIAGAREA
2-3	Irrelevant
4-59	`parameter area`
60-4139	Contents of the `data area` or the `selection area` in the length passed

Structure of ADMI-USERAREA

The contents of the `parameter area` and the data passed are logged in the UTM system code. The `parameter area` is logged as it is passed by the program unit, i.e. without return values. The subreturn code is set to zero.

The entry counter is logged after control has returned from the UTM system code and before branching back to the program unit.

3.3.6.7 ETPND entry

The layout of the ETPND areas is the same for UTM system code and KDCROOT.

The entry has the following structure:

Byte	Meaning
0-7	Component name
8-9	Colon and blank
10-17	Name of the component in the ETPND area
18-20	Version of the component
21	Version of the macro library
22-29	Compilation date in the form yyyymmdd
30-33	Load address of this component in the system
34-35	== means that a valid ETPND entry was found for the component name (bytes 0-7). ?? means that the component name (bytes 0-7) was found in the loaded system or in KDCROOT but the component does not have an ETPND.

3.3.6.8 **KB communication area**

The communication area consists of the KB header, the KB return area and the KB program area of the generated length.

The filed names are shown for COBOL. For details on the field names for C / C++ refer to the appendix of the openUTM manual „Programming Applications with KDCS".

Bytes	Field names and contents	
0-7	KCBENID:	UTM user ID
8-15	KCTACVG:	TAC used to start this service.
16-17	KCTAGVG:	day
18-19	KCMONVG:	month
20-21	KCJHRVG:	year
22-24	KCTJHVG:	working day
25-26	KCSTDVG:	hour
27-28	KCMINVG:	minute
29-30	KCSEKVG:	second
31	KCKNZVG:	service indicator
32-39	KCTACAL:	TAC used to address the program
40-41	KCTACAL:	hour
42-43	KCMINAL:	minute
44-45	KCSEKAL:	second
46	KCAUSWEIS:	card ID: A (ID card in reader) or blanks.
47	KCTAIND:	transaction ID: F (first) or N (follow-up transaction)
48-55	KCLOGTER:	name of the LPAP partner (sender)
56-57	KCTERMN:	type of terminal or printer, see table for PTERM statement
58-59	KCLKBPB:	maximum length of the KB program area as generated
60-61	KCHSTA:	number of stacked services from the point of view of the current service.
62	KCDSTA:	change in the number of stacked services
63	empty	
64	KCPRIND: program ID. "A" = program unit run in asynchronous service "D" = program unit run in interactive service	

Note: "Date of the start of service" spans bytes 16-24; "Time of the start of service" spans bytes 25-30; "Time of the program start" spans bytes 40-45.

KDCS communication area

Bytes	Field names and contents		
65	KCOF1: legal OSI TP functions "B" = basic functions "H" = basic functions + handshake functions "C" = basic and commit functions with chained transactions "O" = other combination blank if service was not started via OSI TP		
66	KCCP: UTM client protocol "0" = asynchronous service "1" = LU6.1 "2" = OSI TP "3" = UPIC "4" = TIAM "5" = APPLI "6" = SOCKET		
67	KCTARB: TA is marked for rollback.		
68-71	KCYEARVG: 4-digit year specification for the start of the service		
72-83	empty (FILLER)		
84-115	KCRFELD		
84-85	KCRDF: return area for screen functions KCRWVG: number of waiting services in the case of DGET		
86-87	KCRLM: actual length of the message		
	INFO CK call	MGET call	SIGN ON call
88	KCRINFCC: KDCS return code of checked KDCS call	KCVGST: service status	KCRSIGN1: primary code
89		KCTAST: transaction status	KCRSIGN2: secondary code
90		empty	
91	empty	KCRMGT: message type	
92-94	KCRCCC: KDCS return code		
95	KCRCKZ: identifies: P (production application)		
96-99	KCRCDC: internal return code		
100-107	KCRMF: format ID or edit profile ID return		
108-115	KCRPI: service ID return KCRUS: user ID in the case of SIGN ST or creator of message with DGET KCRQN: name assigned by openUTM to the temporary queue in the case of QCRE NN		
116 and up	KB program area		

KDCS communication area

3.3.7   Memory areas in UTM cluster applications

The memory areas that are global to the cluster consist of the following tables:

UF-HDR	Header of the cluster user file
UF-ENT	Entries in the cluster user file
JF-1-HDR	Header of the first file in the cluster administration journal
JF-1-ENT	Entries in the first file in the cluster administration journal
JF-2-HDR	Header of the second file in the cluster administration journal
JF-2-ENT	Entries in the second file in the cluster administration journal
BUF-SGMT	Segments of buffer management at local node level
GF-HDR	Header of the cluster GSSB file
GF-ENT	Entries in the cluster GSSB file
LF-HDR	Header of the cluster lock file
LF-DLK	Area for deadlock detection in the cluster lock file
LF-ENT	Entries in the cluster lock file
CF-HDR	Header of the cluster configuration file
CF-ENT	Entries in the cluster configuration file
UL-HDR	Header of the cluster ULS file
UL-ENT	Entries in the cluster ULS file

3.3.8   Summary

The dump is concluded with the summary information, a table of contents and a message section that contains the messages that are output while preparing the UTM dump.

The summary information is an extract of all dump information. It contains the data frequently required for diagnosis. This saves you time looking through the dump, especially at the preliminary diagnosis stage and when looking for duplicates.

The summary information contains significant generation parameters of the application, version number and correction status of the operating system and openUTM, start parameters of the application, the most important current table entries and the last records of DIAGAREA and TRACE area.

Some UTM dumps will not contain all tables, e.g. the PEND ER dump. Some data will therefore be missing in the summary of such dumps.

The INFO operand serves to control the output of summary information and of the dump.

If the dump was produced by KDCUPD, only the first page of the summary information is written.

3.4 Sample dumps

In the following you will find two examples of a UTM dump.
General procedure: Look for the KCRCCC and KCRCDC return codes in the dump and
refer to for the meanings of the return codes.

3.4.1 Sample dump FH01

1. Look up DIAGAREA --> last call: PEND ER

    The return codes are found in the preceding entry of the DIAGAREA or in the KB

2. Look up KB -->   Contents KCRCCC (bytes 92-94):   70Z
                           Contents KCRCDC (bytes 96-99):   FH01 (formatting error)

    The FHS return code (secondary return code MSRC) is found in the field KCRMF (bytes
    100-107 in KB): 080C (format not available, see "FHS" User Guide). An extract of the summary
    information is given at the end of the sample dump.

Example

Summary information

```
  KDCDEF related information

  UTM-mode: S
  Coldstart and first task: NO
  Number of asynchron tasks:   255
  Accounting generated: NO                                 UTM-D: NO
  Filebase name: STRBLS

  Maximum values generated

  KBPROG:        32767                                     FORMIO length:   32764
  SPAB:          32767                                     FORMAREA length: 32764
  MPUT length:   32764                                     Restart buffer length:       65536
  CPU-time:          0 msec

  System information

  BS2000 version: 2170
  Hardware type: XS31                                      Hostname: HOST0001

  Settings at start time

  KDCMON: OFF
  Version number of KDCD V06.3A00
  Number of tasks:   255                                   Number of asynchron tasks:   255
  Formatting system: FHS

  Database related information


  Information on REP's

  KDCRTST:
AUTM Dump Utility (V06.3A00) - BS2000 Application = STRBLS       TSN = 686Q     ITN = FFFFE52C   Dump Nr.   2              Page    18
        REASON = PENDER    CREATED  15-01-21  11:03:55   PRINTED  15-01-21  11:06:21
ATA:
  Information on exits

  Input exit:          NO                                  Format input exit:    NO
  USERFORM input exit: NO                                  Line input exit:      NO
  Start exit:          YES                                 Format exit:          NO
  Shut exit:           YES                                 SIGNON conversation: YES
  MSGTAC:              NO                                   BADTAC:               YES
  Logging exit:        NO
  Active exits: NO EXIT ACTIVE


  Current state

  Current TAC:   KDCSGNTC                                  Program name:  KDCSIGN1
  FGG number of program:      0
  Terminal type: T9750                                     Kind of terminal: DIALOG
  Programming language of current TAC: ILCS
  Actual length of KBPROG:    2                            Actual length of SPAB:     600


  Diagnostic information

  Dump reason: PENDER                                      Dump created: 11-10-24  10:15:26
  KCRCCC: 7OZ                                              KCRCDC: FH01
  FHS secondary returncode: 0000                           Formatting mode: OUTPUT FORMATTING

  Last ROOT action: CALL_WAIT                              Action index:    2
```

```
AUTM Dump Utility (V06.3A  ) - BS2000 Application = STRBLS        TSN = 686Q    ITN = FFFFE52C   Dump Nr.  2            Page    19
            REASON = PENDER    CREATED  15-01-21  11:03:55      PRINTED  15-01-21  11:06:21


  Recently written diag entries


    Last 15 entries of not trusted DIAGAREA recently written

    0003    01600110 0000    0002C6C8 C3D37E7E C031E390 DE0E5500    2C00F0F0 F0F00000 00027E7E 00004040    ..FHCL==..T.......0000....==..
            01600130 0020    00000000 00000000 00000000 40404040    40404040 40404040 40404040 40404040    ............
            01600150 0040    40404040 40404040 40404040 40404040    01051000 FFFFFFFF 7E7E7E7E 7E7E7E7E                 ....~~~~========
            01600170 0060    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================================
            01600190 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0004    01600198 0000    0003C6C8 C3D37E7E C031E390 DE12E100    0000F0F0 F0F00000 00027E7E 00004040    ..FHCL==..T.......0000....==..
            016001B8 0020    00000000 00000000 00000000 40404040    40404040 40404040 40404040 40404040    ............
            016001D8 0040    40404040 40404040 40404040 40404040    01051000 FFFFFFFF 7E7E7E7E 7E7E7E7E                 ....~~~~========
            016001F8 0060    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================================
            01600218 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0005    01600220 0000    0004C6C8 C3D37E7E C031E39F 88BE4701    0400C6C3 F0F18000 00027E7E 00004040    ..FHCL==..T.h.....FC01....==..
            01600240 0020    00000000 00000000 00000000 40404040    40404040 40404040 40404040 40404040    ............
            01600260 0040    40404040 40404040 40404040 40404040    01051000 FFFFFFFF 7E7E7E7E 7E7E7E7E                 ....~~~~========
            01600280 0060    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================================
            016002A0 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0006    016002A8 0000    0005E5C7 C9C47E7E C031E3AC B6F26100    C6000000 00000000 00000000 00000000    ..VGID==..T..2/.F...............
            016002C8 0020    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    ...............................
=           01600308 0060    00000000 00000000 00000000 00000000    00000000 00150000 E2E3C1D9 E3E4D740    .......................STARTUP
            01600328 0080    E2E3C1D9 E3E4D740                                                            STARTUP
    0007    01600330 0000    0006E5C7 C9C47E7E C031E3AF 29596101    C6000000 00000000 00000000 00000000    ..VGID==..T.../.F...............
            01600350 0020    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    ...............................
=           01600390 0060    00000000 00000000 00000000 00000000    00000000 00190000 E2E3C1D9 E3E4D740    .......................STARTUP
            016003B0 0080    E2E3C1D9 E3E4D740                                                            STARTUP
    0008    016003B8 0000    0007D2C4 C3E27E7E C031E3AF 55C9BD00    C3D6D5E3 40004040 40404040 40404040    ..KDCS==..T..I].CONT .
            016003D8 0020    40404040 00000000 40404040 40404040    40404040 40404040 4040D9C9 0000D9C9                                RI..RI
            016003F8 0040    D5D2F0F0 F0E3F0F0 F0F0D9D4 C6404040    4040D9D7 C9404040 40407E7E 00000000    NK000T0000RMF     RPI   ==....
            01600418 0060    00000000 00000000 D3D6C7E3 C5D94040    C2C5D5C9 C4404040 7E7E7E7E 7E7E7E7E    ........LOGTER BENID  ========
            01600438 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0009    01600440 0000    0008D2C4 C3E27E7E C031E3B4 6C697F00    E6C1C9E3 40400000 00004040 40004040    ..KDCS==..T.%.".WAIT  ....
            01600460 0020    40404040 40404040 40400000 00004040    40404040 40404040 40407E7E 00000000    ..                          ..
            01600480 0040    4040F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 00000000    000T0000                    ==....
            016004A0 0060    00000000 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            016004C0 0080    7E7E7E7E 7E7E7E7E                                                            ========
    000A    016004C8 0000    0009E5C7 C9C47E7E C031E3B4 6C6A6B00    C6050001 00000001 00000006 00000000    ..VGID==..T.%^..F...............
            016004E8 0020    00000000 00000000 00000000 00000000    00000000 00000000 00000000 00000000    ...............................
=           01600528 0060    00000000 00000000 00000000 00000000    00000000 001A0000 D2C4C3E2 C7D5E3C3    .......................KDCSGNTC
            01600548 0080    D2C4C3E2 C7D5E3C3                                                            KDCSGNTC
    000B    01600550 0000    000AD2C4 C3E27E7E C031E3B4 6CD02000    C9D5C9E3 00000002 02580000 00000000    ..KDCS==..T.%...INIT............
            01600570 0020    00000000 00000000 00000000 00000000    00000000 00000000 40404040 00004040    ............
            01600590 0040    4040F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A66DA     000T0000                    ==a...
            016005B0 0060    0101E02A 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            016005D0 0080    7E7E7E7E 7E7E7E7E                                                            ========
    000C    016005D8 0000    000BD2C4 C3E27E7E C031E3B4 6CD34800    C9D5C6D6 E2C90034 00000000 00000000    ..KDCS==..T.%L..INFOSI..........
            016005F8 0020    F040F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A693E    F040F0F0 F0E3F0F0 ...........OO
            01600618 0040    F040F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A693E    0 000T0000                  ==a...
            01600638 0060    0101E02A 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            01600658 0080    7E7E7E7E 7E7E7E7E                                                            ========
    000D    01600660 0000    000CD2C4 C3E27E7E C031E3B4 6CD6C700    E2C9C7D5 E2E30030 00000000 00000000    ..KDCS==..T.%OG.SIGNST..........
            01600680 0020    00000000 00000000 00000000 40404040    40404040 40404040 40404040 0030C3F0    ............................CO
            016006A0 0040    F140F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A69FC    1 000T0000                  ==a...
            016006C0 0060    010A5690 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            016006E0 0080    7E7E7E7E 7E7E7E7E                                                            ========
AUTM Dump Utility (V06.3A  ) - BS2000 Application = STRBLS        TSN = 686Q    ITN = FFFFE52C   Dump Nr.  2            Page    20
            REASON = PENDER    CREATED  15-01-21  11:03:55      PRINTED  15-01-21  11:06:21

    000E    016006E8 0000    000DD2C4 C3E27E7E C031E3B4 6CDA2500    D4D7E4E3 D5C50000 00A14040 40404040    ..KDCS==..T.%...MPUTNE...⁻
            01600708 0020    40405CC6 D6D9E2C9 C7D50001 00000000    00000000 00000000 00004040 0000C3F0     *FORSIGN.................CO
            01600728 0040    F140F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A6BA2    1 000T0000                  ==a.,s
            01600748 0060    0101E02A 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            01600768 0080    7E7E7E7E 7E7E7E7E                                                            ========
    000F    01600770 0000    000ED2C4 C3E27E7E C031E3B4 6CDF7300    D7C5D5C4 D2D70000 0000E3C1 C3E2C9C7    ..KDCS==..T.%...PENDKP....TACSIG
            01600790 0020    D5F20000 00000000 00000000 00000000    00000000 00000000 00004040 0000C3F0    N2........................CO
            016007B0 0040    F140F0F0 F0E3F0F0 F0F04040 40404040    40404040 40404040 40407E7E 810A6C52    1 000T0000                  ==a.%.
            016007D0 0060    0101E02A 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            016007F0 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0010    016007F8 0000    000FC6C8 C3D37E7E C031E3B4 6CE0F000    0800F0F0 F0F00000 00027E7E 00014040    ..FHCL==..T.%.0...0000....==..
            01600818 0020    0135A081 88000000 00000000 40404040    40404040 40404040 40404040 40404040    ...ah.......
            01600838 0040    40404040 40404040 40404040 40404040    0101E02A FFFFFFFF 7E7E7E7E 7E7E7E7E                 ....~~~~========
            01600858 0060    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================================
            01600878 0080    7E7E7E7E 7E7E7E7E                                                            ========
    0011    01600880 0000    0010D2C4 C3E27E7E C031E3CD 21936701    C3D6D5E3 4000C6C8 F0F14040 40404040    ..KDCS==..T..l..CONT .FH01
            016008A0 0020    40404040 00000000 40404040 40404040    40404040 40404040 40404040 0000C3F0    ....                    ..CO
            016008C0 0040    F140F7F0 E9E3C6C8 F0F14040 4040F0F0    F0F04040 40404040 40407E7E 810A6C52    1 70ZTFH01    0000        ==a.%.
            016008E0 0060    00000000 00000001 E3D7F6F0 F0F0F0F1    40404040 40404040 7E7E7E7E 7E7E7E7E    ........TP600001        ========
            01600900 0080    7E7E7E7E 7E7E7E7E                                                            ========
AUTM Dump Utility (V06.3A  ) - BS2000 Application = STRBLS        TSN = 686Q    ITN = FFFFE52C   Dump Nr.  2            Page    21
            REASON = PENDER    CREATED  15-01-21  11:03:55      PRINTED  15-01-21  11:06:21
```

```
AUTM Dump Utility (V06.3A  ) – BS2000 Application = STRBLS      TSN = 686Q    ITN = FFFFE52C  Dump Nr.  2              Page   22
            REASON = PENDER   CREATED  15-01-21  11:03:55    PRINTED  15-01-21  11:06:21


   TABLE OF CONTENTS FOR DUMP NR.   2, REASON = PENDER, TSN = 686Q

         1       5     KDCROOT : Program Table
         2      89     KDCROOT : Load Module Table
         3     145     KDCROOT : Load Contxt Table
         3     152     KDCROOT : Area Table
         3     161     KDCROOT : Exit Table
         3     166     KDCROOT : Library Table
         4     177     KDCROOT : Memory Pool Table
         4     183     KDCROOT : Message Mod Table
         4     189     KDCROOT : UTM Diagarea
         6     293     KDCROOT : KB
         6     303     KDCROOT : SPAB
         6     319     KDCROOT : MPUT Buffer
         6     332     KDCROOT : FORMUSER Buffer
         6     339     KDCROOT : RESTART Buffer
         7     350     KDCROOT : IO Buffer
         7     358     KDCROOT : ROOTDATA
         8     404     KDCROOT : ROOT Trace
         9     467     KDCROOT : FORM USER Area
         9     474     KDCROOT : HLL USER Area
         9     479     KDCROOT : Root gen by user
        10     544     KDCROOT : Table Descriptors
        11     588     KDCROOT : Administration DIAGAREA
        11     597     KDCROOT : Administration USERAREA
        11     603     KDCROOT : ETPNDS—TU—LIST
        13     740     KDCROOT : ROOT—CMP—AREA
        13     748     KDCROOT : STRT—PAR—AREA
        15     803     KDCROOT : TAM
        15     810     KDCROOT : TSKM
        15     814     KDCROOT : Table Descriptors
        15     824     KDCROOT : DB Diagarea
        16     860     KDCROOT : DB User Area
        16     866     KDCROOT : VGM Area
        16     872     KDCROOT : DB Info Program Unit
        16     877     KDCROOT : DB Info Application
        16     881     KDCROOT : DB Summary Info
        16     890     KDCROOT : DB Table
        17     894     Summary information
        19     967     Recently written diag entries
        22    1054     TABLE OF CONTENTS FOR DUMP NR.   2, REASON = PENDER, TSN = 686Q
```

3.4.2 Sample dump STXIT

1. Look up DIAGAREA --> last call: PEND ER

    The return codes are found in the plain text of the PENDER entry or in the KB.

2. Look up KB

Transaction code:	(bytes 32-39):	COMMAND
Contents KCRCCC	(bytes 92-94):	70Z
Contents KCRCDC	(bytes 96-99):	XT5C

```
                        Meaning of  XT        5C


                The STXIT routine              Interrupt weight 5C:
                was called                     address error
```
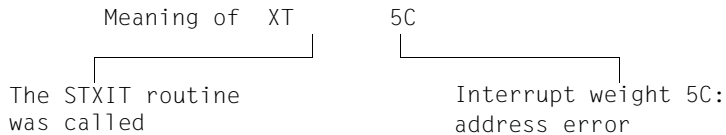
3. Analyze registers (CONTXT-AREA)

    The PC is in the middle of an instruction at an odd-numbered address. To enable further
    diagnosis, the program unit belonging to the transaction code COMMAND must be analyzed.

Example

```
CONTEXT-AREA....

  TABLE:  CONTEXT-AREA

        PC=010F44CE  IW=48 ILC=00  PM=0C  CC=00  AMODE=31  STXITPI + 00000476

        R00 00000002   R01 810F44B0    R02 01028040   R03 01033B00
        R04 0101A2C0   R05 010194EC    R06 01019000   R07 01051000
        R08 01AD0000   R09 00001AD0    R10 010F4058   R11 01011998
        R12 0154CB08   R13 015814B8    R14 810F41A4   R15 00000000


TABLE:  UTM-DIAGAREA

  0027   01601430 0000    0026D2C4 C3E27E7E C0039AFE 67F41400    ..KDCS==.....4..
         01601440 0010    D4C7C5E3 00000050 00000000 00000000    MGET...&........
         01601450 0020    00004040 40404040 40400000 00000000    ..     ......
         01601460 0030    00000000 00000000 00000000 0004D6C3    ..............OC
         01601470 0040    40D4F0F0 F0E3F0F0 F0F04040 40404040     M000T0000
         01601480 0050    40404040 40404040 40407E7E 810F41A4             ==a..u
         01601490 0060    010F4CC2 00000002 E3D7F6F0 F0F0F0F1    ..<B....TP600001
         016014A0 0070    D2D3C1E4 E2404040 7E7E7E7E 7E7E7E7E    KLAUS   ========
         016014B0 0080    7E7E7E7E 7E7E7E7E                      ========
  0028   016014B8 0000    0027D2C4 C3E27E7E C0039B01 C29EC601    ..KDCS==....B.F.
         016014C8 0010    D7C5D5C4 C5D97A40 404040E2 E3E7C9E3    PENDER:    STXIT
         016014D8 0020    40E6C9E3 C840E7E3 F4F840C5 D5E3C5D9     WITH XT48 ENTER
         016014E8 0030    C5C44040 4040405A 5A404040 40404040     ED     !!
         016014F8 0040    40404040 40404040 40404040 40404040
         01601508 0050    40404040 40404040 40407E7E 810F41A4             ==a..u
         01601518 0060    00000000 00000002 E3D7F6F0 F0F0F0F1    ........TP600001
         01601528 0070    D2D3C1E4 E2404040 7E7E7E7E 7E7E7E7E    KLAUS   ========
         01601538 0080    7E7E7E7E 7E7E7E7E                      ========
  0029   01601540 0000    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================
         016015B0 0070    7E7E7E7E 7E7E7E7E 7E7E7E7E 7E7E7E7E    ================
```

4 UTM message concept

When a UTM application executes, openUTM generates messages that provide you with information on particular events.

A **UTM message** consists of a **message number**, a fixed **message text** and variable parameters referred to as inserts. These **inserts** are dynamically assigned current values when the respective message is output. Examples of inserts are the name of the application or of the communication partner, counters, return codes, etc.

Each message is identified by means of its **message number**. UTM message numbers start with the letters K or P followed by a 3-digit number, e.g. K008.

UTM messages serve different purposes and can be sent to different recipients (**message destinations**). Within certain limits you can define message destinations yourself on an application-specific basis.

Some explanations and examples for UTM messages are listed below:

– A user can be informed of a certain event at the terminal by means of a message and can be requested to make an input if necessary.

   Example

   During the sign-on check the password was found to be invalid. openUTM requests that the user repeat the KDCSIGN input at the terminal.

– An event in the UTM application is logged by means of a message in the UTM log file SYSLOG. In this way, data is collected for monitoring the application run and for diagnostic purposes.

   Example

   A message is issued stating that the size of the page pool in KDCFILE has exceeded a certain level.

–   If certain UTM messages are generated, and you have defined the message destination MSGTAC for them, openUTM calls an event-driven program unit in the application (see section on "MSGTAC event service" in the openUTM manual „Programming Applications with KDCS"). This event service can, for example, issue asynchronous calls to the administrator using FPUT. This makes it possible to use programmed administration to react to events for which openUTM generates messages.

Example

If the event "service abort" occurs, message K017 or K055 is generated. The MSGTAC service can act on this message by locking the TAC, for example, and sending a message to the administrator.

In order to output a message, openUTM utilizes message modules specific to openUTM. These contain the properties and texts of all UTM messages.

You can, to a certain extent, structure the output of UTM messages for specific applications.

●   You can change the destination (recipient) of a message. Message destinations for specific applications are defined in a standard message module. You will find more information about changing the UTM message module in section "Modifying message output" on page 134.

●   You can change the message texts or translate them into a different language.

●   You can create a number of your own message modules and link them into the application. This gives you the option of tailoring the output of messages to your users, e.g. by issuing them in different languages.

●   You can include message inserts in the message text or remove inserts from it.

4.1   Message module, message definition file

When outputting a message, openUTM accesses the UTM message modules. These contain the following, among other things, for each message.

– the message number Knnn or Pnnn
– the message text
– the message destinations
– the inserts

The message definition file SYSMSH.UTM.063.MSGFILE and the following two system message modules are supplied with openUTM.

KCSMSGS (German message texts)
KCSMSGSE (English message texts)

The system message modules contain default settings for the message destinations and German or English message texts. The message definition file is used as an input file if the user wishes to change the UTM messages. It contains message texts in both English and German.

> **CAUTION!**
>
> You may only use the openUTM tools KDCMTXT and KDCMMOD to process the message definition file! Any other write access, such as with an editor will **destroy** this file!

The message definition file can be expanded with message texts in other languages. You can translate the message texts and enter them in the message definition file with the KDCMTXT tool (see page 136ff). From the message definition file you can use the KDCMMOD tool (see page 143ff) to create your own message modules.

A certain type of message output is defined in the standard message modules. If you wish to change this, you must generate one or more of your own message modules.

You can use several message modules with message texts in different languages for UTM applications. This allows you to issue messages in the language assigned to each individual user.

4.2 Message destinations

openUTM generates messages while an application is active. Each message is sent to one or more of the following destinations:

SYSLST   Output to SYSLST.

SYSOUT   Output to SYSOUT.

STATION   Clients connected via a PTERM or a TPOOL with PTYPE≠APPLI, SOCKET or UPIC.

SYSLINE   System line of the terminal; user screen formats are maintained. If a message is to be output to the system line, both SYSLINE and STATION must be specified as destinations

CONSOLE   Console of the system operator; the application name is also output

PARTNER   Clients connected via a PTERM or a TPOOL with PTYPE=APPLI or SOCKET.

SYSLOG   System log file (see section "UTM log file SYSLOG" on page 153).

MSGTAC   MSGTAC program unit (see control statement TAC in the openUTM manual "Generating Applications" and in the section "MSGTAC event service" in the openUTM manual „Programming Applications with KDCS").

USER-DEST-1 to USER-DEST-4
              User-specific message destinations to which you can assign a USER queue, a TAC queue, an asynchronous TAC or an LTERM partner as concrete message destinations at generation time.

              Either cyclically or when requested by the user, WinAdmin and WebAdmin can retrieve messages from these UTM queues, output these in lists and save them in the configuration database if required.

              The assignment between USER-DEST-1..4 and the concrete destination is defined by means of the KDCDEF statement MSG-DEST.

4.2.1 Output format of the messages

Depending on the message definition, openUTM outputs the messages in different formats:

● to SYSLOG and MSGTAC:
  the message header and the current values of the parameters, as described in the
  section "Structure of UTM system messages" on page 158

● to CONSOLE:
  the application name, the message number and the message text with the text param-
  eters

● to USER-DEST-1...USER-DEST-4:
  – the message header and the current value of the inserts as described in the section
    "Structure of UTM system messages" on page 158, if USER-DEST-n is generated
    with MSG-FORMAT=FILE
  – the date/time followed by the message number, message text and inserts if USER-
    DEST-n is generated with MSG-FORMAT=PRINT

● to all other destinations:
  the message number and the message text with inserts. In certain error situations
  during an application run (abnormal termination of an application, program or task)
  openUTM may only output a message to SYSOUT, SYSLST or CONSOLE, even
  though the message was meant for other destinations (e.g. SYSLOG). This is done to
  prevent any further errors that might result.
  For the same reason, in certain error situations openUTM does not take the message
  text from the message module that can be modified by the user, but from the system
  message module.

Header with date and time for messages to SYSLST / SYSOUT

When outputting messages to the message destinations SYSLST and SYSOUT, by default
openUTM precedes the messages with a header containing the date and time. One benefit
of this is that it allows errors to be diagnosed more easily.

The header has the following format:

yyyy-mm-dd hh:mm:ss

Messages K038 and K044 are always output without this header. You can use the start
parameter UTM-MSG-DATE=NO to suppress the output of this header for all other
messages as well.

4.2.2   UTM messages to the BS2000 console

If UTM messages are to be output to the BS2000 console (message destination = CONSOLE), the BS2000 message file must contain the UTM message UTM0100. openUTM uses this message as a "container" for the appropriate current UTM message. openUTM enters the application name in addition to the message text in messages that are to be output to the BS2000 console.

4.2.3   UTM messages to a TS application

If UTM messages are intended for a transport system application of the type PTYPE=APPLI or SOCKET (message destination = PARTNER), you must ensure that the application recognizes the messages and responds accordingly. If it does not, it can happen, for example, when two UTM applications are linked as TS applications, that the applications keep sending each other messages such as
`K009 Transaction code K009 is invalid`.

USP header in the case of UTM messages to a socket application

For UTM messages to a socket application (PTYPE=SOCKET, message destination = PARTNER), you can specify at generation whether openUTM is to precede the message with a USP header (open**U**TM **S**ocket **P**rotocol header). One of the purposes of the USP header is to notify the length of the received messages to the socket partner.

To do this, specify the following at KDCDEF generation in the USP-HDR= operand of the PTERM or TPOOL statement:

`USP-HDR = MSG or USP-HDR = ALL`

If you generated `USP-HDR = NO` (default setting), a USP header is not created.

You will find a description of the USP header in the openUTM manual „Programming Applications with KDCS".

4.2.4 **UTM messages to user-specific message destinations**

When a message occurs for which USER-DEST-1 ...USER-DEST-4 has been defined as
the message destination, UTM creates an asynchronous job to this message destination
internally. This asynchronous job is assigned the user KDCMSGUS and the LTERM partner
KDCMSGLT as the originator. If the asynchronous job is rejected because, for example, the
message destination is locked (STATUS=OFF), the message is lost to the message desti-
nation. If a message is created for this message destination again, UTM tries to create an
asynchronous job again.

If an asynchronous TAC is generated as the message destination, UTM starts the program
assigned to the TAC every time the relevant message is created (only one message can be
read with FGET in a program run).

4.2.5 **UTM messages to MSGTAC**

MSGTAC is a special asynchronous program unit that users can program themselves (see
the openUTM manual „Programming Applications with KDCS").

If there is an MSGTAC program, and a message occurs for which MSGTAC was defined as
the message destination, the asynchronous service MSGTAC is started. The MSGTAC
program can read a number of messages in a single program unit run.

The MSGTAC program unit runs under the internal UTM user ID KDCMSGUS with
KSET=MASTER and PERMIT=ADMIN. It must be defined in a TAC statement with
`TAC KDCMSGTC,PROGRAM=... .`

4.3 Modifying message output

The message definition file defines the way in which openUTM issues standard messages,
i.e. which text is used, which destination it is sent to, etc., together with constraints as to the
extent to which they can be modified. If you wish to change these default characteristics,
you must generate your own message module and link it to the UTM application. The
following changes may be made:

– Adding or removing message destinations (with KDCMMOD), such as logging further
  messages to SYSOUT or entering the destination MSGTAC as a requirement for using
  an MSGTAC service, for example.

– Translating message texts into a different language and incorporating them in the
  message definition file with the KDCMTXT tool (page 136).

– Changing message texts with the KDCMMOD tool (page 143), such as adding or
  removing inserts (defined for the message) to/from a message text, converting texts
  into lowercase/uppercase letters, etc.

– Defining text constants. These can be used in many message texts. Control characters
  are also included here.

  Example

  You can have certain messages signaled audibly.

> **i** However, individually designed message output is only possible for UTM applica-
> tions that have been linked with the modified message module. It will have no effect
> on other UTM applications in the same computer and do not require any changes
> to the BS2000 system.

You can also create a number of your own message modules and link them into the appli-
cation. This gives you the option of tailoring the output of messages to your users, e.g. by
issuing them in different languages.
The message destinations are taken from an application-specific standard message
module (see "Generation of a user-specific message module" on page 135).

Users may only perform modifications to the message definition file with the aid of the tools
KDCMTXT and KDCMMOD.

The KDCMTXT tool should be used when translating all message texts into another
language. If only a few texts are modified for a language contained in the message
definition file, the KDCMMOD tool can also be used.

Generation of a user-specific message module

The KDCMMOD tool is available for generating a user-own message module. From the
message definition file and a user-generated modification definition it creates an
ASSEMBLER source program which, after compilation, yields the application-specific
message module. This is linked to the program units present in an application.

An application-specific module is checked for validity by openUTM in the start phase and is
then accessed instead of the system message module.

In cases where the application-specific module cannot be accessed, the system message
module is accessed instead (e.g. when terminating a UTM task). The system message
module is present in the UTM system code.

Users can also create a number of their own message modules and link them into the appli-
cation, but only one module can be the application-specific message module. The message
module that acts as the application-specific message module is specified by means of the
LOCALE parameter in the KDCDEF MESSAGE statement during generation. This appli-
cation-specific message module is also referred to as the standard message module. All
other message modules are user-specific message modules. The message destinations
are taken from the standard message module.

You will find more information on this topic, e.g. how to link your own message modules with
the application program, in the openUTM manual "Using openUTM Applications under
BS2000 Systems".

Messages in other languages

The KDCMTXT tool is available for the creation of a message module in other languages.
It can be used to extend the message definition file by adding message texts in other
languages.

Message output exceptions

When UTM messages are output during operation, openUTM takes the message text and
the current message destinations from the standard message module or from a user-
defined message module (if one or more user-defined message modules have been
generated). However, the following exceptions to this exist:

– For some messages output by ROOT modules (e.g. for K078), English message texts
  are programmed in, and the output for these messages thus cannot be changed.

– In the start and end handling of the tasks, for technical reasons openUTM cannot
  access the application's message module(s). In these cases, openUTM takes the
  message text and the message destinations from the KCSMSGS message module,
  which contains German message texts. When the UTM subsystem starts up,
  KCSMSGS is loaded as part of the UTM system code.

If you want openUTM to output all messages in English in an English-language application, proceed as follows:

1. Before the UTM subsystem starts up, delete the KCSMSGS message module from the SYSLNK.UTM.063.TPR library.

2. Copy the KCSMSGSE module from SYSLNK.UTM.063 there instead, and rename the entry from KCSMSGSE to KCSMSGS.

4.3.1   Messages in other languages - KDCMTXT tool

The message definition file SYSMSH.UTM.063.MSGFILE supplied with openUTM contains the texts of the UTM messages in German and English. If openUTM is to output messages in another language, users can translate the UTM message texts into this language and incorporate them in the message definition file by using the KDCMTXT tool.

⚠ **CAUTION!**

You can only make modifications and extensions to the message definition file using the KDCMTXT utility. Other methods of write access, e.g. using an editor will destroy the file!

KDCMTXT serves to insert the translations of the standard message texts into the message definition file. The texts must be passed to KDCMTXT in the form of special control statements. On the basis of this extended message definition file, KDCMMOD can then be used to create the ASSEMBLER source program for a new message module, containing the messages in the desired language. KDCMMOD is described starting on page 143. After it has been assembled, this message module can be linked into the user's application in place of the system message module.

Note

When defining message text, the inserts allowed for the corresponding message may only be used once.

4.3.1.1 Inputs and outputs

```
┌─────────────────────────────┐     ┌─────────────────────────────┐
│  Message definition file     │     │  Control statements with    │
│                              │     │  new message texts via      │
│                              │     │  SYSDTA                     │
└─────────────────────────────┘     └─────────────────────────────┘
              │                                    │
              ▼                                    ▼
        ┌─────────────────────────────────────────────┐
        │              KDCMTXT tool                     │
        └─────────────────────────────────────────────┘
              │                                    │
              ▼                                    ▼
   ┌──────────────────────┐           ┌──────────────────────┐
   │  Extended            │           │  Output log           │
   │  message definition  │           │  (SYSLST and SYSOUT)  │
   │  file                │           └──────────────────────┘
   └──────────────────────┘
              │
              ▼
   ┌──────────────────────┐
   │  KDCMMOD             │
   └──────────────────────┘
              │
              ▼
   ┌──────────────────────┐
   │  ASSEMBLER source    │
   │  code for a new      │
   │  message module      │
   └──────────────────────┘
```
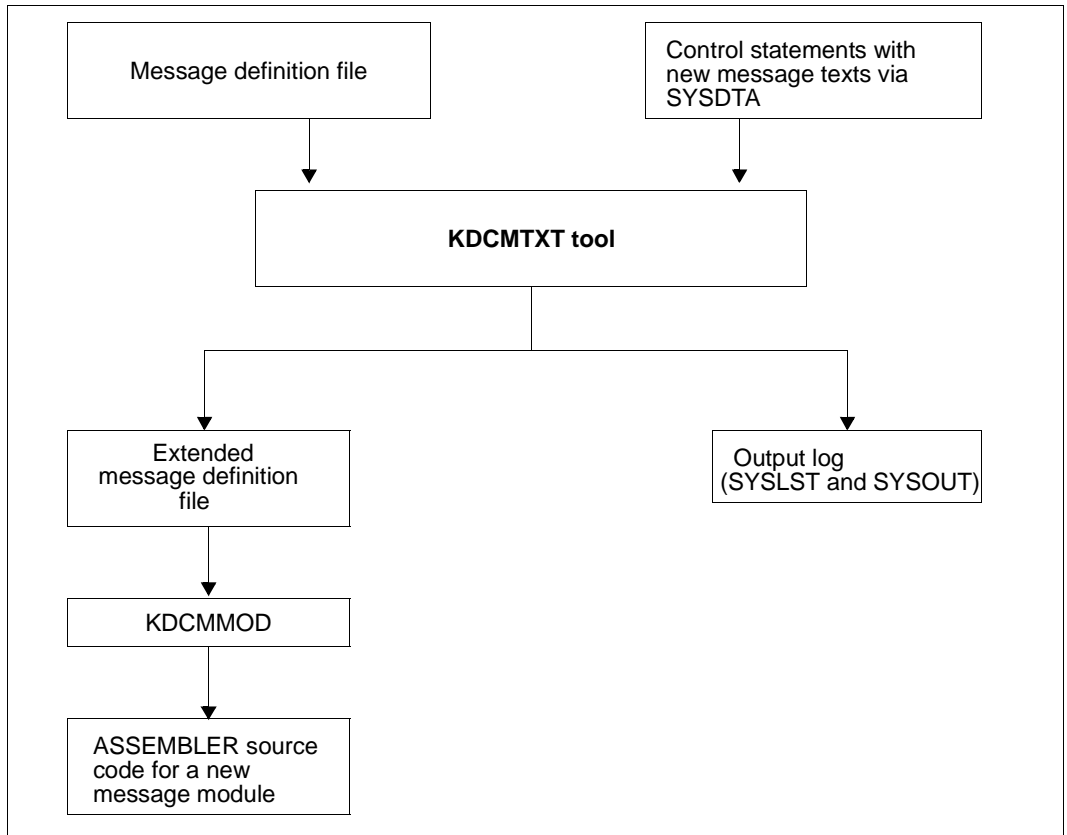
Creating messages in other languages with KDCMTXT

4.3.1.2 Calling KDCMTXT

The KDCMTXT tool is supplied in the SYSLNK.UTM.063.UTIL library. It is started by means of /START-EXECUTABLE-PROGRAM, e.g.

```
START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL
                ,ELEM=KDCMTXT)
```

KDCMTXT reads the control statements from SYSDTA.

> **i** You can also start KDCMTXT using the SDF START-KDCMTXT command, see also openUTM manual "Using openUTM Applications under BS2000 Systems", section, "Starting UTM tools via separate SDF commands".

4.3.1.3   Control statements for KDCMTXT

KDCMTXT recognizes the following control statements:

OPTION	Specify name of message definition file
FU	Define functional unit and natural language
MSGBASE	Select message group for FU SYS: UTM (K messages) or XAPTP (P messages)
STDTXT	Define start of standard message text
ENDTXT	Define end of standard message text
END	Define end of input of control statements

The following rules apply to the input of statements:

– Comment lines are marked by means of an asterisk (*) in column 1.

– If a line ends with a comma, KDCMTXT then interprets the following line as a continuation line of the statement.

It is expedient to write the control statements for KDCMTXT to a file.

When the control statement file KDCMTXT is created, the file SYSDAT.UTM.063.MTXTIN which is supplied with openUTM may be of assistance. This file contains the German and English standard message texts in the syntax of the KDCMTXT control statements. This file may be used as a template for the user's own input file.

OPTION statement

OPTION can be used to specify the name of the message definition file which is to be modified by KDCMTXT.

The OPTION statement must, if specified, be the first control statement; otherwise a syntax error will be reported. If no OPTION statement is specified, KDCMTXT uses the message definition file SYSMSH.UTM.063.MSGFILE supplied as standard with openUTM.

For safety reasons it is advisable to copy the message definition file and work on the copy with KDCMTXT.

Operation	Operands
OPTION	MSGFILE=filename

filename   Name of the message definition file which is to be extended.

Default: SYSMSH.UTM.063.MSGFILE

FU statement

This statement can be used to define the functional unit and the natural language.

Operation	Operands
FU	[functional-unit][,][LANG=language-id]

functional-unit

> Functional unit for which the new natural language is to be entered. At present, the only value permitted is SYS. SYS represents the transaction monitor.
>
> SYS: K messages (K001-K399) or P messages (P001 - P049)
>
> Default: SYS

LANG=language-id

> The language ID can consist of a maximum of 3 characters. It can be freely assigned, but it must be unique within a message definition file.
>
> You can use the language IDs found on cars or the language IDs defined in ISO IS/R639, for example.

Country IDs on cars	ISO IS/R639	Language
DK	Da	Danish
D	De	German
GB	En	English
E	Es	Spanish
FI	Fi	Finnish
F	Fr	French
GR	Gr	Greek
I	It	Italian
NL	Nl	Dutch
N	No	Norwegian
P	Pt	Portuguese
S	Sv	Sweden

> Default: GB

It is only necessary to specify the comma if both *function-unit* and LANG=*language-id* are specified, e.g. FU SYS, LANG=DK.

The LANG= option is only used to identify the texts in the message definition file and bears no relation to the language identifier for a user generated with the KDCDEF parameter LOCALE.

For any given functional unit, all the message texts for one language must be defined for one language so that the message texts for this language and functional unit are incorporated in the message definition file.

After the FU statement has been processed, the MSGBASE UTM statement is executed implicitly.

⚠ **CAUTION!**
The message definition file supplied already contains the German and English message texts with language identifiers D and GB. If one of these identifiers is specified, KDCMTXT overwrites the corresponding standard message texts in the message definition file with the new message texts.

MSGBASE statement (only for FU SYS)

openUTM makes a distinction between two message groups within FU SYS: Messages of the UTM group, which start with the letter K, and messages of the XAPTP group, which start with the letter P.

The MSGBASE statement allows you to select the message group to which subsequent STDTXT statements are to apply until the next MSGBASE or FU statement is issued.

Operation	Operands
MSGBASE	{ UTM | XAPTP }

UTM   Select the UTM message group (K messages).

XAPTP   Select the XAPTP message group (P messages).

STDTXT and ENDTXT statements

The STDTXT control statement introduces the definition of the standard text for a message. The following input lines describe the standard text.

The ENDTXT statement closes the definition of the message text. The translated message text is passed to KDCMTXT as follows:

Operation	Operands
STDTXT	msg–no text
ENDTXT	

msg-no   Message number for which the standard text is intended.

                 msg–no specifies the K or P message that is to be modified. The message number must lie within the message number range of the specified functional unit. The message number must be specified as three digits **without** the prefixed **K** or **P** .

                 Mandatory operand.

text   The message text must be specified here; the syntax is the same as with the KDCMMOD utility (see page 149). The same length restrictions similarly apply.

The control statements STDTXT and ENDTXT and also the message text must each begin in a new line.

The STDTXT / ENDTXT statements for a single functional unit and a single language must all be located after the FU statement for this functional unit and must precede the next FU statement.

KDCMTXT will only incorporate the message texts for an additional language in the message definition file, if the input data contains message texts for all the messages for this language and if there are no errors in the entries. If the message texts for a language are already present in the message definition file, KDCMTXT will then also accept individual messages. Existing message texts with the same message number and the same language identifier are then overwritten.

END statement

This statement terminates the input of control statements to KDCMTXT. END has no operands.

4.3.1.4 **KDCMTXT log**

After the KDCMTXT tool has processed all the modifications and created a new message definition file, it outputs an edited list of all messages to SYSLST and SYSOUT as a log (output log). The list contains the message texts edited in the form in which they would be output to a data display terminal. Here, the inserts are filled to the appropriate length as follows, depending on type:

Type	Fill character
CHAR	@
INT	#
HEXA	X

4.3.2 **Generating a user-specific message module with KDCMMOD**

As a UTM user you can make certain modifications to the standard messages. For this
purpose the KDCMMOD tool is provided which creates a file from the message definition
file and a modification description written by you. This file contains an ASSEMBLER source
program which after compilation creates a new user-specific message module. This new
module can then be linked to the application program in place of the system message
module and used.

The KDCMMOD tool stores the generated ASSEMBLER source program in the file
SRC.ASSEMB.object-module-name, where object-module-name denotes the name of the
message module specified in the GEN statement.

The following can be specified or modified:

– the natural language in which the messages are to be generated

– the fixed message texts

– the message destinations

– message attributes

– control characters.

There is considerable scope for modifying the message texts but the length restrictions
must be observed; see page 150.

It is possible to remove inserts from the message text that are contained in the standard text
or include inserts of a message that were not contained in the standard text. In addition, it
is possible to alter the sequence of the inserts within the text. Each insert may only appear
a maximum of once in the message. The message numbers can also be moved to another
place in the message text or even removed altogether. However, this should only be done
in certain exceptional cases since it can make diagnoses on the basis of the messages
more difficult.

The message destinations can be modified within certain limits (see section "Destinations
of UTM messages" on page 404). You should also note that different destinations may be
required or prohibited for each message.

4.3.2.1   KDCMMOD inputs and output
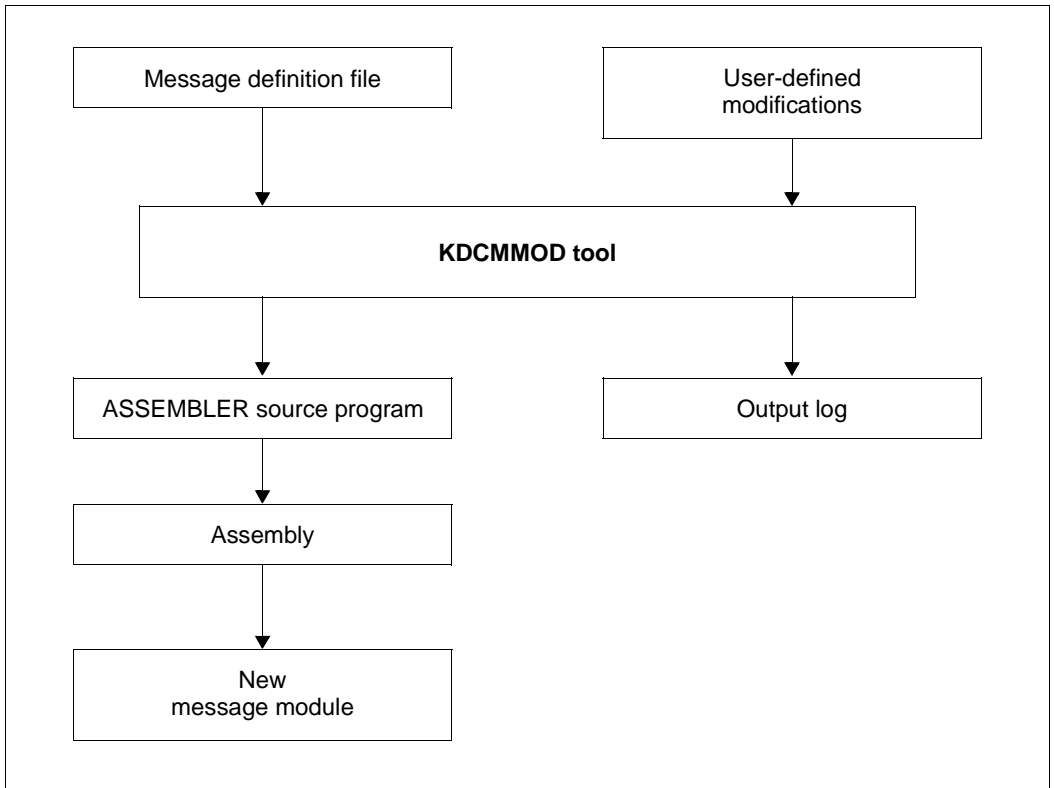
Sequence for creating an own message module

Once the KDCMMOD tool has processed all the changes, it writes an edited list of all messages to SYSOUT as a log. In addition, the program creates an overview of the message definitions (insert list, destinations, compression) derived from the framework definitions and modifications.

In the output list the defined standard texts are edited in the form in which they would be output to a terminal, with the inserts being filled as follows to the appropriate length depending on the type:

CHAR   with   @-characters,
INT   with   #-characters,
HEXA   with   X-characters.

The messages for KDCMMOD may be found starting on .

4.3.2.2 Starting and terminating KDCMMOD

The KDCMMOD tool is started as follows:

```
START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL
                    ,ELEM=KDCMMOD)
```

> **i** You can also start KDCMMOD using the SDF START-KDCMMOD command, see also openUTM manual "Using openUTM Applications under BS2000 Systems", section, "Starting UTM tools via separate SDF commands".

KDCMMOD reads the statements of the modification description from the system file SYSDTA. Individual lines are read, whereby

– a comment line is marked by a "*" character (asterisk) in the first column
– for lines that have a comma as the last character the next line is interpreted as a continuation line.

All other lines are analyzed individually by the program.

The program run of KDCMMOD is terminated with the control statement END.

4.3.2.3 Control statements for KDCMMOD

KDCMMOD recognizes the following control statements:

CONSTANT	Define constants
END	Terminate input of control statements
ENDMSG	Terminate message definition
GEN	Generate messages for a functional unit
MODMSG	Modify messages
MSGBASE	Select message group: UTM (K messages) or XAPTP (P messages)
OPTION	Specify name of the message definition file

Observe the following rules regarding sequence of KDCMMOD control statements:

1. If used, the OPTION statement must be the first control statement.

2. The CONSTANT statement must be in the input file before a text definition can refer to it.

3. The GEN statement must appear in the input file before any associated MODMSG/ENDMSG control statements.

4. The MSGBASE statement affects all subsequent MODMSG statements.

5. The END statement must be at the end of the input file.

If an error occurs in a statement for KDCMMOD, the incorrect statement is ignored following output of a corresponding error message (K6xx). This does not terminate the program run.

CONSTANT statement

The CONSTANT statement defines a constant which can susequently be used in the definition of message texts.

Operation	Operands
CONSTANT	constant-name, constant-value

constant-name   denotes the name of a constant and can be up to 8 characters long. If a constant with the same name already exists, the CONSTANT statement is rejected with an error message.

                        Mandatory operand.

constant-value   assigns a value to a constant. The value must be specified either in the hexadecimal format (X'......') or in the printable format '...'. The maximum length is 50 characters.

                        Mandatory operand.

Several constants are already contained in the message definition file (see also). You can find further logical control characters in the VTCSET macro.

END statement

The statements for KDCMMOD are terminated with the **END statement**. END must be entered as the last statement.

Operation	Operands
END	

No operands.

ENDMSG statement

The message text definition is closed with the **ENDMSG statement**.

Operation	Operands
`ENDMSG`	

No operands.

The ENDMSG statement must always be in a separate line.

GEN statement

The **GEN statement** specifies for which functional unit and in which natural language the messages are to be generated. The HGEN statement may be given only **once** per program run and must appear before any MODMSG statements.

Operation	Operands

`GEN`	`[functional-unit]`
	`[,LANG=language-id]`
	`,MODULE=object-module-name`

functional-unit

> Name of the functional unit for which all messages are to be generated. The only currently possible functional unit is SYS, i.e. the messages of the transaction monitor are generated.
>
> Default value: SYS

LANG=language-id

> denotes the natural language for which the message texts are to be generated. A language identifier must be specified for which message texts are contained in the message file.
>
> Default value: GB

MODULE=object-module-name

> Denotes the name of the message module.The name corresponds to the MODULE name in the MESSAGE statement of the KDCDEF tool (see the section "KDCDEF control statements" in the manual "Generating and Handling Applications").
>
> Mandatory operand.

MODMSG statement

You use the MODMSG statement to modify a message in the functional unit that was specified in the GEN statement. The modifications can be made to the text, the insert and the message destinations. Only those modifications which are permitted under the general framework definition are possible (see).

Operation	Operands
MODMSG	`msg-no`
	`[,BEL= {Y`ES `| N`O`}` [1]
	`[,COMPRESSION= {YES	N`O`}]`
	`[,EXTEND= {YES	N`O`}]`
	`[,CONSOLE= {YES	N`O`}]` `[,MSGTAC= {YES	N`O`}]` `[,PARTNER= {YES	N`O`}]` `[,STATION= {YES	N`O`}]` `[,SYSLINE= {YES	N`O`}]` `[,SYSLOG= {YES	N`O`}]` `[,SYSLST= {YES	N`O`}]` `[,SYSOUT= {YES	N`O`}]`
	`[,USER-DEST-1= {YES	N`O`}]` `[,USER-DEST-2= {YES	N`O`}]` `[,USER-DEST-3= {YES	N`O`}]` `[,USER-DEST-4= {YES	N`O`}]`
	`[text]` [2]

[1] The operands of the MODMSG statement must be separated by commas. When continuation lines are used in a MODMSG statement, the comma must always be the last character in the previous line (as a continuation character).

[2] The line preceding [text] must not contain a comma at its end.

msg-no   indicates which K or P message is to be modified. The message number must be in the message number range of the functional unit specified in the GEN command. The message number must be given **without** the **K** or **P** prefix.

Mandatory operand.

BEL= indicates whether an audible signal is given when the message is output to the destination STATION or SYSLINE.

    YES An acoustic signal is triggered.

    NO No acoustic signal is triggered.

        Default for all messages: NO

COMPRESSION=

    YES Superfluous blanks are removed from the message.

    NO Superfluous blanks remain in the message.

        The default value varies for the different messages. To find out the value set as the default for the individual messages, refer to section "Destinations of UTM messages" on page 404.

EXTEND=

    YES The message text is to be output in extended line mode.

    NO The message text is not to be output in extended line mode.
        Default: framework definition in SYSMSH.UTM.063.MSGFILE.

        The default value varies for the different messages. To find out the value set as the default for the individual messages, refer to section "Destinations of UTM messages" on page 404.

Message destinations CONSOLE, MSGTAC etc.
        denotes the message destination to which the message is to be sent (YES) or not sent (NO). Only those destinations may be specified that are 'allowed' (+) or declared 'standard' (S) in the framework definitions (see table on page 389). For all message destinations that are not specified in the MODMSG command, the framework definitions apply.

        To assign the user-specific message destinations USER-DEST-1,..., USER-DEST-4 to the concrete message destination, you have to use the KDCDEF statement MSG-DEST.

        You will find detailed descriptions of the various message destinations on page 130.

text After the MODMSG command has been issued a new message text can be defined. If no new text is defined, the text as described in the message definition file is used.

        A new message text is defined immediately after the MODMSG command in one or more lines. It consists of a series of **text elements** separated either by commas or by end-of-line characters.

Text = text-element,text-element,.....
text-element,.....

The first text element must always occur in a new line.

Length restrictions

The message text, including the message number and any inserts which may be present, must not exceed 512 characters in length. For the lengths of the inserts, refer to the table on page 392. If the text is longer than 512 characters, KDCMMOD rejects it with message K686. The message definition is not then updated.

Message texts having the message destination SYSLINE (=systemline) must not exceed 80 characters in length. If the text is longer than 80 characters, KDCMMOD gives a warning with message K687. On output, longer messages are limited to 80 characters and the excess is truncated.

The following text elements can be used:

Literal A literal is a character string enclosed in single quotes. Double quotes in a literal are used to designate a single quote in the literal.

Insert The text element insert is the name of a field (insert) in the message text that is preceded by the "&" character. Only such field names (inserts) may be used in the text that are specified in the framework definition of the corresponding message. These field names are listed in section "Destinations of UTM messages" on page 404.
Insert fields are areas in the message texts into which current values are inserted before the message is output.

Constant The text element constant is the name of a constant which is preceded by the "#" character.

Built-in function

Function that is executed at the point in question. The following built-in function is available:

MSGID
To simplify diagnosis, all message texts should start with MSGID.

This function returns the character string for the message identifier which corresponds to the message number.

Each MODMSG statement must be terminated with an **ENDMSG** statement.

MSGBASE statement (for FU SYS)

openUTM makes a distinction between two message groups within FU SYS: Messages of the UTM group, which start with the letter K, and messages of the XAPTP group, which start with the letter P.

The MSGBASE statement allows you to select the message group to which subsequent MODMSG statements are to apply within the SYS functional unit.

Operation	Operands
MSGBASE	{ <u>UTM</u> | XAPTP }

UTM   Default. Select the UTM message group (K messages).

XAPTP   Select the XAPTP message group (P messages).

OPTION statement

The **OPTION statement** is used to define the name of the message definition file which the KDCMMOD tool is to process.

Operation	Operands
OPTION	MSGFILE=filename

filename   Name of the message definition file to be processed.

If the OPTION statement is omitted, the file SYSMSH.UTM.063.MSGFILE is used by default.

Example

The user wishes to assign the destination MSGTAC for message K006 but otherwise retain the unchanged messages with English texts.

The following procedure provides the ASSEMBLER source file for the desired user-specific message module where the message definition file is called $TSOS.SYSMSH.UTM.063.MSGFILE and the KDCMMOD is contained in the SYSLNK.UTM.063.UTIL library:

```
/BEGIN-PROCEDURE LOGGING=N
/ASSIGN-SYSDTA TO=*SYSCMD
/ASSIGN-SYSLST TO=LIST.KDCMMOD
START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(LIB=SYSLNK.UTM..063.UTIL
                 ,ELEM=KDCMMOD)
```

```
OPTION MSGFILE=$TSOS.SYSMSH.UTM..063.MSGFILE
GEN SYS,LANG=D,MODULE=KCSMSGSD
MODMSG 6, MSGTAC=YES
ENDMSG
END
/ASSIGN-SYSDTA TO=*PRIMARY
/ASSIGN-SYSLST TO=*PRIMARY
/END-PROCEDURE
```

As a result of the procedure, the KDCMMOD tool generates the ASSEMBLER source file
SRC.ASSEMB.KCSMSGSD. This file must be assembled, the result of the assembly being
an object module with the name KCSMSGSD, which contains the user-specific messages.

In order for a user-specific message module to be used, the statement MESSAGE
MODULE=KCSMSGSD, LIB=... must be included in the generation with KDCDEF.

4.4 UTM log file SYSLOG

openUTM writes a log file for each application, the SYSLOG file. openUTM writes all the messages which indicate SYSLOG as their destination to this file. You can - within certain limits - specify which messages these are to be (see section "Destinations of UTM messages" on page 404). The SYSLOG file must be created by the user before the UTM application is started.

You can create the system log file SYSLOG as:

– a simple file with the name *filebase*.SLOG or a file with the link name SYSLOG. *filebase* is the base name of the KDCFILE from the MAX statement.

– a file generation group SYSLOG-FGG (**F**ile **G**eneration **G**roup)
It is sufficient to create a file generation group; the individual generations of the FGG are created by openUTM for you.
If you have generated automatic monitoring of the size of the SYSLOG file, then the SYSLOG must be created as an FGG.

The openUTM manual "Using openUTM Applications" describes in detail in chapter 8 how to create the system log file SYSLOG.

Events occurring during the execution of the application which could be useful for monitoring purposes or for making checks later are logged in the SYSLOG file in the form of UTM messages (e.g. K033, K070). In particular, the SYSLOG file contains important information for diagnostics.

Each time the application is started subsequently, the SYSLOG file is overwritten by openUTM. The logging information from the previous run is lost. Therefore, you should backup the contents of the file if necessary when the application terminates.

4.4.1 Evaluating the SYSLOG file

openUTM creates the SYSLOG file in PAM format. To print out this file in readable form or edit it using a program, you must process the file. openUTM offers the following editing tools for this purpose:

● KDCCSYSL

Converts the PAM file into a SAM file. KDCCSYSL is provided in the SYSLNK.UTM.063.UTIL library.

● KDCPSYSL

From the SAM file, KDCPSYSL creates another SAM file which also contains the message texts. KDCPSYSL is provided in the SYSLNK.UTM.063.UTIL library.

To make it easier to evaluate the SYSLOG file, the following sample procedures are
supplied with openUTM:

● SYSPRC.UTM.063(SYSLOG)

    This procedure allows you to evaluate a simple SYSLOG file or an individual file gener-
    ation of a SYSLOG-FGG. <filebase>.SYSLOG is expected as the input file.

● SYSPRC.UTM.063(SLOG-FGG)

    This procedure allows you to evaluate an individual or all file generations of a SYSLOG-
    FGG. In one loop, the procedure calls the two tools KDCCSYSL and KDCPSYSL for the
    individual file generations. The evaluation can also be made while the application is
    running.

4.4.1.1 The KDCCSYSL tool – converting the SYSLOG file to SAM format

The KDCCSYSL tool (convert SYSLOG) converts the system log file created by openUTM
as a PAM file into a SAM file. For each UTM message, KDCCSYSL creates a record of
variable length. The structure is described by KCMSGx data structures, which are available
for all programming languages and are supported by openUTM, for example KCMSGC for
COBOL.

Calling KDCCSYSL

```
/SET-FILE-LINK FILE-NAME=file_in,LINK-NAME=SLOGPAM    -
               [,SUPPORT=DISC(SHARED-UPDATE=YES)]
/CREATE-FILE FILE-NAME=file_out,
/SET-FILE-LINK FILE-NAME=file_out,LINK-NAME=SLOGSAM,ACCESS-METHOD=SAM

/START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL
               ,ELEM=KDCCSYSL)
```

i	You can also start KDCCSYSL using the SDF START-KDCCSYSL command, see also openUTM manual "Using openUTM Applications under BS2000 Systems", section, "Starting UTM tools via separate SDF commands".

Meaning of parameters:

file_in   Name of the SYSLOG file

file_out   Name to be assigned to the SAM file

SHARED-UPDATE=YES must be specified in the SET-FILE-LINK command if the
SYSLOG file is to be edited while the application is running. In this case, however, note that
openUTM buffers SYSLOG records, i.e. does not write them directly to the PAM file.

If the SYSLOG file is to be evaluated while the application is running, it is advisable to issue the administration command KDCDIAG SYSLOG=WRITE beforehand. This means that SYSLOG UTM messages that may still be buffered in the virtual memory will be written immediately to the SYSLOG file by openUTM.
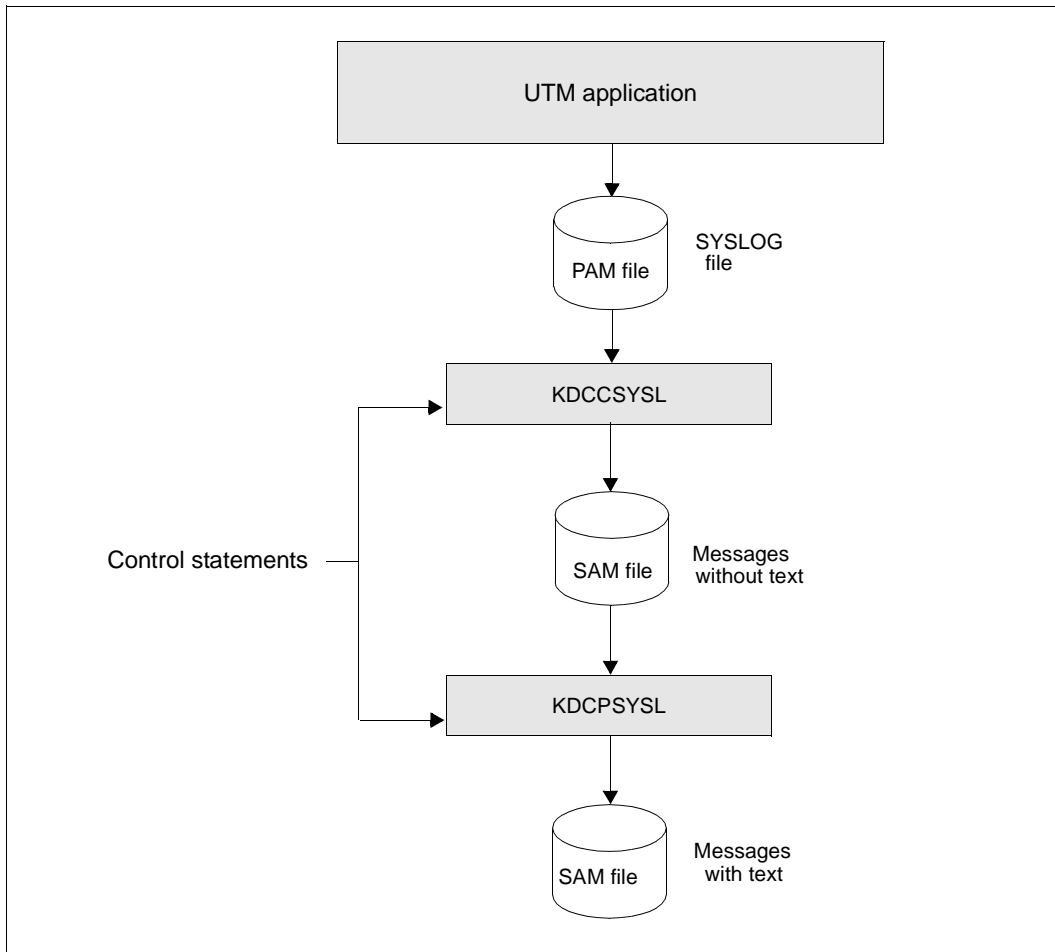
Editing the SYSLOG file

4.4.1.2 The KDCPSYSL tool – inserting message texts

The editing tool KDCPSYSL reads the SAM file created with KDCCSYSL and from it creates a SAM file which contains the message number and message text with current text parameters for each UTM message.

The date and time are inserted at the start of the UTM message. The inserts that are not already contained in the message text are added at the end. Inserts that provide hexadecimal information are prepared in printable form (e.g. BCAM info word in UTM message K065).

You can also edit the SYSLOG file with message texts from your own message module. In this case, enter the name of your message module after the UTM message:

```
K608 Please enter name of user message module, a blank, or /EOF.
```

If a user-specific module library was assigned beforehand with /SET-TASKLIB LIBRARY=, then KDCPSYSL searches for the specified module in this library.

If a blank or "/EOF" is entered in response to the UTM message or if the module is not found in the TASKLIB, then KDCPSYSL edits the UTM messages using the standard message module of openUTM.

The file created with KDCPSYSL is also a SAM file, which contains one or more print lines for each UTM message. This file, which can be used to obtain a quick overview, can be printed out using the PRINT-FILE command:

```
/PRINT-FILE FILE-NAME=file_out,LAYOUT-CONTROL=PARAMETERS(
            CONTROL-CHARACTERS=EBCDIC)
```

Calling KDCPSYSL

```
/SET-FILE-LINK FILE-NAME=file_in,LINK-NAME=SLOGSAM
/CREATE-FILE FILE-NAME=file_out,
/SET-FILE-LINK FILE-NAME=file_out,LINK-NAME=SLOGEDIT,ACCESS-METHOD=SAM
[/SET-TASKLIB LIBRARY=omlname]

/START-EXECUTABLE-PROGRAM FROM-FILE=*LIB-ELEM(LIB=SYSLNK.UTM.063.UTIL
                  ,ELEM=KDCPSYSL)
```

> **i** You can also start KDCPSYSL using the SDF START-KDCPSYSL command, see also openUTM manual "Using openUTM Applications under BS2000 Systems", section, Starting UTM tools via separate SDF commands".

KDCPSYSL responds with the UTM message:

```
K608 Please enter name of user message module, a blank, or /EOF.
```

Enter the following:

␣   or

K2 and /EOF and /R   or

message_module_name

Meaning of parameters:

file_in   Name of the output file of KDCCSYSL (=input file of KDCPSYSL)

file_out   Name of the output file of KDCPSYSL

omlname   Name of the library (OML) containing the user-specific message module

message_module_name
                Name of the user-specific message module

4.4.1.3 KDCCSYSL and KDCPSYSL messages

The editing tools KDCCSYSL and KDCPSYSL write their messages to SYSOUT (see section "Messages from the UTM tools KDCCSYSL and KDCPSYSL" on page 382).

4.5 Structure of UTM system messages

Structure of messages whose destination is SYSLOG or MSGTAC:

Message header	Message data, depending on message number
1 28	29

Structure of the messages:

Bytes	Format	Meaning
1	X'40'	Blank
2-5	C'Kxxx'	Message number
6	X'40'	Blank
7-17	C'mm/dd/yyjjj'	Date, where mm=month, dd=day of month, yy=year, jjj=day of year
18	X'40'	Blank
19-24	C'hhmmss'	Time of day, where hh=hour, mm=minute, ss=second
25-28	C'yyyy'	4-digit year specification
29 & up		Message data

This message header corresponds to the following data structure in the COBOL COPY element KCMSGC:

```
      05    MSGKOPF.
*                                     MESSAGE HEADER        *
      07    FILLER-1    PIC X.
*                                     FILLER                *
      07    MSGNR       PIC X(4).
*                                     MESSAGE NUMBER        *
      07    FILLER-2    PIC X.
*                                     FILLER                *
      07    MSGDATE     PIC X(11).
*                                     DATE OF ORIGIN        *
      07    FILLER-3    PIC X.
*                                     FILLER                *
      07    MSGTIME     PIC X(6).
*                                     TIME OF ORIGIN        *
      07    MSGYEAR     PIC X(4).
*                                     YEAR OF ORIGIN        *
**************************************************************
*         INSERTS OF MESSAGES                               *
**************************************************************
      05    KXXX        PIC X(152).
         .
```

5 UTM messages

The following tables contain all the messages that can be output by openUTM. The messages from the UTM tools are included in this list. The descriptions have been extended by supplementary information, which explains the reaction to the specific message.

The name of an insert is prefixed by "&". UTM messages can contain more inserts than standard messages. The meaning of the inserts may be found in the tables "Inserts in messages" on page 392ff and "Destinations of UTM messages" on page 404ff.

If a UTM application employs a sign-on service, messages K001, K002, K004, K005, K006, K007, K008, K030, K031, K097, K109, K110, K120, K123, K125 and K145 are not generated.

5.1 Messages of the transaction monitor

K001 Connected to application &APPL - input please

K002 Connected to application &APPL - please sign on

K003 Command &CMD is not permitted at this time

K004 User identification &USER is invalid - please sign on

K005 User identification &USER is locked - please sign on

K006 Invalid password - please sign on

K007 User &USER already signed on - please sign on

K008 Sign-on accepted - input please

K009 Transaction code &TAC is invalid (&RCDC) - input please

The insert &RCDC contains the incompatible return code KCRCDC, see section "Internal return code KCRCDC" on page 426.

K010 Transaction code &TAC is locked - input please

K011 Transaction &ATAC1 accepted - input please

K012 &NUMMSGS asynchronous message(s) present

K013   Error in &CMD - input please

K014   No base format defined - input please

K015   Formatting error &RCDC/&RCF2A - input please

K016   Application shutdown pending - please sign off

K017   Service &TCVG terminated by UTM (&RCCC/&RCDC &RCF2A) - input please

This message informs a terminal user of the abnormal termination of the service started by this user.

The complete message K017 of the SYSLOG file contains the following inserts:

&PTRM	Name of the PTERM from which the service was started.
&PRNM	Name of the processor to which the terminal is connected.
&BCAP	Name of the BCAM application to which the user has signed on.
&LTRM	Name of the LTERM from which the service was started.
&USER	Name of the user who started the service.
&TCVG	Name of the service TAC of the service.
&RCCC	Value of the compatible KCRCCC return code.
&RCDC	Value of the incompatible KCRCDC return code.
&RCF2A	Secondary FHS or VTSU return code (if any).
&TAC	Only for RCDC=KMxx return codes: Name of the TAC that caused the service to terminate. In all other cases this insert contains a blank.

K018   Sign-off for application &APPL accepted - please sign on

K019   Sign-off for application &APPL accepted

K020   No message(s) present

K021   No input within the specified period

K022   The following message from &BCAP may have already been sent.

K023   &OMSG2

The message is generated by the administration command KDCSEND or KDCSHUT WARN or the corresponding calls to the administrator interface.

K024   Input message lost - please repeat

K025 Message from application &BCAP to LTERM &LTRM was truncated

A message to an LTERM partner could not be transmitted completely, and was output in truncated form. The following error recovery actions are possible:

– increase the length specification in the TRMSGLTH operand for KDCDEF if this length is smaller than the maximum packet length for BCAM;

– increase the maximum packet length for BCAM if it is smaller than TRMSGLTH or

– insert "new line" (X'15') control character in output in line mode (in order to make fragmentation by VTSU possible).

K026 Broadcast to &LTRM accepted - input please

K027 Terminal &LTRM is locked - contact administrator or sign off

K028 Please enter password
> &PAS1 <

K029 Please insert card

K030 Card reader required - please sign on

K031 Wrong card - please sign on

K032 UTM-D connection message: &CON/&PRNM/&BCAP/&LPAP/&USER; reason1: &RCF1B; reason2: &RCF2B

In the following description of the return codes &RCF1B and &RCF2B, PLU (Primary Logical Unit) is the application in which SESCHA PLU=NO was generated, and SLU (Secondary Logical Unit) is the application in which PLU=YES was specified.

The insert &USER contains the UTM session name of the connection (LSES name).

&RCF1B	Meaning	
C01	BIND_REJECTED	BIND request or BIND response has been rejected. No session can be set up.
C02	BIND_CHANGED	The BIND parameters have been modified. The BIND request may be modified by the SLU; while PLU can reject but not modify the BIND response. If the BIND response cannot be accepted by PLU, no session is set up.
C03	CONNECTION / SESSION FAILED	A connection or session cannot be set up.
C04	SESSION_FAILED	Session warm start could not be carried out.
C05	STSN_CHANGED	SLU cannot accept the proposed orientation point.

&RCF2B	Meaning and possible action
CR00	Session and connection do not belong to the same LPAP. Action: Check the KDCDEF generation.
CR01	The sessions were not able to agree on a valid orientation point at session warm start. The applications were possibly started with different KDCDEF states. Action: Check whether one of the two applications has been regenerated.
CR02	SLU has suggested a new orientation point. If the session cannot be set up, the applications were probably started with different KDCDEF states. Action: Check whether one of the two applications has been regenerated.
CR03	Quiet command was entered.
CR04	No suitable session available; reasons: – More connections were generated than sessions – KDCLSES command issued with ACT=QUIET – A session has not yet been closed
CR05	Connection setup request from TRANSIT-CD[1] or partner application creates NEA protocols for this connection.
CR06	Connection being established.
CR07	Inconsistency of PLU entry in the connection message from the partner and in the generation. Both partners may have been generated as PLU. Action: Check generation.
CR08	Inconsistency of PLU entry in the connection message from the partner and in the generation. Both partners may have been generated as SLU. Action: Check generation.
CR09 CR10 CR11 CR12	Incorrect structure of a connection message
CR13	Resource bottleneck: no slot available for dynamic tables.
CR14	The name of the PLU session in the connection message is incorrect. Action: Check generation
CR15 CR16 CR19 CR21	Session not yet set up.
CR17 CR18 CR20	No suitable session available; reasons: – More connections have been generated than sessions – KDCLSES command issued with ACT=QUIET – A session has not yet been closed
SC00	No reason given.

&RCF2B	Meaning and possible action
SC01[1]	The partner is responding with an invalid FM profile. UTM supports FM profile 18 only.
SC02[1]	The partner is responding with an invalid TS profile. UTM supports TS profile 4.
SC03[1]	UTM can process a message in small units (request units). For this reason PLU must support so-called multiple RU chains.
SC04[1]	A request cannot be sent off until a previously requested response, if any, has arrived. PLU therefore has to operate in "immediate request mode".
SC05[1]	Depending on the message type, UTM either requests all types of acknowledgment or only a negative acknowledgment. PLU must therefore accept both definite and exception response.
SC07[1]	Data compression is not supported for message transmission.
SC08[1]	PLU must always be able to close a bracket.
SC09[1]	Corresponds to reason SC03, but for SLU.
SC10[1]	Corresponds to reason SC04, but for SLU.
SC11[1]	Corresponds to reason SC05, but for SLU.
SC12[1]	UTM does not support "2-phase commit".
SC13[1]	Corresponds to reason SC07, but for SLU.
SC14[1]	Corresponds to reason SC08, but for SLU.
SC15[1]	FM headers must be permitted.
SC16	Although one partner considers a session cold start to be sufficient, the other session insists on a session warm start.
SC17	he partner has indicated that the session has not terminated ("in bracket") without requesting a warm start. The session is assumed to be terminated (BETB).
SC18[1]	A service is not terminated until indicated to be terminated by both partners. This means that UTM makes use of the so-called "bracket termination Rule 1".
SC19	The USER-DATA in the BIND is not correct. Presumably the partner is sending a faulty BIND. The UTM BCAM trace is required for diagnosis.
SC23[1]	UTM-D operates with "half-duplex flip-flop" send or receive mode. This means that only one partner is authorized to send at any given time.
SC24[1]	Both partners must be responsible for a session restart, if any.
SC25	BIND and KDCDEF generation are inconsistent with respect to the contention winner. Action: Check the specification of the contention winner in both generations (KDCDEF control statement SESCHA CONTWIN=Y/N)

&RCF2B	Meaning and possible action
SC26	PLU is requesting a send authorization at session warm start. If the session is not set up, this is probably due to the applications having been started with different KDCDEF states. Action: Check whether one of the applications has been regenerated.
SC27	SLU is requesting a send authorization at session warm start. If the session is not set up, this is probably due to the applications having been started with different KDCDEF states. Action: Check whether one of the applications has been regenerated.
SC28[1]	openUTM supports "two-stage pacing".
SC29	Inconsistent pacing counter Action: Check pacing count (= window size) in both generations (KDCDEF control statement SESCHA PACCNT=)
SC30[1]	SLU is suggesting an illegal value for "maximum length of message segments from PLU" in the BIND response.
SC31[1]	SLU is suggesting an illegal value for "maximum length of message segments from SLU" in the BIND response.
SC32	PLU is suggesting an illegal value for "maximum length of message segments from SLU" in the BIND request.
SC33	PLU is suggesting a value for "maximum length of message segments from PLU" in the BIND request which is modified by SLU.
SC34[1]	Presentation Service (PS) Usage field format must be defined as "basic format".
SC35[1]	LU session type 6 is the only permissible session type.
SC36[1]	Message encryption is not supported.
SC37	Incorrect name of PLU session in BIND response. Action: Check association of LSES and RSES in both generations (KDCDEF control statement LSES RSES=name)
SC38	Incorrect name of PLU session in BIND request. Action: see reason SC37.
SC39	Incorrect name of SLU session in BIND response. Action: see reason SC37.
SC40	Incorrect name of SLU session in BIND request. Action: see reason SC37.
SC41[1]	User Request Correlation (URC) is not supported by UTM-D.
SY01	Session termination pending. Session cleardown has not yet been completed.

&RCF2B	Meaning and possible action
SY02	PET no send request The session is in the PTC status and there is no message for this partner.
SY03	PEND RS pending. PEND RS handling has not yet been completed.
SY04	Job-submitting conversation active. The job-submitting service is active.
SY05	Job submitter not available. The job submitter is not signed on or the job-submitting service is queued.
SY06	RESTART_VIA_SEND_RQ. The session is in the IN_BRACKET status and there is no message for this partner.
SY07	Session active The session is already active.

[1] Only for heterogeneous links

K033   USER / LSES active : &PTRM/&PRNM/&BCAP/&LTRM/&USER
&REST,&GLOBALSG

The K033 message is output

– following successful connection setup when working without USERs

– following successful KDCSIGN when working with USERs

– following successful linkage of a session in a LU6.1 connection).

&REST	Meaning
Y	Service restart
N	No service restart
U	Undefined, with UTM-D only

If a user signs on via an OSI TP connection, the inserts have the following contents:

Insert	Contents
&PTRM	OSI-CON name
&PRNM	8 blanks
&BCAP	ACCESS-POINT name
&LTRM	OSI-LPAP name

A value is only entered for the insert &GLOBALSG 'Cluster Global Sign' in UTM cluster applications. The insert can have the following values:

&GLOBALSG	Meaning
Y	Global sign-on at the cluster
N	Local sign-on at the node
A	The user was already signed-on globally to the cluster at this node.
' '	Sign-on for an LU6.1 session user or a connection user.

K034   Transaction has been reset

K035   Service restart in progress

K036   Connection setup: &PTRM, &PRNM, &BCAP, &LTRM, &RSLT, &REA1

&RSLT	Meaning
Y	Connection already set up
N	Connection has not been set up; the reason is indicated in &REA1

&REA1	Meaning
X'00'	Connection already set up
X'01'	PTERM/CON name unknown
X'02'	Processor name unknown
X'03'	PTYPE not accepted X' 03' (UTM generation and PDN generation do not match)
X'04'	No LTERM assigned
X'05'	Incorrect connection password
X'06'	STATUS=OFF for this PTERM/CON
X'07'	STATUS=OFF for this LTERM/LPAP
X'08'	Resource bottleneck
X'0A'	Application shutdown
X'0B'	Partner already connected
X'0C'	Connection cleardown being executed
X'0D'	Negative BCAM return code; see corresponding K065 message
X'0E'	Partner generated at another BCAM application
X'0F'	UTM-D connection request rejected, e.g. due to generation error or QUIET command
X'10'	PTERM name unknown and no LTERM pool available for this processor name

&REA1	Meaning
X'11'	No LTERM pool available for PTYPE and PTERM name unknown
X'12'	No further free entry available in terminal pool
X'13'	Partner already connected to LTERM pool
X'14'	Connection rejected due to insufficient characteristics of the transport connection (GROS)
X'15'	MUX connection rejected More detailed description in BIND response
X'16'	User is already connected
X'17'	The connection request has been rejected due to contention. The partner himself has initiated a connection setup.
X'18'	BCAM letter length too small
X'1B'	On application start-up, it was not possible to determine the name of the partner computer.
X'1F'	LPAP is set to quiet
X'20'	No suitable session is available
X'21'	The session is already active
X'22'	Session is being terminated
X'23'	Local page is in PTC and does not have an output message for this session; the session cannot be established until the local service is restarted.
X'24'	PEND RS is being processed
X'25'	The job-submitting service is active
X'26'	The job-submitting user is not available
X'27'	The session cannot be established until the local service is restarted
X'29'	Terminal pool is generated at another BCAM application
X'2A'	Connection request from VTSU rejected, for instance because of unsupported PTYPE or incorrect connection letter
X'2B'	The connection request for an LTERM partner generated with PTYPE = *ANY is rejected because the type of the calling terminal or printer is not supported.
X'2C'	Connection request of a printer for a PTERM partner generated with PTYPE = *ANY and USAGE=D is rejected.
X'2D'	An invalid MUX protocol version is suggested on setting up the MUX connection.
X'2E'	The connection has not yet been completely cleared down, or the MUX session is still in "RELEASE-PENDING" status.
X'44'	Connection request refused by RSO, for instance because the printer is not known to RSO

&REA1	Meaning
X'45'	A connection cancel request occurred while waiting for confirmation of the establishment of a connection
X'46'	PTERM/CON was deleted by the administration
X'48'	There is already a PTERM generated with the name of the multi-pool LTERM.
X'55'	Rejected socket connection setup.
X'58'	Confirmation of connection establishment rejected because connection has been interrupted again in the interim.

This message, which is normally output to SYSLOG, helps to detect problems arising at connection setup.
For message output, the &REA1 insert is edited in printable form while at the program interface with MSGTAC / SYSLOG it was passed as a hexadecimal value.

K038 Syntax error &SYN

K039 Startparameters CLUSTER-FILEBASE and &STRTPAR are mutually exclusive!!

K040 Warning level &WLEV for &PGPOOL no longer exceeded

K041 Warning level &WLEV for &PGPOOL exceeded

K042 Error reading start parameters from SYSDTA

This message is also output if no 2nd END command was specified.

K043 DMS error &DMSE for file &FNAM

The insert &DMSE outputs the DMS error code.

In the case of UTM cluster files, the insert &DMSE can contain the following error codes in addition to the error codes reported by the operating system:

ILCK   An error occurred while initializing a file lock.
LOCK   An error occurred while requesting a file lock.
ULCK   An error occurred while releasing a file lock.
DLCK   An error occurred while destroying a file lock.

K044 Please enter start parameters

K045 --- No message text ---

The only destination permitted for this message is MSGTAC; therefore no message text is defined for this message.

The message is output when the a message output in acknowledgment mode is terminated with a positive print acknowledgment.

No default destination is defined for this message.

K046 Print error: &PALTRM / &CID / &DPID / &ERPRT

The message is output only for printers that are assigned to a printer control terminal. It is primarily intended for programmed error handling with the help of the MSGTAC program unit.

Further inserts of this message are: &PTRM, &PRNM, &BCAP, &LTERM, &IMSG2

The message is output in the following cases:

– Termination of a printout with negative print acknowledgment,

– Repetition of a printout (in addition to the K022 message),

– Connection setup to a printer is rejected by BCAM (negative COMPLETION announcement).

No default destination is defined for this message.

The &ERPRT insert provides information on the error. Independent of this the &IMSG2 insert gives further information.

&ERPRT	Meaning and contents of &IMSG2
A	Negative print acknowledgment. If the printer sends a logical print acknowledgment (return message), &IMSG2 contains the first 32 bytes of this acknowledgment (possibly padded with spaces). The exact structure of this print acknowledgment is described in the respective equipment user's guides.
R	Repetition of a printout. &IMSG2 is without meaning.
C	Connection setup is rejected by BCAM (negative COMPL announcement). &IMSG2 contains BCAM INFOWORD and BCAM USERREASON.

K047   Invalid SVC 134 call (SVC for UTM): opcode = &OPCD1, RC = &RTCD

UTM outputs the message K047, if UTM SVC (SVC 134) cannot be processed successfully. The message contains the inserts &OPCD1 and &RTCD.

The insert &OPCD1 contains the KDCS operation code, which is 4 bytes long and can be printed. It is known by the KDCS calls. There are also internal UTM operation codes, e.g. 'STRT' (see description of UTM-DIAGAREA).

Insert &RTCD is the edited return code (8 characters, hexadecimal representation). The return code is set by UTM or by DSSM.

UTM SVC (SVC 134) return codes

UTM is the source of return codes with a value 000000nn. These codes have the following meaning:

nn	Meaning
04	Illegal ROOTDATA address (UTM error[1]).
08	Illegal ROOTDATA version. Possible causes: – The root connection module (KDCROOT) was generated with a version of UTM macro library that does not suit the UTM system code – Modules which do not suit the UTM version in the UTM macro library were linked to the application from SYSLNK.UTM.063.
0C	Invalid parameter list address (UTM error).
10	Invalid KDCS parameter area address. If the include files made available by openUTM for the KDCS parameter areas are used, this is a UTM error. If user-defined areas are used in the program unit for the KDCS parameter area, this is a user error. Action: check the alignment and address of the parameter area.
14	Illegal KTA address in ROOTDATA, possibly due to: – UTM error – KDCS call in START-EXIT program unit – KDCS call in SHUT-EXIT program unit – KDCS call after UTM Term Application
18	Illegal contents in KTA header. Possible causes: – UTM error – KDCS call in START-EXIT program unit – KDCS call in SHUT-EXIT program unit
1C	Application names in ROOTDATA and KAA do not match, or illegal KAA header (UTM error).
20	Application names in ROOTDATA and KTA do not match (UTM error).
24	Parameter list address not aligned with word boundary (UTM error).

nn	Meaning
28	KDCS parameter area not aligned. If the include files made available by openUTM for the KDCS parameter areas are used, this is a UTM error. If user-defined areas are used in the program unit for the KDCS parameter area, this is a user error (alignment with halfword boundary). Action: check the alignment and address of the parameter area.
2C	Field SVC return code is not preset with X'FFFFFFFF' (UTM error).
FF	SVC call could not be passed to the UTM subsystem. Cause: UTM was not defined or loaded as a subsystem.

[1] In the case of return codes of the UTM error type: Write PR and notify Systems Support.

00mmFFFF return codes:

This return code is set by either UTM or the BS2000 DSSM (see the "Executive Macros" manual, standard header). Possible values and reasons:

mm	Meaning
01	For UTM SVC an incorrect value was specified for UNIT or FUNCTION in the standard header (UTM error[1]).
03	Interface version in the standard header is not supported by the operating system. Action: Check subsystem definition.
04	Parameter list, specified with SVC call is not aligned with word boundary (UTM error).
41	UTM has not yet been loaded as a subsystem (CREATIM=ONCREA), or the load library for UTM does not contain the complete system code of UTM V6.3.
82	When a new application or a new application task is started, it is discovered that the system administrator has put the UTM subsystem into DELETE or HOLD status.

[1] In the case of return codes of the UTM error type: Write PR and notify Systems Support

K048   No additional UTM task can be created

K049 Error &RCCC2 during application startup

UTM issues message K049 whenever the start of a UTM task is aborted due to an error: The table below contains the possible &RCCC2 error codes together with error causes and possible error recovery actions.

Start error codes in message K049

Code	Error cause	Action
02	There is not enough space available for the ROOTDATA area.	Check generation and system config-uration
03	The ROOTDATA area is not allocated or the string ‚ROOTDATA ‘ does not exist.	Reconcile UTM versions
04	KDCROOT and the UTM system modules are of different versions.	Reconcile UTM versions
05	Application cannot be signed on to Name Manager.	System error; please write PR and notify Systems Support
06	The start of a continuation task is aborted due to abortion of the start of the 1st task.	See error code of 1st task
07	The user's message module is not consistent with the message module of the UTM system code (e.g. incorrect number of inserts).	Correct message module and link application again
08	An attempt has been made to start UTM tasks both in dialog and in batch mode.	Start continuation tasks in the same mode as the 1st task
09	Insufficient space for the fixed section of KAA or for the KDCA-FCB.	System error; please write PR and notify Systems Support
10	A UTM task request for 4K of class 6 memory has been rejected.	Check BS2000 generation and memory requirements of the appli-cation
11	Error at 1st OPEN call for the KDCA file of KDCFILE, possibly due to incorrect FILEBASE name specification in start parameters. Possible cause in UTM cluster applications: An online import is running for the node that is to be started.	See DMS error code
12	DMS error when reading 1st KAA page from KDCA file.	See DMS error code
13	DMS error when reading first or last check page of KDCA file; the file cannot be used.	Make copy or regenerate with KDCDEF

Code	Error cause	Action
14	The UTM version in the KAA (KDC Application Area) which is read from the KDCFILE does not match the version of the UTM system code. The error can occur if you attempt to run two UTM applications having the same name in different UTM versions in parallel.	Change name of an application (MAX APPLINAME=...)
15	KDCDEF and UTM code inconsistent with respect to KAA structure.	Regenerate with correct KDCDEF
16	KDCDEF and UTM system modules are of different versions.	Reconcile versions
17	Before the start KDCFILE was processed by KDCUPD, which terminated abnormally	Make KDCFILE consistent by means of either KDCUPD or KDCDEF
18	The KDCFILE in a follow-up task does not have the same generation time as the first task.	Start follow-up task with same KDCFILE file as first task
20	The memory pool for KAA cannot be reserved in class 5 memory by the 1st task of the UTM application due to insufficient address space.	Check BS2000 generation and UTM generation
21	A continuation task of a UTM application is unable to link up with the KAA memory pool due to insufficient address space.	As for 20
22	DMS error when reading an NSR page in KAA.	See DMS error code
23	A KAA NSR page has been destroyed, the KDCA file can no longer be used.	Regenerate with KDCDEF
24	DMS error when reading an SR page in KAA.	See DMS error code
25	A KAA SR page has been destroyed, the KDCA file can no longer be used.	Regenerate with KDCDEF
26	At least one program for a generated event function is missing or an event-driven service is missing; i.e. the program is neither linked nor can it be loaded	Transfer program to the library specified at generation and relink if necessary
27	KCSBKAA returned a bad return code. Error during initialization of the SYSLOG data in the KAA	Write problem report
28	During a (follow-up) start of the UTM application an inconsistency in the database generation between KDCFILE and ROOT was detected. The error occurs when KDCFILE and ROOT source are incompatible. UTM rejects this in order to avoid consequent errors.	Generate KDCFILE and ROOT with the same start parameters

Code	Error cause	Action
29	– The administration program unit (KDCADM) is missing or – TAC KDCSHUT is not generated.	– Add the KDCADM program to the generation and add to the UTM application – Generate TAC KDCSHUT
30	The request from a UTM task for a contiguous area for task-specific UTM tables (= KTA) in class 5 memory has been rejected.	Check BS2000 generation and memory requirements; reduce KTA by changing KDCDEF operands (e.g. check MAX TRACEREC=)
31	An attempt has been made to start more tasks than have been generated or set via administration.	User error
32	Bad return code from KCSBKTA (KCSSLOG may not have been able to initialize the SYSLOG data in the KTA).	Write problem report
33	UTM refuses to start more tasks for the application because the latter has already terminated (normal or abnormal termination).	Usual action
34	The KDCFILE in a follow-up task does not have the same KAA size as the first task.	Start follow-up task with same KDCFILE file as first task
35	While dynamically loading the application program, UTM detects that the application is being aborted.	Usual action
36	Error in Name Manager call for a continuation task of an application.	System error; please notify Systems Support
37	Number of tasks to be started insufficient for PGWT processing or for a UTM cluster application.	Start at least 2 tasks
38	The task has already signed on to the Name Manager but with invalid KAA or KTA address. This error occurs if the application start terminated with errors (xx) and the application program is subsequently restarted with the same task. UTM terminates the task abnormally due to a suspected loop in the start procedure.	Check start procedure
39	When dynamically loading the application program, UTM detected that the generation has been illegally changed.	Check start procedure and generation
40	Number of entries in the program, load module or message module tables in ROOT and in the KDCFILE do not match (see K072 message)	Generate KDCFILE and ROOT with the same start parameters

Code	Error cause	Action
41	A program or load module attribute in ROOT and in the KDCFILE do not match (see K073 message).	Generate KDCFILE and ROOT with the same start parameters
42	The task has failed to create a slot for the task-service table due to insufficient address space. The slot should be located in a memory pool in the class 5 memory.	Check BS2000 generation and memory requirements of the UTM application
43	ROOT and KDCFILE do not match with regard to the sequence of the user message modules.	Generate KDCFILE and ROOT with the same start parameters
44	The generations of KDCFILE and ROOT do not match with regard to the maximum length of the messages to/from clients. (KDCDEF statement MAX TRMSGLTH)	Generate KDCFILE and ROOT with the same start parameters
45	Error in KCSLKLC when setting a TAC class lock	Increase parameter MAX RESWAIT= (...,time2)
46	UTM cluster application: A follow-up task was started with different cluster filebase specifications from the first task.	Start first task and follow-up task with same start parameters
47	The generations of KDCFILE and ROOT do not match with respect to the use of BLS.	Generate KDCFILE and ROOT with the same start parameters
48	Error while initializing buffer management in start case 1 or 2.	Increase the virtual address space. If this does not help, write problem report
49	The version number of the phase started in a continuation task is different to that of the phase of the first task.	Supply phase of first task.
50	The 1st UTM task is trying to set up the cache memory pool, or a continuation task is trying to link up to it. This is not possible due to insufficient address space. The attempts are made for class 5 memory.	Check BS2000 generation and memory requirements of the UTM application
51	The file is not a KDCDEF file; for file name see K082 message (possible cause: KDCDEF from older version)	Supply or generate correct file
52	The file has been inadvertently replaced by another KDCFILE file, e.g. pool file instead of KDCB file; for file name see K082 message.	As for 51
53	The file does not match the KDCA file (e.g. different KDCDEF time); for file name see K082 message; for details on the error cause see the log in SYSOUT.	As for 51

Code	Error cause	Action
54	The consistency check shows that the file has been destroyed. Possibly a version mix. For file name see K083 message	As for 51 Check versions
55	DMS error with KDCA/KDCB file	See DMS error code
56	DMS error with page pool file	See DMS error code
57	DMS error with restart file	See DMS error code
58	Error in conjunction with the SYSLOG file to be made available by the user Possible causes: – Neither the link name SYSLOG nor a file generation group with the name base-name.SLOG exists. Message K043 is additionally output. – Size monitoring was generated (MAX SYSLOG-SIZE), but the SYSLOG made available is not a file generation group.	Check start procedure, SYSLOG file and generation
59	Error when opening SYSLOG file	See DMS error code
60	Error during application sign-on or continuation task to BCAM or socket. For exact error cases see BCAM info word in message K065 (default destination SYSLOG) or message K154 (default destination SYSLOG) for application name with T-PROT=SOCKET. User error or system error.	Evaluate K065 or K154 in the SYSLOG file; for contents of the BCAM word info, consult Systems Support
61	When signing on to BCAM or socket, a continu-ation application task is informed that it is the first task; this error may occur if the UTM application was previously terminated by a BCAM command (e.g. BCLOSE) and a task of that application was not able to terminate, e.g. because it was processing a long transaction or was waiting for the database to be loaded (1st task of the application).	Check if application tasks are still present in the system; load database if necessary; then restart
62	The 1st task of the application has detected inconsistent data in KDCFILE during warm start.	System error; please write PR and notify Systems Support, regenerate KDCFILE with KDCDEF
63	The number of TCB entries in the KAA is not the same as in ROOT.	Regenerate KDCFILE and/or ROOT
64	A request from the application's first task for an 8K work area in class 6 memory made during warm start has been rejected	Check BS2000 generation and UTM generation
65	As for 64 (length = task no. * 2K)	As for 64

Code	Error cause	Action
70, 72, 73	The 1st task of the application has detected inconsistent data in KDCFILE during warm start.	System error; please write PR and notify Systems Support, regenerate KDCFILE with KDCDEF
74	Database problem with DB status check during restart; error cause KU is output, see KCRCDC KUxx error codes	Correct database, restart application
75, 76	The 1st task of the application has detected inconsistent data in KDCFILE during warm start.	System error; please write PR and notify Systems Support, regenerate KDCFILE with KDCDEF
77	A user ID for an open asynchronous process can no longer be found	Write PR
79	A UTM task requests, but is not allocated class 6 memory.	Check BS2000 generation and memory requirements of the application.
80, 81	The application's first task has detected inconsistent data in KDCFILE when setting up the page pool map.	System error; please write PR and notify Systems Support, regenerate KDCFILE with KDCDEF
82	Database problem when erasing DB status information; error cause KUxx is output, see KCRCDC error codes.	Correct database, restart application
83	Database problem when resetting a DB transaction; error cause KUxx is output, see KCRCDC error codes.	Correct database, restart application
84	The task attempts to create a slot for an OSI service table. This attempt is unsuccessful due to a lack of memory space. The slot should be located in a memory pool in class 5 memory.	Check system generation and memory requirements of the application.
85	The task has failed to create a slot for an open service due to insufficient address space. The slot should be located in a memory pool in the class 5 memory.	Check system generation and memory requirements of the application.
86	The application's first task has detected unallocated page pool pages in the cache (during warm start)	System error; please write PR and notify Systems Support
89	Error during name manager call for the application's first task	System error; please write PR and notify Systems Support
90	Error when starting an open distributed asynchronous service	System error; please write PR and notify Systems Support

Code	Error cause	Action
91	Error when starting XAP-TP. A full description of the error can be found under message K124.	System error; please write PR and notify Systems Support
92	No free NODE table was found during a restart for OSI TP with XAP-TP.	System error; please write PR and notify Systems Support
93	A user ID for a transaction which has not yet terminated was found during a restart for OSI TP with XAP-TP.	System error; please write PR and notify Systems Support
94	An inconsistent NODE table was found during a restart for OSI TP with XAP-TP.	System error; please write PR and notify Systems Support
95	DMS error while writing a UTM cluster file (start case 1).	See DMS error code
96	A UTM cluster file does not correspond to the cluster configuration file, e.g. different generation time. File name, see K082 message. For details on the cause of the error see the log message in SYSOUT.	Use the correct UTM cluster file or regenerate the UTM cluster application.
97	DMS error while writing a cluster file (operation code CL_FIRST_NODE)	See DMS error code
98	The KDCFILE does not correspond to the cluster configuration file, e.g. different generation time. File name, see K082 message. For details on the cause of the error see the log message in SYSOUT.	Use the correct KDCFILE or regenerate the KDCFILE.
99	Database problem when starting via IUTMDB: Database system not available.	Check generated database systems
100	An inconsistency was detected between KDCFILE and ROOT while starting the UTM application in the KB length.	Create KDCFILE and ROOT with the same generation parameters
101	An inconsistency was detected between KDCFILE and ROOT while starting a follow-up task in the KB length.	Create KDCFILE and ROOT with the same generation parameters
102	An inconsistency was detected between KDCFILE and ROOT while restarting the application program in the KB length.	Create KDCFILE and ROOT with the same generation parameters
103	An inconsistency was detected between KDCFILE and ROOT while starting the UTM application in the NB length.	Create KDCFILE and ROOT with the same generation parameters.

Code	Error cause	Action
104	An inconsistency was detected between KDCFILE and ROOT while starting a follow-up task in the NB length.	Create KDCFILE and ROOT with the same generation parameters.
105	An inconsistency was detected between KDCFILE and ROOT while reloading the application program in the NB length.	Create KDCFILE and ROOT with the same generation parameters.
106	An inconsistency was detected between KDCFILE and ROOT while starting the UTM application regarding the use of the SQL features.	Create KDCFILE and ROOT with the same generation parameters.
107	An inconsistency was detected between KDCFILE and ROOT while starting a follow-up task regarding the use of the SQL features.	Create KDCFILE and ROOT with the same generation parameters.
108	An inconsistency was detected between KDCFILE and ROOT while reloading the application program regarding the use of the SQL features.	Create KDCFILE and ROOT with the same generation parameters.
109	An inconsistency was detected between KDCFILE and ROOT while starting the UTM application in the number of databases.	Create KDCFILE and ROOT with the same generation parameters.
110	An inconsistency was detected between KDCFILE and ROOT while starting a follow-up task in the number of databases.	Create KDCFILE and ROOT with the same generation parameters.
111	An inconsistency was detected between KDCFILE and ROOT while reloading the application program in the number of databases.	Create KDCFILE and ROOT with the same generation parameters.
112	An inconsistency was detected in the type of database while starting the UTM application (only after a KDCUPD run!)	Create KDCFILE and ROOT with the same generation parameters.
115	An inconsistency was detected between KDCFILE and ROOT while starting the UTM application in the type of database.	Create KDCFILE and ROOT with the same generation parameters.
116	An inconsistency was detected between KDCFILE and ROOT while starting a follow-up task in the type of database.	Create KDCFILE and ROOT with the same generation parameters.
117	An inconsistency was detected between KDCFILE and ROOT while reloading the application program in the type of database.	Create KDCFILE and ROOT with the same generation parameters.

Code	Error cause	Action
118	UTM cluster application: Error reading filebase name of KDCFILE from the cluster configuration file possible causes: – memory bottleneck – error accessing cluster file – invalid cluster file – own node not found For more details, see message K043 and K190.	See message K043 or K190; correct start parameters or generation if necessary.
119	UTM cluster application: The KDCFILE was generated as a UTM cluster application but no CLUSTER-FILEBASE start parameter was specified.	Correct start parameters
120	UTM cluster application: The KDCFILE was not generated as a UTM cluster application but a CLUSTER-FILEBASE start parameter was specified.	Correct start parameters
122	UTM cluster application: Error editing the cluster configuration file for initialization of the KAA	See message K043 and/or K190
123	UTM cluster application: Error registering node	See message K043 and/or K190
125	UTM cluster application: The sequence of node names in the KDCFILE is different from that in the cluster configuration file.	Correct the generation; Regenerate the KDCFILE and, if necessary, the UTM cluster files
126	UTM cluster application: The local computer belongs to no or to a different XCS cluster.	Check configuration of local computer. See message K171.
127	UTM cluster application: Error opening cluster user file on start of first process in application (start case 1)	See message K043 and/or K190
128	UTM cluster application: Error opening cluster user file on start of a follow-up process (start case 2)	See message K043 and/or K190
129	UTM cluster application: Error opening cluster user file when reloading the application program after a program replacement (start case 3)	See message K043 and/or K190
130	UTM cluster application: Error requesting start lock for serialization by KCSCONS (start case 1)	See message K043 and K190
131	UTM cluster application: Error opening adminis-tration journal files on start of a follow-up process (start case 2)	See message K043 and K190

Code	Error cause	Action
132	UTM cluster application: Error opening cluster administration journal files when reloading the application program after a program replacement (start case 3)	See message K043/K190
133	UTM cluster application: Error creating administration journal files on start of first process in application (start case 1)	See message K043/K190
134	UTM cluster application: Error writing online copy on start of first process in application (start case 1)	See message K043 and/or K190
135	UTM cluster application: Error incorporating online copy on start of first process in application (start case 1)	See message K043/K190 and/or K174
136	UTM cluster application: Error editing cluster user file on start of first process in application (start case 1)	See message K043 and/or K190
137	UTM cluster application: Calling KCCCTRL with operation code REGISTER_NODE returns an unknown return code.	System error, please write problem report and inform Service.
138	UTM cluster application: Calling KCCJCTL with operation code WRITE_JOURNAL_PI returns an incorrect return code.	System error, please write problem report and inform Service.
139	UTM cluster application: No lock was requested for start serialization.	System error, please write problem report and inform Service.
140	UTM cluster application: Error concluding registration of node at cluster	See message K043 or K190
141	UTM cluster application: Calling KCCJCTL with operation code JFCT_SET_KAA_INFO returned an incorrect return code.	System error, please write problem report and inform Service.
142	UTM cluster application: Calling KCCJCTL with operation code JFCT_GET_KAA_INFO returned an incorrect return code.	System error, please write problem report and inform Service.
143	UTM cluster application: Error opening administration journal files on start of first process in application (start case 1)	See message K043 and K190
144	UTM cluster application: Error incorporating administration journal	System error, please write problem report and inform Service.
145	UTM cluster application: Calling KCCJCTL with operation code JFCT_SET_KAA_INFO returned an incorrect return code.	System error, please write problem report and inform Service.

Code	Error cause	Action
146	UTM cluster application: The runtime configuration of the node application that is to be started does not match the runtime configuration of the running node applications.	See message K174
147	UTM cluster application: Global administration actions should be incorporated in the warm start. However, the administration journal files cannot be opened.	Check whether the administration journal files exist, see message K043/K190
148	UTM cluster application: Global administration actions should be incorporated in the warm start. Internal error when reading journal files.	System error, please write problem report and inform Service.
149	UTM cluster application: Internal error when cleaning up node-specific information in the journal files.	System error, please write problem report and inform Service.
150	UTM cluster application: Error opening administration journal files on start of first process in application (start case 1)	See message K043 and K190
151	UTM cluster application: Error requesting lock for start serialization of nodes (during warm start).	See message K043 or K190
152	UTM cluster application: Error requesting lock for start serialization of nodes	See message K043 or K190
153	UTM cluster application: Cluster page pool control file defective.	See message K190
154	UTM cluster application: Error opening the LOCK file on the start of a follow-up process (start case 2).	See message K043
155	UTM cluster application: Error opening the LOCK file on the start of the first process in the application (start case 1).	See message K043
156	UTM cluster application: Error opening the LOCK file on the start of a follow-up process (start case 3).	See message K043
157	UTM cluster application: Error while checking the cluster GSSB file	See message K043 or K190
158	UTM cluster application: Error while checking the cluster ULS file	See message K043 or K190
161	UTM cluster application: Error while releasing the ULS locks in the cluster ULS file	See message K190
162	UTM cluster application: Error while releasing the GSSB locks in the cluster GSSB file	See message K190

Code	Error cause	Action
164	UTM cluster application: Error while opening the LOCK file on warm start.	See message K043
165	UTM cluster application: On the warm start of a node application, it was not possible to lock the cluster lock file in the generated time.	Repeat the application start operation
166	UTM cluster application: Internal error when resetting a transaction in PTC state during node recovery.	System error, please write problem report and inform Service.
167	UTM cluster application: Internal error when outputting existing transactions in PTC state during node recovery.	System error, please write problem report and inform Service.
168	As for 167.	As for 167.
169	UTM cluster application: An attempt was made to perform a node recovery for a node application that terminated normally.	Check whether the incorrect node was specified in the start parameter.
170	UTM cluster application: Cluster page pool and KDCFILE generated with different BLKSIZE.	Regenerate all the UTM cluster files.
171	UTM cluster application: A node recovery was started in a dialog.	Start node recovery in batch operation.

In the event of errors, a UTM dump is generated with REASON=STRTnn, plus a CDUMP if useful for diagnostics.

K050 Successful warm start for application &APPL under UTM &VERS / &OST1 / &BMD1

K051 Successful cold start for application &APPL under UTM &VERS / &OST1 / &BMD1

These messages K050 or K051 are output after the successful startup of the application. &VERS contains the UTM version, &OST1 the type of operating system and &BMD1 the bit mode (32/64).
Both messages have the additional inserts &AMOD, &TERM, &ATYP and &FNOD which are not contained in the default message text; these have the following meanings:

– &AMOD contains the application mode of the application:

"S" in the case of UTM-S

"F" in the case of UTM-F

– &TERM contains the termination type entered in the KDCFILE at application startup; the following values are possible:

"C": The KDCFILE was created anew with KDCDEF.

"U": The KDCFILE was updated with KDCUPD.

"N": The last application run was terminated normally.

"A": The last application run was terminated abnormally.

&ATYP contains:

"C" in the case of a UTM cluster application
"S" in the case of a standalone application

&FNOD contains:

" " (blank) in the case of a standalone application
"Y" on the startup of the first node application in a UTM cluster application
"N" on the startup of each subsequent node application in a UTM cluster application

K052 Startup completed - task &TASK activated for application &APPL, System-Task:
&STSK

&STSK specifies whether (Y) or not (N) the process is a system process.

K053 New user log file created; old user log file contains &CNTR records.

K054 Copies of the user log files are not identical.

K055 Asynchronous service &ATAC1 terminated by UTM; KCRCCC=&RCCC;
KCRCDC=&RCDC; USER=&USER; LTERM=&LTRM

K056 Task &TASK terminated

K057 Application run terminated

K058 Abnormal termination of task &TASK

K059 Abnormal termination of application run

K060   Application run aborted; reason = &TRMA

UTM dump error codes

UTM creates a memory dump whenever a UTM application is aborted or a dump requested. Such a dump is produced for each task of the application (see chapter "The UTM dump" on page 53ff).

The error code is indicated in a UTM dump with REASON= in the second line on each page of the edited UTM dump.

If a UTM application is terminated abnormally, openUTM also performs the following actions before restarting insofar as this is possible:

– immediately terminate all transactions currently being processed by the individual processors of the UTM application
– clear all connections to the communication partners of the UTM application
– clear connections to external resource managers (e.g. to database systems)
– close all files
– terminate all processes

For security reasons, when a UTM application terminates abnormally, no attempt is made to restore the KDCFILE, which contains all the data required for running the UTM application, to a consistent status. This is only done on restarting.

The **Group** column in the tables below describes to which reason group the dump code error code belongs. The following groups exist:

A   The cause is a user error, e.g. an error in

   – generating and administering UTM applications
   – generating the system (e.g. division of the address space)
   – programming a format exit

U   The cause is an error in the UTM code.

S   The cause is an error in another system component (software or hardware).

F   The dump is a continuation dump, another task has caused the application to terminate abnormally.

D   The UTM dump was created for diagnostic purposes. The UTM work process continues running.

M   The cause is a memory bottleneck.

X   The cause is an error in the XAP-TP code (OSI TP).

Multiple classifications are possible, e.g. ADS.

You must write a problem report for error diagnosis for all errors of groups U, X and S and all error codes **not** listed in the table below. A number of different documents are required for diagnosis. For detailed information and a list of the required documents, refer to .

Code	Group	Reason
AHQA00	MX	XAP-TP component. KCOCOTA module, QueueAnno() function. The mGetBufferSpace() macro issued the return code LB_NOMEM.
AHSA02	SUX	XAP-TP component. Module KCOCOTA, function SendAnno(). The system interfaces call KCOBRSE supplied a bad return code.
AINF01	S	The return code ist not equal X'00' after the AINFcall.
ALME01	ASU	Illegal system return code during KCSALME call. If this error occurs in a tool, it generally means that the virtual storage is insufficient for the request to be processed. Action: Extend the address space for the ID. For KDCDUMP an estimation of uses virtual storage is given in the chapter „KDCDUMP messages".
ALME02	ASU	UTM tried to create a memory pool for KAA tables, but the memory pool already exists. Possible cause: a task of the 'old' UTM application has not completed program/task termination. Action: check whether all tasks of the 'old' application have left the system. A task may, for instance, still be present in the system because a console message has not been answered. Then restart.
AREC01	SU	Error during $AREC call for the writing of a record by the BS2000 Accounting Routine
ASA006	SUX	XAP-TP component. Module KCOASAM was called with the operation code ASAM_ATTACH. Bad return code from bBuildPAddr.
ASA007	SUX	XAP-TP component. Module KCOASAM was called with the operation code ASAM_ATTACH. First task of the application, but OSS issued return code NOTFIRST.
ASA009	SX	XAP-TP component. Module KCOASAM was called with the operation code ASAM_ATTACH. aufgerufen. OSS returned the return code "INVEREF" for an „attach" call.
ASA010	SX	XAP-TP component. Module KCOASAM was called with the operation code ASAM_ATTACH. OSS returned a bad return code for an „attach" call.

Code	Group	Reason
ASA011	FSX	XAP-TP module The module "KCOASAM" was called with the opcode "ASAM_DETACH". On the "detach" call, OSS returned the return code "ERROR". The task is already in the termination phase after another task has abnormally terminated the application. OSS returned an incorrect error code because openUTM still has to fetch an event from OSS.
ASA012	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_DETACH". On the "detach" call, OSS returned the return code "INVAREF".
ASA013	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_DETACH".On the "detach" call, OSS returned a bad return code
ASA033	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". the returned application context name is too long.
ASA034	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returns the return code "ERROR" on the "assin" call.
ASA035	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returns the return code "INVREF" on the "assin" call.
ASA036	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returns a bad return code on the "assin" call.
ASA043	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returns the return code "ERROR" on a positive "assrs" call.
ASA044	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". Bad return code from "PutElement".
ASA045	SX	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returns a bad return code on a positive "assrs" call.
ASA046	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returned the return code "ERROR" on the negative "assrs" call.
ASA048	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". OSS returned a bad return code on the negative "assrs" call.

Code	Group	Reason
ASA049	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_IND". Bad return code from "PutElement".
ASA051	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". The retruned Application Context Name is too long.
ASA052	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned the return code "ERROR" on the "asscf" call.
ASA053	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned the return code "INVREF" on the "asscf" call.
ASA054	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned a bad return code for an "asscf" call.
ASA060	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned the return code "ERROR" on the "aborq" call.
ASA062	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned a bad return code for an "aborq" call.
ASA064	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned an invalid diagnostic value on a negative association confirmation.
ASA065	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". OSS returned an invalid "result source" value on a negative association confirmation.
ASA066	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF_TIMEOUT". OSS returned the return code "ERROR" on the "aborq" call.
ASA068	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF_TIMEOUT". OSS returned a bad return code on the "aborq" call.
ASA071	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_IND". OSS returned the return code "ERROR" on the "aboin" call.

Code	Group	Reason
ASA072	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_IND". OSS returned the return code "INVREF" on the "aboin" call.
ASA073	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_IND". OSS returned a bad return code on the "aboin" call.
ASA080	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_P_ABORT_IND". OSS returned the return code "ERROR" on the "paboin" call.
ASA081	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_P_ABORT_IND". OSS returned the return code "INVREF" on the "paboin" call.
ASA082	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_P_ABORT_IND". OSS returned a bad return code on the "paboin" call.
ASA083	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". Bad return code from "PutElement".
ASA084	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF". Bad return code from "PutElement".
ASA085	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ASS_CNF_TIMEOUT". Bad return code fromn "PutElement".
ASA088	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned the return code "ERROR" on the "relin" call.
ASA089	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned the return code "INVREF" on the "relin" call.
ASA090	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned a bad return code on the "relin" call.
ASA091	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned an invalid value for "release reason".

Code	Group	Reason
ASA092	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned the return code "ERROR" on the "aborq" call.
ASA094	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned a bad return code on the "aborq" call.
ASA095	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned the return code "ERROR" on the "relrs" call.
ASA097	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_RELEASE_IND". OSS returned a bad return code on the "relrs" call.
ASA099	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_REQ". OSS returned an invalid value for "abort diagnostic".
ASA100	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_REQ". OSS returned the return code "ERROR" on the "aborq" call.
ASA101	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_IND" T Bad return code from "PutElement".
ASA102	SX	XAP-TP component. The module "KCOASAM" was called with the opcode "ASAM_ABORT_REQ". OSS returned a bad return code on the "aborq" call.
ASA104	AX	XAP-TP component. The function "bBuildPAddr" of the module KCOASAM was called. The presentation selector of a local access point or of a remote partner is too long.
ASA105	AX	XAP-TP component. The function "bBuildPAddr" of the module KCOASAM was called. The session selector of a local access point or of a remote partner is too long.
ASA116	SX	XAP-TP component. The function "BuildRemoteAet" of the module KCOASAM was called. The APT has "form2" format, but the AEQ does't have "form2" format.
ASA120	SX	XAP-TP component. The function "BuildRemoteAet" of the module KCOASAM was called. the APT has the "form1" format, but the AEQ does't have "form1" format.
ASA122	SX	XAP-TP component. The function "BuildRemoteAet" of the module KCOASAM was called. The APT has neither the "form1" format nor the "form2" format.

Code	Group	Reason
ASA128	M	XAP-TP component. The „CopyDefinedContext" function of the module KCOASAM was called. Bad return code of „PutElement".
ASA137	M	XAP-TP component The module "KCOASAM" was called with the opcode "ASAM_GO_IND". The association is not locked.
ASA139	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_P_ABORT_IND. Bad return code from "PutElement".
ASA151	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_RELEASE_IND. Bad return code from "PutElement".
ASA152	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_ABORT_REQ. Bad return code from "PutElement".
ASA153	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_ABORT_REQ. Bad return code from "SacfSeparator".
ASA155	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_ABORT_IND. Bad return code from "SacfSeparator".
ASA156	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_P_ABORT_IND. Bad return code from "SacfSeparator".
ASA157	SX	XAP-TP component. KCOASAM was called with the operation code ASAM_RELEASE_IND. Bad return code from "SacfSeparator".
ASIO01	AU	File is not identified as open in FCB during KCSASIO call. Reason may also be a CLOSE ALL in the program unit run.
ASIS99	D	Normal execution of command "KDCSHUT KILL" or the call to the program interface for administration with the opcode KC_SHUTDOWN and the sub-opcode1 KC-KILL.
BFMM05	M	UTM cluster application It was not possible to request another buffer segment. The virtual address space may be too small.
BFMM21	ASU	UTM cluster application A timeout occurred while releasing a file lock. Action see CCFG19.
BFMM22	SU	UTM cluster application A bad return code was returned when releasing a file lock.

Code	Group	Reason
BRSE01	S	Invalid parameter QUEUE_DISCIPLINE when creating a bourse or when sending or receiving an announcement. Invalid parameter LENGTH_OF_ANNO when sending an announcement. Bad return code form the bourse interface.
BRSE02	S	Error on enqueuing of the task at WORK bourse (chamber 1).
BRSE03	S	Error on enqueuing of the task at task specific bourse (chamber 1).
CACHT1	F	Another task has terminated the application abnormally (=continuation dump)
CACHT2	F	(See CAHCT1) Another task has terminated the application abnormally (=continuation dump)
CACHT3	F	(See CAHCT1) Another task has terminated the application abnormally (=continuation dump)
CACHT4	F	(See CAHCT1) Another task has terminated the application abnormally (=continuation dump)
CACHT5	F	(See CAHCT1) Another task has terminated the application abnormally (=continuation dump)
CC-...	D	Diagnostic dump generated on the basis of a primary KDCS return code. The prefix CC- is followed by the primary KDCS return code (e.g. CC-84Z). Activation and deactivation via the message dump function.
CCFG07	SU	UTM cluster application Module KCCCFG, entry KCCCGFB Bad return code from KCSGLHN.
CCFG19	ASU	UTM cluster application Module KCCCFG, opcode CCFG_READ_CLUSTER_FILE Timeout when requesting shared lock. Action see section „Actions when locking UTM cluster files" below this table.
CCFG20	SU	UTM cluster application Module KCCCFG, opcode CCFG_READ_CLUSTER_FILE Bad return code when requesting shared lock.
CCFG21	ASU	UTM cluster application Module KCCCFG, Timeout when requesting exclusive lock. Action see CCFG19.
CCFG22	SU	UTM cluster application Module KCCCFG Bad return code when requesting the exclusive lock.
CCFG29	ASU	UTM cluster application Module KCCCFG, timeout when releasing file lock. Action see CCFG19.

Code	Group	Reason
CCFG30	SU	UTM cluster application Module KCCCFG, opcode CCFG_READ_CLUSTER_FILE Bad return code when releasing the file lock.
CCFG32	ASU	UTM cluster application Module KCCCFG, opcode CCFG_REGISTER_COMPLETE Invalid version of cluster configuration file.
CCFG33	ASU	UTM cluster application Module KCCCFG, opcode CCFG_REGISTER_COMPLETE Corrupt cluster configuration file.
CCFG34	ASU	UTM cluster application Module KCCCFG, opcode CCFG_REGISTER_COMPLETE Invalid application name in cluster configuration file.
CCFG35	ASU	UTM cluster application Module KCCCFG, opcode CCFG_REGISTER_COMPLETE The cluster configuration file was regenerated during operation.
CCFG41	ASU	UTM cluster application Module KCCCFG, opcode CCFG_OPEN_CLUSTER_FILE Timeout when initializing global lock. Action see CCFG19.
CCFG42	SU	UTM cluster application Module KCCCFG, opcode CCFG_OPEN_CLUSTER_FILE Bad return code when initializing the global lock.
CCFG43	ASU	UTM cluster application Module KCCCFG, opcode CCFG_CLOSE_CLUSTER_FILE Timeout when destroying global lock. Action see CCFG19.
CCFG44	SU	UTM cluster application Module KCCCFG, opcode CCFG_CLOSE_CLUSTER_FILE Bad return code when destroying the global lock.
CCFG45	SU	UTM cluster application Module KCCCFG, opcode CCFG_INIT_KAA Bad return code from KCSGLHN
CCFG49	S	UTM cluster application Module KCCCFG, opcode READ_CLUSTER_FILE Bad return code when reading the cluster file.
CCFG50	A	UTM cluster application Module KCCCFG, opcode READ_CLUSTER_FILE Corrupt cluster file.
CCFG51	A	UTM cluster application Module KCCCFG, opcode READ_CLUSTER_FILE Invalid version of cluster configuration file.

Code	Group	Reason
CCFG52	A	UTM cluster application Module KCCCFG, opcode READ_CLUSTER_FILE Invalid application name in cluster configuration file.
CCFG53	A	UTM cluster application Module KCCCFG, opcode READ_CLUSTER_FILE The cluster configuration file was regenerated during operation.
CCFG55	A	UTM cluster application Module KCCCFG, opcode NODE_FAILURE Corrupt cluster configuration file.
CCFG56	A	UTM cluster application Module KCCCFG, opcode NODE_FAILURE Invalid version of cluster configuration file.
CCFG57	A	UTM cluster application Module KCCCFG, opcode NODE_FAILURE Invalid application name in cluster configuration file.
CCFG58	A	UTM cluster application Module KCCCFG, opcode NODE_FAILURE The cluster configuration file was regenerated during operation.
CCFG62	ASU	UTM cluster application Module KCCCFG, opcode CCFG_RESET_START_SERIALIZATION Timeout when releasing lock. Action see CCFG19.
CCFG63	SU	UTM cluster application Module KCCCFG, opcode CCFG_RESET_START_SERIALIZATION Bad return code when releasing lock.
CCFG64	ASU	UTM cluster application Module KCCCFG, opcode CCFG_RESET_START_SERIALIZATION Timeout when destroying lock. Action see CCFG19.
CCFG65	SU	UTM cluster application Module KCCCFG, opcode CCFG_RESET_START_SERIALIZATION Bad return code when destroying lock.
CCFG71	AU	UTM cluster application KCCCFG module, opcode READ_CLUSTER_FILE or CHECK_CLUSTER_FILE The local node is marked as failed in the cluster configuration file.
CCFG72	AU	UTM cluster application KCCCFG module, opcode CCFG_NODE_FAILURE The local node is marked as failed in the cluster configuration file.

Code	Group	Reason
CCKF02	SU	UTM cluster application Module KCCCKF Unexpected return code from KCCFILA when opening the KDCFILE.
CCKF04	SU	UTM cluster application Module KCCCKF Unexpected return code from KCCFILA when closing the KDCFILE.
CDTN02	M	XAP-TP component. Module KCOCOHF, function CheckDtnidTtnid(). The macro mGetBufferSpace() issued the return codeLB_NOREM.
CFMM05	D	On send, the length in DCF does not match that specified in the letter header.
CLREST	D	UTM cluster application The dump is only generated during a node application warm start with test mode enabled. It is used for diagnostic purposes following possible errors during the warm start of the cluster.
CONS03	A	The KDCFILE was overwritten during operation. Possible cause: KDCDEF run in current application. Action: Check start procedure
CSND04	MX	XAP-TP component. Invalid return code after calling PutElement() to request a dynamic buffer for concatenator send data.
CSND05	SX	XAP-TP component. Invalid return code after calling an OSS presentation function.
CTPF04	M	XAP-TP component: The return code of PutElement was not equal to DM_OK.
DC....	D	Diagnostic dump generated on the basis of a secondary KDCS return code. The prefix DC- is followed by the secondary KDCS return code (e.g. DCKS17). Activation and deactivation via the message dump function.
DIAGCL	D	UTM cluster application A node application has been terminated abnormally. Before termination, all the other node applications were informed of the abnormal termination if this was still possible. The informed node applications write a diagnostic dump.
DIAGDP	D	A diagnostic dump has been generated by means of the administration command "KDCDIAG DUMP=YES" or by calling the program interface for administration with the opcode KC_CREATE_DUMP.
DMCA00	M	XAP-TP component. Module KCOCODM, function ConnectDynMemArea(). The function ConnectSharedMem() issued the return code MEM_NOMEM.
EDITER	ADS	Error in conjunction with the edit profile used. The dump only contains KDCROOT and KTA data.

Code	Group	Reason
EHHP00	M	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The return code from mGetBufferSpace() was not equal to LB_OK.
EHHP01	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_datain() returned the return code P_ERROR.function
EHHP02	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_datain() returned the return code P_INVREF.
EHHP03	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_datain() returned an unknown return code
EHHP04	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_typein() returned the return code P_ERROR.
EHHP05	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_typein() returned the return code P_INVREF.
EHHP06	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_typein() returned an unknown return code .
EHHP07	SX	XAP-TP function. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns 0 for length of user data after calling the OSS function p_typein().
EHHP08	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_synin() returned the return code P_ERROR.
EHHP09	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_synin() returned the return code P_INVREF.
EHHP10	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_synin() returned an unknown return code.
EHHP12	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns zero for length of user data after calling the OSS function p_synin().
EHHP13	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_syncf() returned the return code P_ERROR.

Code	Group	Reason
EHHP14	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_syncf() returned the return code P_INVREF.
EHHP15	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_syncf() returned an unknown return code.
EHHP17	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns zero for length of user data after calling the OSS function p_syncf().
EHHP18	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkgin() returned the return code P_ERROR.
EHHP19	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkgin() returned the return code P_INVREF.
EHHP20	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkgin() returned an unknown return code.
EHHP21	S	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkgin() did not return the token S_T_MINOR.
EHHP22	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkpin() returned the return code P_ERROR.
EHHP23	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkpin() returned the return code P_INVREF.
EHHP24	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkpin() returned an unknown return code.
EHHP25	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_tkpin() did not return the token S_T_MINOR.
EHHP26	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns zero for length of user data after calling the OSS function p_tkpin().
EHHP27	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_minin() returned the return code P_ERROR.

Code	Group	Reason
EHHP28	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_minin() returned the return code P_INVREF.
EHHP29	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_minin() returned an unknown return code.
EHHP30	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns zero for length of user data after calling the OSS function p_minin().
EHHP31	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_mincf() returned the return code P_ERROR.
EHHP32	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_mincf() returned the return code P_INVREF.
EHHP33	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function p_mincf() returned an unknown return code.
EHHP34	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function CalcUserDataLth() returns zero for length of user data after calling the OSS function p_mincf().
EHHP35	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function o_event() returned the unexpected event P_MAJIN.
EHHP36	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The function o_event() returned the unexpected event P_MAJCF.
EHHP37	SX	XAP-TP component. Module KCOXFEH, function HandlePresEvent(). The OSS function o_event() returned an unexpected event type.
EHRP01	MX	XAP-TP component. Module KCOXFEH, function ReloadPresEvent(). The macro mGetBufferSpace() issued a return code not equal to LB_OK.
EHSP00	SX	XAP-TP component. Module KCOXFEH, function StorePresEvent(). StorePresEvent() is to store a presentation event in DynMem. However, an event is already stored for the corresponding association (mValLth(g, &pAss->PendingEvnt.h) > 0).

Code	Group	Reason
EHSP01	M	XAP-TP component. Module KCOXFEH, function StorePresEvent(). The function PutElement() issued the return code DM_NOMEM. Response: Increase the value for the size of the OSI scratch area in the KDCDEF generation (parameter MAX OSI-SCRATCH-AREA).
EKAA11	S	UTM cluster application Timeout when releasing the file lock for the JKAA file. Action see CCFG19.
EKAA12	S	Incorrect return code from KCCGFLK when releasing the file lock for the JKAA file.
EKAA13	S	UTM cluster application Timeout when destroying the file lock for the JKAA file. Action see CCFG19.
EKAA14	S	Incorrect return code from KCCGFLK when destroying the file lock for the JKAA file.
ENCERR	DSU	UTM expected an encrypted message, but received an unencrypted message. There is no application abort pending. The dump is only for diagnostic purposes.
ENDE01	SU	The KAA pointer in KTA doesn't point to a KAA.
ENDE02	AU	Interrupt with locked KAA. Possible cause: CANCEL KILL by operator.
ENDE03	ASU	Abnormal task termination between writing a confirmatory record and the (temporary) end of the UTM transaction. Possible cause: Shutdown of system with running UTM application or RUNOUT-STXIT or CANCEL during this period.
ENDE04	SU	The cache cannot be released.
ENDE05	SU	The KAA cannot be released.
ENDE06	ASU	Error when opening the SYSLOG file. This error may occur after abnormal termination of a UTM task (e.g. by CANCEL) if the user accesses the SYSLOG file before the application has terminated. It causes 1 PAM page (at most) with SYSLOG messages to be lost. Action: Check start procedure (access to SYSLOG)
ENDE07	AS	A task which was the only participant at the time in a SLOT memory pool was terminated abnormally (e.g. CANCEL-JOB, TERM macro). The memory pool is lost when the task is terminated and the application can no longer run. Action: Avoid abnormal termination of UTM tasks.
ENDE10	ASU	A task terminated abnormally and already included the start phase (system dump, FORCE-JOB-CANCEL etc.). The application is then also ended abnormally.

Code	Group	Reason
ENDE11	DSU	The application was terminated normally, during sign-off of the OSS access point, OSS delivered a return code other than OK. If test mode is activated, a UTM diagnostic dump is initiated.
ENDE12	ASU	A task terminated (either normally or abnormally). Not enough tasks remain to continue processing the application (e.g. because there are tasks in the PGWT).
ENDE14	F	A task terminated (either normally or abnormally) and the task has still not called KCSTRMA. The status of the application is, however, TERM_APPL. KCSTRMA was called with ENDE14 to ensure that all tasks generate a UTM dump if the application terminates abnormally.
ENDE15	ADU	A task has terminated abnormally in TPR (system dump, CANCEL KILL etc.). Because this is a follow-up task of the application during the start phase, the entire application is not terminated abnormally. Instead, only the diagnostic dump is written.
ENDE16	AD	Abnormal end-of-application due to abnormal termination of task in TU (TERM-STXIT, CANCEL, LOGOFF, etc.). On every abnormal task termination in TU, a diagnostic dump with reason ENDE16 is written if test mode is active.
ENDE19	ASU	Only system tasks are running for the application. All normal tasks were termi-nalet. wurden alle beendet. The reason may be an error in the start procedure or a start error when restarting the tasks after PEND ER or program exchange.
ENDPET	A	An UTM-D application cannot be terminated normally, because services still exist with the transaction status PTC (prepare to commit) or because no acknowledgments have been received for asynchronous messages sent to other applications. In this case no UTM dump is generated. openUTM carries a warm start at the next application start.
EVGE00	M	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The return code of the macro mGetBufferSpace() was not equal to LB_OK.
EVGE01	S	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The return parameter <o_echain> contained an invalid value after calling the OSS function o_event().
EVGE03	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_ACS and the user data were not yet received completely on the call.
EVGE05	ASX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_ERROR. This can also happen when „BCAM" is terminated („BCEND").

Code	Group	Reason
EVGE06	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_INVEREF.
EVGE07	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_TIMEINT.
EVGE08	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_WAKEINT.
EVGE09	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_SYSTEM.
EVGE10	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_LOOK.
EVGE11	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_TRANSPORT.
EVGE12	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned the return code O_SESSION.
EVGE13	SX	XAP-TP component. Module KCOXFEV, function GetOssEvent(). The OSS function o_event() returned an unknown return code.
EVNT03	ASX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_ERROR. This can also occur when BCAM is terminated (BCEND). The error also occurs if a UTM application with OSI TP has been started and terminated again in the dialog with two tasks and another UTM application with the same ACCESS- POINT is then started in the dialog as a follow-up task in one of the two tasks.
EVNT04	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_INVEREF.
EVNT05	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_TIMEINT.

Code	Group	Reason
EVNT06	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_WAKEINT.
EVNT07	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_SYSTEM.
EVNT08	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_TRANSPORT.
EVNT09	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_SESSION.
EVNT10	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_PRESENTATION.
EVNT11	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned the return code O_ACSE.
EVNT12	SX	XAP-TP component. Module KCOXFEV, function aputm_event(). The OSS function o_event(O_EVALLOOK) returned an unknown return code.
EXPI32	ASU	UTM cluster application Bad return code from KCCGFLK when locking the cluster GSSB file. Action see CCFG19.
EXPI34	ASU	UTM cluster application Bad return code from KCCGFLK when locking the cluster ULS file. Action see CCFG19.
EXPI35	ASU	UTM cluster application Bad return code from KCCGFLK when locking the cluster ULS file. Action see CCFG19.
EXPI36	ASU	UTM cluster application Bad return code from KCCGFLK when locking the cluster GSSB file. Action see CCFG19.

Code	Group	Reason
EXPI51	A	A periodic write was called by a task because the conf. area of the task has become full. The task wishes to write another conf. page but no pages have become free through periodic write. The cause may be a conf. area which was generated too small, with the result that restart information of a single trans-action is longer than the conf. area (e.g. transaction with access to "numerous" page pool pages). Action: Increase the number parameter in the generation statement "MAX RECBUF=(number,...).
EXPI95	A	See EXPI51, but occurs in RESET_TA_RTN.
FMMM10	A	An input message cannot be stored because the page pool is full. Action: Increase page pool by increasing MAX PGPOOL=(number,...) in KDCDEF generation.
FMSM05	DSU	KCDFMSM was called with opcode PRSP_BID even though the session status is not WAIT_OF_BID_RSP.
FMSM12	S	Illegal LSES_WORK_STATE when receiving RTR.
FMTERR	ADS	The formatting routine (FHS or user exit) reports an error to the calling UTM module (KCSWAIT, KCSISLP). The application is not aborted. The dump serves to diagnose the formatting error. It only contains KDCROOT data and is output in test mode only.
FREE01	AX	XAP-TP component. Module KCOXFFO, function ap_free(). More than APFREE_MAX_TO_REL storage areas are to be released.
GFLK04	ASU	Unexpected return code from KCSDLM on initialization of lock. Possible cause: The MSCF subsystem has been terminated.
GFLK22	ASU	Unexpected return code from KCSDLM. Possible cause: The MSCF subsystem has been terminated.
GFLK31	SU	Bad return code in function KCSGLHN (Get Local Host Name).
GFLK33	ASU	CheckDLMinfo routine: Unexpected return code from KCSDLM. Possible cause: The MSCF has been terminated.
GFLK42	ASU	The Distributed Lock Manager has repeatedly returned the return code RETRY LATER. The cause may be, for example, that a subsystem has not been started. See also last K190 message.
GFLK43	ASU	As for GFLK42.
GFLKT1	F	UTM cluster application Lock hierarchy infringed after Term Application (continuation dump).
GFLKT2	F	UTM cluster application As for GFLKT1 (continuation dump).

Code	Group	Reason
GMDT11	SU	UTM cluster application Module KCCGMDT, opcode UPDATE_AND_UNLOCK_GSSB. unexpected return code from KCCGFLK (lock GSSB file). Action see CCFG19.
INPERR	AD	Error in INPUT exit was detected.
IODxxx	ASU	ASIO return code: DMS error occurred and could not be corrected, Dxxx=DMS error code
ISLP00	A	VTSU subsystem is not loaded.
ISLP01	S	The maximum SCB langth ist larger than expected by UTM. Possible cause:New VTSU subsystem version is loaded which uses larger SCB lengths.
ISLP02	ASU	VTSU error when initializing a dynamic SCB. Possible cause: VTSU subsystem has not yet been, or is no longer, loaded.
ISLP03	ASU	VTSU error when analyzing the first 8 characters of the message. Possible cause: VTSU does not recognize function key.
ISLP04	SU	VTSU error when analyzing the UTM command.
ISLP05	ASU	VTSU error when editing a dialog input message. Possible cause: As for ISLP03.
ISLP06	SU	VTSU detects a faulty output message.
ISLP09	AU	Task-specific buffer for restart information too small. Action: Increase MAX RECBUF=(...,length) specification in KDCDEF generation.
ISLP11	SU	Protocol error.
ISLP12	AU	Task-specific buffer for restart information too small. Action: Increase MAX RECBUF=(...,length) specification in KDCDEF generation
ISLP13	AU	The value specified in the statement MAX TRMSGLTH=<length> at application generation time is too small for even a message fragment to be stored in it during message fragmentation by VTSU.
ISLP42	ASU	Error when calling VTSU to query the length of an SCB (control block for VTSU) when starting a UTM task. Possible reasons: - A data display terminal type is not supported by VTSU. - VTSU could not be dynamically loaded. Actions: Check VTSU library.
ISLPT1	F	Another task has terminated the application abnormally (=continuation dump)
ISLPT4	F	See ISLPT1

Code	Group	Reason
JFC011	ASU	UTM cluster application Timeout on KCCGFLK call in KCCJFCT with opcode CLOSE_FILES. Action see CCFG19.
JFC012	ASU	UTM cluster application Error on KCCGFLK call in KCCJFCT with opcode CLOSE_FILES. Action see CCFG19.
JFC016	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode CHECK_UNPROC_ENTRIES.
JFC018	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode CHECK_UNPROC_ENTRIES.
JFC055	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode SET_NODE_PROCESSING_STATE.
JFC058	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode SET_NODE_PROCESSING_STATE.
JFC067	SU	UTM cluster application Error when reading the administration page in KCCJFCT with opcode SET_NODE_CR_SEQNR.
JFC071	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode GET_NODE_CR_SEQNR.
JFC072	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode GET_NODE_CR_SEQNR.
JFC076	SU	UTM cluster application Error on the readControlPage call in KCCJFCT with opcode RESET_NODE_CR_SEQNR.
JFC082	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode RESET_DYNADM_LOCK.
JFC083	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode RESET_DYNADM_LOCK.
JFC400	ASU	UTM cluster application Timeout on KCCGFLK call (lock) in KCCJFCT, internal function setFileLock. Action see CCFG19.

Code	Group	Reason
JFC401	ASU	UTM cluster application Error on KCCGFLK call (lock) in KCCJFCT, internal function setFileLock. Action see CCFG19.
JFC402	SU	UTM cluster application Error on KCCGFLK call (lock) in KCCJFCT, internal function setFileLock.
JFC404	SU	UTM cluster application Timeout on KCCGFLK call (unlock) in KCCJFCT, internal function release-FileLock.
JFC405	ASU	UTM cluster application Error on KCCGFLK call (unlock) in KCCJFCT, internal function release-FileLock. Action see CCFG19.
JFC420	SU	UTM cluster application Error on KCCFILA call (write), first administration page in the journal file in KCCJFCT, internal function initControlPages.
JFC421	SU	UTM cluster application Error on KCCFILA call (write) in KCCJFCT, internal function initControlPages.
JFC430	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT, internal function switchFiles.
JFC431	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT, internal function switchFiles.
JFC460	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT, internal function checkControlPages.
JFC501	SU	UTM cluster application Error on KCCFILA call in KCCJFCT, internal function readPages.
JFC507	SU	UTM cluster application Error on KCCFILA call in KCCJFCT, internal function writePages.
JFC512	SU	UTM cluster application Error on KCCFILA call in KCCJFCT, internal function writeControlPage.
JFC550	SU	UTM cluster application Error when reading an administration page in KCCJFCT with opcode GET_JOURNAL_FILE_INFO.
JFC561	SU	UTM cluster application Error on readControlPage call in KCCJFCT with opcode SET_COPY_STATE.
JFC570	SU	UTM cluster application Error on readControlPage call in KCCJFCT with opcode SET_KAA_INFO.

Code	Group	Reason
JFC572	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode SET_KAA_INFO.
JFC580	SU	UTM cluster application Error on readControlPage call in KCCJFCT with opcode GET_KAA_INFO.
JFC581	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode GET_KAA_INFO.
JFC583	SU	UTM cluster application Error on the 3rd readControlPage call in KCCJFCT with opcode GET_KAA_INFO.
JFC584	SU	UTM cluster application Error on the 4th readControlPage call in KCCJFCT with opcode GET_KAA_INFO.
JFC590	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode SET_GLOBAL_ADM_LOCK.
JFC591	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode SET_GLOBAL_ADM_LOCK.
JFC594	SU	UTM cluster application Error on the 3rd readControlPage call in KCCJFCT with opcode GET_GLOBAL_ADM_LOCK.
JFC595	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode RESET_GLOBAL_ADM_LOCK.
JFC596	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode RESET_GLOBAL_ADM_LOCK.
JFC598	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT with opcode GET_GLOBAL_ADM_LOCK.
JFC599	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT with opcode GET_GLOBAL_ADM_LOCK.
JFC611	SU	UTM cluster application Error on the 2nd readControlPage call in KCCJFCT, routine checkAllPro-cessed.
JFC613	SU	UTM cluster application Error on the first readControlPage call in KCCJFCT, routine checkAllPro-cessed.

Code	Group	Reason
JFC626	SU	UTM cluster application Error on readControlPage call in KCCJFCT, routine checkAndRepair.
LATCT1	F	During a request for a lock, the status of an application is set to TERM APPLICATION. The task is also terminated (= continuation dump).
LCACT1	F	During a request for a lock, the status of an application is set to TERM APPLICATION. The task is also terminated (= continuation dump).
LKAAT1	F	During a request for a lock, the status of an application is set to TERM APPLICATION. The task is also terminated (= continuation dump).
LKLC26	FU	During locking of an entry, a situation is detected where the service is already entered in a queue chaining facility. System error or continuation dump if the application is currently being abnormally terminated (through error or KDCSHUT KILL). The case of a continuation dump may be recognized from the fact that the reason for the abort in message K060 is not LKLC26, and that a further dump has already been generated.
LKLC42	AU	Bourse wait time elapsed. Response: Change KDCDEF generation, increase the RESWAIT= (... , time2) parameter in the MAX statement. This may also be caused by a hardware error that results in an operator message that needs to be answered but the answer was not sent or it was sent too late. Check the console protocol in this case.
LKLC51	AU	See LKLC42
LKLC64	AU	See LKLC42
LKLCT1	F	Another task has terminated the application abnormally (=continuation dump)
LKLCT2	F	See LKLCT1
LKLCT3	F	See LKLCT1
LKLCT4	F	See LKLCT1
LKMTC0	F	Another task has abnormally terminated the application (= continuation dump before KCSBRSE call)
LKMTC1	F	Another task has abnormally terminated the application (=continuation dump after KCSBRSE call)
LKMTKP	D	This diagnostic dump is generated if TESTMODE=ON provided that the PEND KP flag is not reset in the lock field when the unlock is performed.
LPCMT1	F	During a request for a lock, the status of an application is set to TERM APPLICATION. The task is also terminated (= continuation dump).
LWRT02	SU	Error when opening the user log file <filebase>.USLA (copy A).
LWRT03	SU	Error when opening the user log file <filebase>.USLB (copy B).

Code	Group	Reason
LWRT04	ASU	Error on SETL call for user log file <filebase>.USLA. Write manipulations may have been performed to the current file generation or the base may have been modified. Use the KDCLOG command to switch while the application is running. Action if the next UTM application start is also aborted with LWRT04: Delete the user log file (i.e.the entire FGG) and regenerate it.
LWRT05	ASU	Error on SETL call for user log file <filebase>.USLB The reason may be as described for LWRT04.
LWRT06	SU	Error on SETL call for user log file <filebase>.USLB during positioning back to beginning of file following an unsuccessfully concluded write job (e.g. due to disk storage bottleneck).
LWRT07	SU	Error on SETL call for user log file <filebase>.USLA during positioning back to beginning of file following an unsuccessfully concluded write job (e.g. due to disk storage bottleneck). Error with PUT call for writing an LPUT record to the user log file <filebase>.USLA
LWRT09	SU	Error with PUT call for writing an LPUT record to the user log file <filebase>.USLB
LWRT10	SU	Error with CLOSE call for the user log file <filebase>.USLB.
LWRT11	SU	Error with CLOSE call for the user log file <filebase>.USLA.
LWRT17	A	KCSLWRT requested a buffer via KCSALME for writing to the user log file, and the buffer (class 6 memory) cannot be made available. Action: Check memory requirements and BS2000 generation.
LWRT19	SU	Faulty RELSE call for copy A of user log file.
LWRT20	SU	Same as LWRT19, but for copy B.
MACF02	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MACF03	M	XAP-TP component. The return code of SetTimer() was not equal to TI_OK.
MACF04	M	XAP-TP component. The return code of GetLogRecord was not equal to MACF_OK.
ME....	D	Diagnostic dump generated on the basis of a specific UTM message. The prefix ME is followed by the message number of the UTM message (e.g. MEK135). Activation and deactivation via the message dump function.
MFCR04	M	XAP-TP component. The return code of GetLogRecord was MACF_NO_MEM.
MFCR07	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR08	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.

Code	Group	Reason
MFCR09	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR10	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR11	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR16	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR17	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR18	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR19	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR20	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR21	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFCR24	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFDM03	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFDM04	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFDM05	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFDM06	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM05	AX	XAP-TP component. On TP_RECOVER_REQ, no free dialog table entry for a transaction branch is available. Possible cause: the number of associations in the previous application run was greater than the number of associations in the current run.
MFRM06	AX	XAP-TP component. See MFRM05
MFRM07	AX	XAP-TP component. See MFRM05
MFRM08	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.

Code	Group	Reason
MFRM09	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM10	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM11	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM12	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM13	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM14	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM15	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM16	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM17	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM18	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM19	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM21	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM24	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFRM25	AX	XAP-TP component. No free table entry available for a log damage record. Response: delete log damage records with TP_UPDATE_LOG_DAMAGE_REQ or increase the value of nMaxLogDamRec.
MFT102	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFT103	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT104	M	XAP-TP component. The return code of ChangeDescriptor was not equal to DM_OK.
MFT105	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.

Code	Group	Reason
MFT106	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT107	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT108	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFT109	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT110	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT111	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT113	M	XAP-TP component. The return code of ChangeDescriptor was not equal to DM_OK.
MFT114	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT115	M	XAP-TP component. The return code of ChangeDescriptor was not equal to DM_OK.
MFT119	M	XAP-TP component. The return code of GetLogRecord() was not equal to MACF_OK.
MFT120	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT121	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT122	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT123	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT124	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT126	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT127	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT128	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT129	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.

Code	Group	Reason
MFT130	M	XAP-TP component The return code of PutElement was not equal to DM_OK.
MFT131	M	XAP-TP component The return code of PutElement was not equal to DM_OK.
MFT132	M	XAP-TP component The return code of PutElement was not equal to DM_OK.
MFT133	M	XAP-TP component. The return code of CopyElement was not equal to DM_OK.
MFT134	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT135	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT138	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT139	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT141	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFT142	M	XAP-TP component. The function CopyElement issued a return code other than DM_OK.
MFT147	M	XAP-TP component. The function PutElement issued a return code other than DM_OK.
MFT151	M	XAP-TP component. The function PutElement issued a return code other than DM_OK.
MFTP03	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFTP04	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP05	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFTP06	M	XAP-TP component. The return code of PutElement was not equal to DM_OK.
MFTP07	M	XAP-TP component. The return code of SetTimer was not equal to TI_OK.
MFTP10	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP11	M	XAP-TP component. The return code of RequestBuffer() was not equal to LB_OK.

Code	Group	Reason
MFTP12	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP15	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP16	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP17	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP18	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP19	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP20	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MFTP24	M	XAP-TP component. The return code of the macro mGetBufferSpace() was not equal to LB_OK.
MOVE03	AU	Invalid overlapping of destination and source area in KCSMOVE.
MSCFER	DSU	Diagnostic dump following an error on an MSCF call when a task signs on or signs off at MSCF.
MSG000	DU	KCSCRMS was called with an invalid message ID. A diagnostic dump is written and the message "K000 MESSAGE NOT DEFINED" is output.
MUX001	DS	Incorrect length specification in protocol header. Possible reason: partner is not a MUX partner.
MUX003	DSU	Unexpected return code from KCSLKLC. Possibly due to protocol error in partner application
MUX004	DS	Incorrect opcode in protocol header. Possible reason: partner is not a MUX partner.
NDRCVY	D	The dump is only generated at the end of a node recovery with test mode enabled. It is used for diagnostic purposes following possible errors after a node recovery.
NMTE00	M	XAP-TP component. Module KCOCOHF, function NewMemTabEntry(). The function RequestBuffer() issued the return code LB_NOMEM.
NMTE02	M	XAP-TP component. Module KCOCOHF, function NewMemTabEntry(). The macro mGetBufferSpace() issued the return code LB_NOMEM.
NQATT1	F	The status of the application is TERM APPLICATION. The task is thus also terminated (= continuation dump).

Code	Group	Reason
NQCAT1	F	(See NQATT1) The status of the application is TERM APPLICATION. The task is thus also terminated (= continuation dump).
NQKAT1	F	(See NQATT1) The status of the application is TERM APPLICATION. The task is thus also terminated (= continuation dump).
NQPCT1	F	(See NQATT1) The status of the application is TERM APPLICATION. The task is thus also terminated (= continuation dump).
NRDBER	A	The node recovery must be terminated abnormally due to errors during database recovery. No UTM dump is generated.
ODIA00	SX	XAP-TP component. Module KCOCOHF, function OssDiagInfo(). The OSS function o_error() returned the return code O_ERROR.
ODIA01	SX	XAP-TP component. Module KCOCOHF, function OssDiagInfo(). The OSS function o_error() returned the return code O_INVEREF.
ODIA02	SX	XAP-TP component. Module KCOCOHF, function OssDiagInfo(). The OSS function o_error() returned an invalid return code.
OREA00	SX	XAP-TP component. The OSS function o_reason() returned the return code O_ERROR.
OREA01	SX	XAP-TP component. The OSS function o_reason() returned the return code O_INVEREF.
OREA02	SX	XAP-TP component. The OSS function o_reason() returned an unknown return code.
OSAFT2	F	Return code APEXT_TERMAPPL from XAP-TP.
OSGO01	SX	XAP-TP component. Module KCOCOHF, function OssGo(). The OSS function o_go() returned the return code O_INVCREF.
OSGO02	SX	XAP-TP component. Module KCOCOHF, function OssGo(). The OSS function o_go() returned the return code O_ERROR.
OSGO03	SX	XAP-TP component. Module KCOCOHF, function OssGo(). The OSS function o_go() returned an unexpected return code.
OSTM01	AS	Serious error during interaction with database system.
OSTM05	AU	Task-specific buffer for restart information is too small. Response: Increase MAX RECBUF=(...,length) in the KDCDEF generation.
OSTM06	AS	Serious error during interaction with database system.

Code	Group	Reason
OSTM07	A	A log record cannot be backed up, since the page pool is full. Response: Increase the size of the page pool. Do this by increasing MAX PGPOOL=(number,...) in the KDCDEF generation.
OSTP01	SX	XAP-TP component. Module KCOCOHF, function OssStop(). The OSS function o_stop() returned the return code O_INVCREF.
OSTP02	SX	XAP-TP component. Module KCOCOHF, function OssStop(). The OSS function o_stop() returned the return code O_ERROR.
OSTP03	SX	XAP-TP component. Module KCOCOHF, function OssStop(). The OSS function o_stop() returned an unexpected return code.
PCMM05	AU	For KCSPCMM with the opcode Get Pagechain, the NR_PAGES parameter is invalid or larger than the number of all page pool pages. Response: increase MAX PGPOOL=number in KDCDEF generation.
PCTR00	M	XAP-TP component. Module KCOCOHF, function PrepareCtrlReq(). The macro mGetBufferSpace() issued the return code LB_NOMEM.
PEND02	A	No further TACB can be written for "System PEND ER" because the page pool on KDCFILE is full. Action: Increase page pool by increasing MAX PGPOOL=(number,...) in KDCDEF generation.
PEND03	AS	The DB system has reported a serious error, see DB-DIAGAREA in UTM dump.
PEND04	S	The DB system supplied an impermissible return code, see DB-DIAGAREA in UTM dump.
PEND05	AS	See PEND03
PEND07	ASU	STXIT between writing of a confirmatory record and the (temporary) end of the UTM transaction (KCSEXPI call END_TA). Possible cause: error in the DB connection module. A timer STXIT during this period is ignored.
PEND11	S	Inconsistent XID
PEND26	A	With "System PEND ER" for a socket client, it is not possible to write a K017 message for a KDCDISP that may be required later, since the page pool on KDCFILE is full. Measure: As with PEND02
PEND97	A	In the case of a "System PEND ER" for a socket client, it is no longer possible to write a K017 message for any KDCDISP that may subsequently be necessary because the cluster page pool is full. Action: Increase the size of the cluster page pool by setting a larger value for CLUSTER PGPOOL=(number,...) in the KDCDEF generation.

Code	Group	Reason
PEND98	A	In the case of "System PEND ER", it is no longer possible to write a TACB because the cluster page pool is full. Action: Increase the size of the cluster page pool by setting a larger value for CLUSTER PGPOOL=(number,...) in the KDCDEF generation.
PENDER	ADU	Not caused by the application being aborted, but by a user-specified PEND ER, or a UTM-internal PEND ER following KCRCCC ≥ 70Z or STXIT.
PENDT1	F	Another task has terminated the application abnormally (=continuation dump before KCSBRSE call).
PENDT2	F	Another task has terminated the application abnormally (=continuation dump after KCSBRSE call).
PENDT3	F	Another task has terminated the application abnormally (= continuation dump after call by KDCROOT with PEND ER).
PLCA00	SX	XAP-TP component. Module KCOXFPL, function ActivateCmxCallback(). The CMX function t_callback() returned the return code T_ERROR.
PLCC00	SX	XAP-TP component. Module KCOXFPL, function CmxCallback(). The system function select() returned an unexpected return code.
PLCC01	S	XAP-TP component. Module KCOXFPL, function CmxCallback(). The system function select() returned an unkown return code.
PLCD00	SX	XAP-TP component. Module KCOXFPL, functionDeactivateCmxCallback(). The CMX function t_callback() returned the return code T_ERROR.
PMIO20	ASU	Integrity IDs of page inconsistent both in original and also in duplicate file. Possible cause: KDCFILE files were overwritten during live operation by copying in the start procedure. Action in this event: Correct start procedure. If this is not the cause, write PR. The following documentation is required for diagnosis: UTM dump, KDCFILE files, start procedure and task/process log. In the case of cluster applications, the cluster page pool files are also required.
PMIO22	ASU	The page type specified in the cache control table and the page type in the header of the scanned page do not match. Possible cause: see PMIO22 Action in this case: as for PMIO22
PMIO23	ASU	The page pool page to be read or written is not allocated. Cause and response: as for PMIO22
POLL03	MX	XAP-TP component. Module KCOXFPL, function ap_poll(). The return code of the macro mGetBufferSpace() was not equal to LB_OK.

Code	Group	Reason
POLL05	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() reports an ACSE event and the user data were not yet received completely on the o_event() call.
POLL07	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_ERROR.
POLL08	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_INVEREF.
POLL09	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_TIMEINT.
POLL10	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_WAKEINT.
POLL11	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_SYSTEM.
POLL12	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function ap_poll() returned the return code O_LOOK.
POLL13	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_TRANSPORT.
POLL14	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_SESSION.
POLL15	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned the return code O_SHUTDOWN.
POLL16	SX	XAP-TP component. Module KCOXFPL, function ap_poll(). The OSS function o_event() returned an unexpected return code.
PPMM10	ASU	UTM cluster application Timeout in the lockFile routine during attempt to lock a cluster page pool file. Action see CCFG19.
PPMM12	SU	UTM cluster application Timeout in the unlockFile routine during attempt to unlock a cluster page pool file. Action see CCFG19.

Code	Group	Reason
PPMM14	SU	UTM cluster application Bad return code from KCSBFMM in routine readControlPage.
PPMM15	ASU	UTM cluster application Control page of the cluster page pool has an invalid page header in the readControlPage routine.
PPMM41	SU	UTM cluster application Error when reading or writing the first CONS page of a cluster page pool file.
PPMM42	SU	UTM cluster application Bad return code from KCSBFMM when requesting a buffer.
PRIV02	SU	The attempt to set a system privilege failed.
PRIV03	SU	The attempt to reset a system privilege failed.
PWRT03	AMU	KCSPWRT has requested memory (via KCSALME), but did not get it. Action: Check memory requirements operating system generation.
PWRT06	F	With KCSPWRT, term application was called during the Periodic Write by another UTM task of the application (= continuation dump).
RALC01	AU	Index is outside the permitted range when KCSRALC is called.
RCV009	MX	XAP-TP component. Module KCOXFRV, function ap_rcv(). The return code of the function CopyElement() was not equal to DM_OK.
RCV012	MX	XAP-TP component. Module KCOXFRV, function ap_rcv(). Inconsistency in the Boolean variables <bSwitchToNextTtnid> and <bClearTtnid>.
RESTRT	D	The dump is only created during a warm start and in the debugging mode. It is used to diagnose any eventual errors in the warm start.
RQOB00	M	XAP-TP component. Module KCOCOHF, function ReqOssInBuff(). The function RequestBuffer() issued the return code LB_NOMEM.
RVCS03	M	XAP-TP component. Module KCOXFRV, function CheckSaRetc(). The return code issued by the function SetAttribute() was SA_NOMEM.
SACT14	MX	XAP-TP component. Invalid return code after calling PutElement() to request a dynamic memory area for the SACF action COPY.
SACT27	M	XAP-TP component. Invalid return code after calling PutElement() to perform save in SACF action FLUSHALL or FLUSHPAR.

Code	Group	Reason
SACT28	M	XAP-TP component. Invalid return code after call of ReAllocElement() or PutElement() in SACF action QUEUE. Action: Increase the value of the KDCDEF parameter MAX OSI-SCRATCH-AREA.
SAT002	S	Internal SAT error.
SAT003	A	Subsystem SATCP not available. Please inform the BS2000 security officer.
SAT004	A	SAT function not currently executable. Please inform the BS2000 security officer.
SAT005	A	SAT record not written. Please inform the BS2000 security officer.
SC0005	SU	The half session is generated as PLU. An RU request arrived but the request code is not SDT, STSN or CLEAR.
SC0006	SU	The half session is generated as PLU. An RU response arrived but the request code is not RQR.
SC0007	SU	#FSM.SESS is not in "ACTIVE" status.
SC0008	SU	The half session is generated as SLU. An RU request arrived but the request code is not RQR.
SC0009	SU	The half session is generated as SLU. An RU response but the request code is not SDT,STSN or CLEAR.
SC0010	SU	Protocol error for at least one of the FSMs #FSM_DT, #FSM_STSN or #FSM_RQR.
SC0011	SU	Sense code is set by at least one of the FSMs #FSM_DT or #FSM_CNTL_IMMED_EXP.
SC0014	SU	FSM_SESS_LU_LU returned an unallowed return code.
SC0019	ADU	Protocol error reported by FSM. Action: Check whether both partners are generated as DTP partners in the KDCDEF generation.
SC0020	ADU	The status of the PLU-FSM is not permitted. Action: Check whether both partners are generated as DTP partners in the KDCDEF generation.
SC0021	ADU	The status of the SLU-FSM is not permitted. Action: Check whether both partners are generated as DTP partners in the KDCDEF generation.
SC0023	DSU	When checking UNBIND-RSP for FSM_SESS_LU_LU an error has been found
SC0024	DSU	When switching UNBIND-RSP for FSM_SESS_LU_LU an error has occurred

Code	Group	Reason
SC0025	DSU	When checking UNBIND Request for FSM_SESS_LU_LU an error has been found
SC0026	DSU	When switching UNBIND Request for FSM_SESS_LU_LU an error has occurred (diagnostic dump)
SC0027	DSU	FSM_STSN_SEND returns S,RQ,STSN
SC0028	DSU	Unauthorized STSN-REQ for PLU_TO_SLU
SC0029	DSU	Unauthorized STSN-REQ for SLU_TO_PLU
SC0030	DSU	ACT_SC_RECEIVE could not set any return code
SC0032	A	The session cannot be activated since there are no connections available. Action: Change KDCDEF generation, include more CON statements.
SC0034	DSU	The partner has made a protocol error on setting STSN response
SDCS02	M	XAP-TP component. Module KCOXFSD, function CheckSaRetc(). The SetAttribute() return code was SA_NOMEM.
SDUI01	M	XAP-TP component. Module KCOXFSD, function UserDataIn(). The macro mGetBufferSpace() issued the return code LB_NOMEM.
SEND05	SU	Severe BCAM error when processing a BCAM SENDLET calls. Error when canceling a timer for output terminals.
SG-nnn	D	Diagnostic dump generated on the basis of a KCRSIGN1/2 return code after a SIGN ON call. The prefix SG- is followed by the return code (e.g. SG-U17). Activation and deactivation via the message dump function.
SIGND1	DU	The call of the Kerberos macro returned an unexpected return code. Only if debugging mode is active.
SIGND2	DU	The call of the Kerberos macro returned an unexpected return code. Only if debugging mode is active.
SIGND3	DU	The call of the Kerberos macro returned an unexpected return code. Only if debugging mode is active.
SIGND4	DU	The call of the Kerberos macro returned an unexpected return code. Only if debugging mode is active.
SLOG05	SU	KCSLKLC returned a bad return code when attempting to lock the message buffers (possibly TIMEOUT).
SLOG06	SU	KCSLKLC returned a bad return code when attempting to unlock the message buffers.
SLOG07	SU	KCSLKLC returned a bad return code when attempting to lock the SYSLOG file (possibly TIMEOUT).
SLOG08	SU	KCSLKLC returned a bad return code when attempting to unlock the SYSLOG file.

Code	Group	Reason
SLOG09	SU	Attempt to write message buffer to current SYSLOG file failed (the DMS error code in the preceding K043 message may provide information about the cause of the error).
SLOG10	SU	The attempt by a follow-up task to switch to the current SYSLOG file generation failed (please refer to any preceding K043 message).
SLOG21	ASU	In the administration call KDCSLOG INFO or in the call to the program interface for administration with the opcode KC_SYSLOG and the sub-opcode1 KC_INFO, KCSSLOG called the component KCSFST in order to determine the attributes of the file generation group of the SYSLOG (only when the SYSLOG is created as a FGG). KCSFST gave a DMS error (also take note of preceding K043 message).
SLOT07	AM	The address of a slot must be calculated in KCSSLOT. The task does not yet know the memory pool that contains the slot. The attempt to link up with the memory pool failed due to insufficient address space. Action: Check memory requirements and BS2000 generation.
SLOT11	SU	Internal procedure GET_MP: When requesting an additional slot pool, the size of the additional slot pool is different from the size of the first slot pool..
SM2001	SU	Call of KCSSM2 with incorrect opcode.
SM2002	SU	Call of KCSSM2 with opcode SM2_CONNECT when KAA.STATE.SM2 = SM2_OFF.
SM2003	SU	Call of KCSSM with "write data" opcode, but KAA.SM2.RECORD_ADDR = NULL.
SM2004	SU	Incorrect parameters with DSSM call CONNECT_SS.
SM2005	SU	Incorrect parameters with DSSM call DISCONNECT_SS
SM2006	SU	DSSM call CONNECT_SS, but task is already connected.
SM2007	SU	Call of KCSSM2 when KAA.STATE.SM2 = SM2_NO.
SMSG00	AU	SMSG00 is a diagnostic dump and does not lead to the end of the application run; a SMSG00 dump is only taken when test mode is activated. An SMSG dump is created if KCSSMSG wants to send a message to a message destination of the type MSG-DEST and receives a bad return code from KCSFPUT. The return code can be taken from the NKB table of the UTM dump; the CC and DC codes of the DPUT call are at distance X'5C' in this table. One possible reason is that the queue/TAC is locked.
SMSG03	ASU	The component KCSSLOG supplied a bad return code on writing a message to the SYSLOG. Cause: See DMS return code in K043 message
SMSG09	D	SMSG09 is a diagnostic dump and does not cause the application run to terminate; an SMSG09 dump is only created if test mode is enabled.

Code	Group	Reason
SND005	M	XAP-TP component. Module KCOXFSD, function ap_snd(). The UserDataIn() function returned the UDIN_NOMEM return code.
SND007	MX	XAP-TP component. Module KCOXFSD, function ap_snd(). The return code of the function GetVarLthAttr() was not equal to GA_OK.
SSO002	DSU	Diagnostic dump; created after a serious error when the SRPMSSO subsystem is called.
STnnnn	ADSU	Error when processing the start of a UTM task, where nnnn is the number indicating the error cause in the UTM message "K049 Error <nnnn> during application startup".
STPF10	SX	XAP-TP component. Invalid indicator for HeurRep parameter in TP-HEUR-REP-RI.
STPF11	SX	XAP-TP component. Invalid indicator for FuUnits parameter in TP-BEG-DIAL(dialogue)-RC.
STPF12	SX	XAP-TP component. Invalid indicator for Result parameter in TP-BEG-DIAL(dialogue)-RC.
STPF13	SX	XAP-TP component. Invalid indicator for Diagnostic parameter in TP-BEG-DIAL(dialogue)-RC.
STPF14	SX	XAP-TP component. Invalid indicator for user data parameter in TP-BEG-DIAL(dialogue)-RC.
STPF15	SX	XAP-TP component. Invalid indicator for Result parameter in TP-BEG-DIAL(channel)-RC.
STPF16	SX	XAP-TP component. Invalid indicator for Diagnostic parameter in TP-BEG-DIAL(channel)-RC.
STPF17	SX	XAP-TP component. Invalid indicator for HeurRep parameter in TP-(ABORT-AND-)HEUR-REP-RI.
STPT02	SX	XAP-TP component. Ungültiger Wert in p_udt.p_udtnxt.
STPT10	SX	XAP-TP component. Invalid indicator for InitTPSUT parameter in TP-BEG-DIAL(dialogue)-RI.
STPT11	SX	XAP-TP component. Invalid indicator for RecTPSUT parameter in TP-BEG-DIAL(dialogue)-RI.
STPT12	SX	XAP-TP component. Invalid indicator for FuUnits parameter in TP-BEG-DIAL(dialogue)-RI.
STPT13	SX	XAP-TP component. Invalid indicator for BegTrans parameter in TP-BEG-DIAL(dialogue)-RI.
STPT14	SX	XAP-TP component. Invalid indicator for Conf parameter in TP-BEG-DIAL(dialogue)-RI.

Code	Group	Reason
STPT15	SX	XAP-TP component. Invalid indicator for LastPartId parameter in TP-BEG-DIAL(dialogue)-RI.
STPT16	SX	XAP-TP component. Invalid indicator for user data parameter in TP-BEG-DIAL(dialogue)-RI.
STPT17	SX	XAP-TP component. Invalid indicator for FuUnits parameter in TP-BEG-DIAL(channel)-RI.
STPT18	SX	XAP-TP component. Ungültiger ChanUtil parameter in TP-BEG-DIAL(channel)-RI.
STPT19	SX	XAP-TP component. Invalid indicator for LastPartId parameter in TP-BEG-DIAL(channel)-RI.
STPT20	SX	XAP-TP component. Invalid indicator for CcrTokReq parameter in TP-BID-RI.
STPT21	SX	XAP-TP component. Invalid indicator for LastPartId parameter in TP-BID-RI.
STPT22	SX	XAP-TP component. Invalid indicator for Result parameter in TP-BID-RC.
STPT23	SX	XAP-TP component. Invalid indicator for Conf parameter in TP-END-DIAL-RI.
STPT24	SX	XAP-TP component. Invalid indicator for ConfUrg parameter in TP-HSK-RI.
STPT25	SX	XAP-TP component. Invalid indicator for ConfUrg parameter in TP-HSK-AND-GRT-CTRL-RI.
STPT26	SX	XAP-TP component. Invalid indicator for Type parameter in TP-DEFER-RI.
STPT27	SX	XAP-TP component. Invalid indicator for Reason parameter in TP-TOKEN-GIVE-RI.
STPT28	SX	XAP-TP component. Invalid indicator for Correlator parameter in TP-TOKEN-GIVE-RI.
STRT01	DU	When changing the application program the number of processes which still are to be changed is less than zero, i.e. the counter is inconsistent. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.
SYPM01	AU	Task-specific buffer for restart information is too small. Response: increase MAX RECBUF=(...,length) in KDCDEF generation.
SYPM03	AS	Serious error in communication with DB system (FITA).
SYPM11	AU	Task-specific buffer for restart information is too small. Response: increase MAX RECBUF=(...,length) in KDCDEF generation.
TADR03	SU	The so-called Application Reference in the announcement is invalid.

Code	Group	Reason
TC0040	DSU	The input message length supplied by the transport system does not match the length contained in the message.
TC0041	DSU	The input message length supplied by the transport system is shorter than the value in the Data Count Field (DCF) in the Transmission Header (TH).
TCPI13	SU	Invalid length of the IP address (TCP_GET_HOST_BY_ADDR).
TCPI14	SU	Invalid length of the IP address (TCP_CONNECTION_REQUEST).
TIEV19	SU	Inconsistence between the announced length and the actual length of data.
TIMR02	SU	Error on $NCANTI.
TIMR03	SU	Error on $NCHNGTI.
TINF01	SU	Error when reading the task priority.
TINF02	SU	Error when setting the task priority.
UMDT12	SU	KCCUMDT opcode UMDT_LOCK_ULS Bad return code from KCSBFMM when reading a ULS administration file.
UMDT13	SU	KCCUMDT opcode UMDT_LOCK_ULS Bad return code from KCCLKMT.
UMDT15	ASU	UTM cluster application KCCUMDT opcode UMDT_UNLOCK_ULS: Error on locking the ULS file. Action see Action see section „Actions when locking UTM cluster files" below this table.
UMDT32	SU	UTM cluster application KCCUMDT function readHdrPage Bad return code from KCSBFMM when requesting a buffer.
UMDT33	ASU	UTM cluster application KCCUMDT releaseFileLock function: Timeout while unlocking the cluster ULS file. Action see UMDT15.
UMDT34	SU	UTM cluster application KCCUMDT releaseFileLock function: Incorrect return code from KCCGFLK when unlocking the cluster ULS file. Action see UMDT15.
UMDT42	SU	UTM cluster application KCCUMDT function addrUlsEntry Bad return code from KCSBFMM when requesting a buffer.
UMDT44	SU	UTM cluster application KCCUMDT setFileLock function: Incorrect return code from KCCGFLK when locking the cluster ULS file. Action see UMDT15.

Code	Group	Reason
UMDT52	ASU	UTM cluster application KCCUMDT allocateUlsEntry function: Incorrect return code from KCCUSF when locking the cluster user file. Action see UMDT15.
UMDT63	ASU	UTM cluster application CUMDT entry KCCUGLI: The cluster ULS file cannot be locked. Action see UMDT15.
UMDT73	SU	UTM cluster application KCCUMDT function checkFile Bad return code from KCSBFMM when requesting a buffer.
UMDT74	SU	UTM cluster application KCCUMDT function checkFile Bad return code from KCSBFMM when requesting a buffer.
UMDT77	SU	UTM cluster application KCCUMDT function checkFile Bad return code from KCSBFMM when requesting a buffer.
UMES02	DSU	System error. For meaning see message K146.
USF002	ASU	UTM cluster application. Module KCCUSF, opcode OPEN_FILE. Timeout when initializing global file lock. Action: Increase the value of the FILE-LOCK-TIMER-SEC or FILE-LOCK-RETRY parameter in the CLUSTER statement of the KDCDEF generation.
USF003	SU	UTM cluster application Module KCCUSF, opcode OPEN_FILE. Invaild return code when initializing the global file lock.
USF009	ASU	UTM cluster application. Module KCCUSF, opcode CUSF_RESET_PTC_FLAG. Five timeouts on locking the cluster user file. Action see USF002.
USF013	ASU	UTM cluster application. Module KCCUSF, opcode CUSF_SIGNOFF_USER_WITH_CONTEXT. Five timeouts on locking the cluster user file. Action see USF002.
USF018	ASU	UTM cluster application. Module KCCUSF, opcode CLOSE_FILE. Timeout when destroying file lock. Action see USF002.
USF019	SU	UTM cluster application Module KCCUSF, opcode CLOSE_FILE. Invaild return code from KCCGFLK when destoying the file lock.

Code	Group	Reason
USF021	SU	UTM cluster application Module KCCUSF, function setFileLock Invaild return code from KCCGFLK when requesting the exclusive lock.
USF023	ASU	UTM cluster application. Module KCCUSF, function releaseFileLock. Timeout when releasing file lock. Action see USF002.
USF024	SU	UTM cluster application Module KCCUSF, function releaseFileLock Invaild return code from KCCGFLK when releasing the file lock.
USF025	ASU	UTM cluster application. Module KCCUSF, function readPage. Timeout when requesting shareable file lock. Action see USF002.
USF026	SU	UTM cluster application Module KCCUSF, function readPage Invaild return code from KCCGFLK when requesting the shareable file lock.
USF027	ASU	UTM cluster application. Module KCCUSF, function readPage. Timeout when releasing file lock. Action see USF002.
USF028	SU	UTM cluster application Module KCCUSF, function readPage. Invaild return code from KCCGFLK when releasing the file lock.
USF051	SU	UTM cluster application Module KCCUSF, function writePage Error when writing the cluster user file.
USF052	SU	UTM cluster application Module KCCUSF, function readPage. Error when reading the cluster user file.
VTSUER	ADS	VTSU-B reports an error connected with the applied character set or the encryption. The application is not aborted. The Dump serves the diagnosis of the error reported by VTSU-B. The Dump contains only KDCROOT and KTA data.
WAI102	DU	Diagnostic dump.
WAI112	DU	When switching the system protocol file the number of processes which still are to be switched is less than zero, i.e. the counter is inconsistent. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.

Code	Group	Reason
WAI113	DU	When changing the application program the number of processes which still are to be changed is less than zero, i.e. the counter is inconsistent. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.
WAI114	DU	See WAI113
WAI118	DU	Aktion PROC_ACCESS_FROM_USER_INDX: UTM_INTERNAL_MSG_ANNO received but no UTM_INTERNAL_MSG_ANNO, UTM_START_PROG_ANNO, UTM_ROLLBACK_PET_ANNO and no UTM_COMNMIT_PET_ANNO. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.
WAI119	DU	Aktion PROC_ACCESS_FROM_USER_INDX: UTM_INTERNAL_MSG_ANNO received but no UTM_START_PROG_ANNO, UTM_ROLLBACK_PET_ANNO and no UTM_COMNMIT_PET_ANNO. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.
WAI120	DU	When administering the diagnostic trace settings the number of processes which still have to update the settings is less than zero, i.e. the counter is inconsistent. The application is terminated abnomallly if debug mode is activated, otherwise a diagnostic dump is written.
WAIT01	SU	Bad return code when calling KCSBRSE for the work bourse in action block ENQUEUE_WORKBOURSE von KCSWAIT.
WAIT02	DS	Invalid timer announcement type (action block ENQUEUE_WORKBOURSE).
WAIT03	S	Invalid type in announcement and no timer announcement (action block ENQUEUE_WORKBOURSE).
WAIT28	SU	Bad return code from Module KCSSLOG (before ENQUEUE_WORKBOURSE).
WAIT38	S	KCSWAIT starts a TU action for a service which is in PGWT status.
WAIT55	SU	Invalid code for a internal timer announcement (action block ANALYSE_INTERNAL_TIMER).
WAIT78	ASU	Severe error when calling the BCAM socket functionality.
WAITT1	F	Another task has terminated the application abnormally (=continuation dump before calling KCSBRSE).
WAITT2	F	Another task has terminated the application abnormally (=continuation dump after calling KCSBRSE).

Code	Group	Reason
XATT02	F	XAP-TP component. Module KCOXFEX, function apext_att(). The function bCheckAndSetState() issued a bad return code and the application status was WAITING_DUMP_APPL. A different task had already caused the application to abort.
XATT04	MX	XAP-TP component. Module KCOXFEX, function apext_att(). The return code of the function EstablishBuffer() was not equal to LB_OK.
XATT12	MX	XAP-TP component. Module KCOXFEX, function apext_att(). The return code of the function EstablishBuffer() was not equal to LB_OK.
XATT13	MX	XAP-TP component. Module KCOXFEX, function apext_att(). The return code of the function RequestBuffer() was not equal to LB_OK.
XFAG09	M	XAP-TP component Module KCOXFHF, function GetAttribute(). The function RequestBuffer() issued the return code LB_NOMEM.
XFGA07	SUX	XAP-TP component. Module KCOXFHF, function GetAttribute(). The function AllocUserMem() returned an unexpected return code when reading the attribute AP_DTNID in the single task mode.
XFGA11	M	XAP-TP component. Module KCOXFHF, function GetAttribute(). The macro mGetBufferSpace() issued the return code LB_NOMEM.
XFGE01	F	XAP-TP component. Module KCOXFHF, function bCheckAndGetCallEnv(). The function bCheckAndSetState() issued a bad return code and the application status was WAITING_DUMP_APPL. A different task had already caused the application to abort.
XFSA07	MX	XAP-TP component. Module KCOXFHF, function SetAttribute(). The function PutElement() issued a return code other than DM_OK when setting the AP_DTNID attribute.
XFTM01	SX	XAP-TP component. Module KCOXFHF, function TraceMgmt(). The OSS function o_tron() returned the return code O_ERROR.
XFTM02	SX	XAP-TP component. Module KCOXFHF, function TraceMgmt(). The OSS function o_tron() returned the return code O_INVEREF.

Code	Group	Reason
XFTM03	SX	XAP-TP component. Module KCOXFHF, function TraceMgmt(). The OSS function o_tron() returned an unknown return code.
XFTM04	SX	XAP-TP component. Module KCOXFHF, function TraceMgmt(). The OSS function o_troff() returned the return code O_ERROR.
XFTM05	SX	XAP-TP component. Module KCOXFHF, function TraceMgmt(). The OSS function o_troff() returned an unknown return code.
XINI06	ASX	XAP-TP component. Module KCOXFEX, function apext_init(). The OSS function o_create() returned the return code O_ERROR. Possible causes: - If the in insert XPFUNC has the value o_create() and the insert XPRET the value -1 in the corresponding UTM message P001, then the error may be that the OSS subsystem has not been started. - A further OSI TP application was started in the dialog in the same task. However, OSS is still reserving resources of a previously terminated application and this is the reason for termination. Occurs in particular if the application has previously been started in two tasks in the dialog and then terminated again. Remedy: The resources are freed after a LOGOFF BUT.
XINI07	SX	XAP-TP component. Module KCOXFEX, function apext_init(). The OSS function o_create() returned an unknown return code.

Actions when locking UTM cluster files

The following workaround is recommended for all errors that occur when requesting or releasing locks for UTM cluster files.

► In the CLUSTER statement of the KDCDEF generation, increase the value of the FILE-LOCK-TIMER-SEC parameter or FILE-LOCK-RETRY parameter.

For a more detailed diagnosis, please also see the most recent K190 messages.

K061 Dump file &FNAM created

K062 Dump file could not be created.

K063 Formatting error &RCF1A/&RCF2A for format &FMTN, asynchronous message for LTERM = &LTRM

K064   Message discarded : &PTRM/&PRNM/&BCAP/&LTRM &DEVC &FIL1A &FIL2A &FIL3 &VTRC &CBRC &IMSG2 &REA1

Meaning

1. An invalid or unexpected announcement was received at the work bourse. In this event, insert &REA1 has a value of X'05' and insert &IMSG2 contains the announcement received.

2. A message from an LTERM partner has been received which is logically inconsistent for UTM, e.g. because it does not comply with the strict dialog, or a message cannot be output because it is rejected by the VTSU.

   As an insert the message contains a 4-byte return code which VTSU passes to UTM and, in addition, the return code which VTSU passes in the VTSUCB.

   If the insert &REA1 has the value X'06', the insert contained in the message is the return code passed to UTM by the BS2000 subsystem ASECO.

3. A message which does not contain a valid transmission header was received from an LU6.1 partner.

For diagnostic purposes the first 32 characters of the message are output. In this way blocked input devices, for example, can be detected. *Caution*: these characters may sometimes contain the remains of an earlier message in the case of length < 32 or K key.

In response to this message the connection is cleared down or an automatic KDCDISP is performed.

The standard record output to SYSLOG has the following structure:

INSERTS	Meaning
&DEVC	Terminal or printer type; the codes can be taken from the macro DCSTA with TYPE=TCHAR.
&FIL1A	Status of the application; possible values X'02' = start phase X'03' = normal run X'04' = shut warn X'05' = shut grace X'07' = fast shut X'08' = term application

INSERTS	Meaning
&FIL2A	Status of the client or printer: X'00' = Connection clearing down X'01' = Connection not established X'02' = UTM waiting for a connection request to be confirmed X'03' = Connection established, user not yet signed on X'04' = UTM waiting for ID card to be inserted X'05' = UTM waiting for a password to be entered X'06' = Sign-on via MUX X'07' = 2nd part of the sign-on service X'08' = LTERM with USAGE=O: Printer connected, otherwise user logged on
&FIL3	Status of the physical terminal or printer (2 bytes), serves as a diagnostic aid in the event of errors. The meaning of the individual bits may be found below.
&VTRC	VTSU return code
&CBRC	Return code in the VTSUCB
&IMSG2	The first 32 characters of the message (in the case of stations the first 8 characters contain the header of the station log)
&REA1	Reason for message K064: X'01' = Bad VTSU return code X'02' = Invalid function key X'03' = No positive print acknowledgment X'04' = Bad return code from ISLP X'05' = Inconsistent input message X'06' = LU6.1: The announcement length is shorter than the DCF in the transmission header X'07' = Input message for disconnected MUX connection X'08' = Inconsistent message from an LU6.1 partner X'09' = UPIC input message with illegal protocol X'0A' = Inconsistent physical or logical acknowledgment X'0B' = Input message from a socket client generated as an output partner X' 0C' = An unexpected GO signal was received for a UPIC or socket connection.

The individual bits of the &FIL3 insert have the following meaning:

x	x	x					x	x					

1 = Partner is a dialog partner (client)
0 = Partner is a printer

1 = Partner is a station (terminal)
0 = Partner is an application

1 = Last output was in extended line mode

1 = Waiting for transport or logical print
  acknowledgement

1 = Last output was unformatted
0 = Last output was formatted
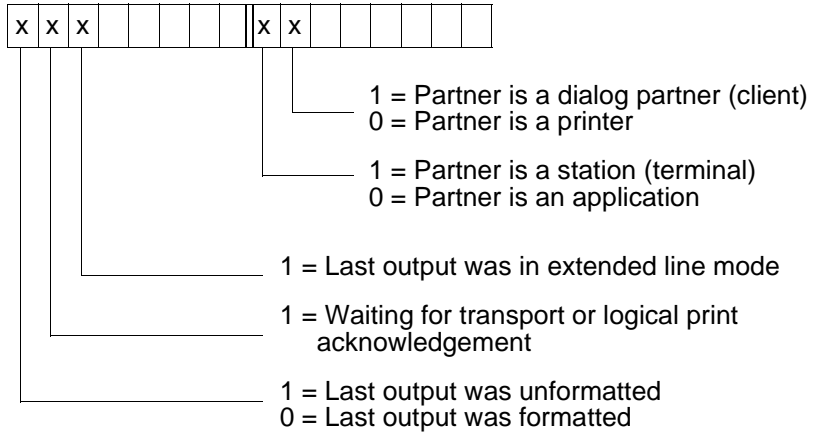
K065   Net message: &PTRM/&PRNM/&BCAP/&LTRM &FIL1B &FIL2B

Inserts &FIL1B and &FIL2B have the following meaning:

Inserts	Meaning
&FIL1B	BCAM call or BCAM, UTM or RSO announcement type
&FIL2B	BCAM info word, (see table on page 235), UTM announcement code or RSO return code

&FIL1B is output in printable form. The values X'E0', X'E2' and X'F0' to X'FF' identify a UTM (timer) announcement. The values are explained in the following table.

&FIL1B	Meaning	Meaning of &FIL2B
X'01' - X'4C'	BCAM call or BCAM announcement [1]	BCAM info word
X'56'	RSO announcement	Asynchronous RSO return code (printable, see K150), the value and meaning can be found in the description of message K150 or obtained from System Support.
X'E0'	Idle timeout for an LU6.1-session (see KDCDEF-statement SESCHA, parameter IDLETIME)	None
X'E2'	Timeout on reception of message fragments at a socket connection since the message was not fully received within the permitted period of 10 minutes (internal value)	None
X'F0'	UTM anno	If the first byte of &FIL2B is X'FD', then a shutdown announcement from MSCF was received and the application terminates.
X'F1' - X'FA'	UTM anno	None
X'FB'	UTM anno: Timeout waiting for an acknowledgment from a printer or TS application after sending a queued message (see KDCDEF statement MAX, parameter LOGACKWAIT)	None

&FIL1B	Meaning	Meaning of &FIL2B
X'FC' - X'FE'	UTM anno	None
X'FF' X'FF'	UTM anno: timeout	The first byte of &FIL2B specifies the cause of the timeout:
		X'21' Expiry of the timer which monitors the establishment of an LU6.1 session. See KDCDEF statement UTMD, parameter CONCTIME=(**time1**,....).
		X'22' Expiry of the timer which monitors the reception of the acknowledgment for a queued message sent via an LU6.1 session. See KDCDEF statement UTMD, parameter CONCTIME=(....,**time2**).
		X'30' Expiry of the PEND-KP timer. See KDCDEF statement MAX, parameter TERMWAIT or PGWTTIME.
		X'40' to X'43' Expiry of the PTERM idle timer. See KDCDEF statement PTERM or TPOOL, parameter IDLETIME or KDCDEF statement MAX, parameter PGWTTIME.

[1]   The value and meaning of the BCAM call and announcement can be obtained from the BCAM diagnostic documentation or System Support

Codes of the BCAM info word

The table below the codes of the BCAM info word, their meaning and possible actions to recover errors.

BCAM has an additional interface from DCAM V10 onwards. Different return codes are generated depending on usage. The first column of the table therefore contains the new return code (arranged in ascending order in byte 4) and the old return code in the second column (arranged in ascending order in byte 1).

With a BCAM call the new code is returned (&FIL1B < X'40'), with a BCAM announcement the old code is returned (&FIL1B ≥ X'40').

New code	Old code	Meaning	Actions
00820004	04000000	BCAM not active	BCEND-DCSTART
0001xx08 00010014 00010108 00010208 00010308	08000000	Illegal structure of parameter list: Letter too long CONNB not specified NAME incorrect APID not specified	Inform software maintenance staff
00010408 00010508 00010608 00010708 00010808 00010908 00010A08 00010B08 00010C08 00010D08 00010E08 00010F08 00011008 00011108 00011208 00011308 00011408 00011508 00011608 00011708 00011808 00011908 00011A08 00011B08 00011C08	08000000	DID not specified NAME not specified PNAME not specified CID not specified LETAR not specified TELAR not specified NBUFS exceeds maximum distribution code length=0 distribution code length exceeds maximum NGBS not specified GROB not specified distribution code not specified distribution bourse ID not specified APREF not specified BLTH not specified user buffer length = 0 TAKAR not specified RTACK incorrect telegram length = 0 LANNO incorrect NBUFS not specified EVENT incorrect RCCOD incorrect Number of routes outside permitted limit OWNPN incorrect	
0080000C	0C00000	Timeout	Check partner
00400010	10000000	No message received	UTM BCAM trace
00010014	14000000	Connection message too long (letter too long)	Check structure of the connection message
00010018	18000000	DISTAB error	Inform software maintenance staff

New code	Old code	Meaning	Actions
0480xx1C 0480011C	1C040000	Resource bottleneck memory for BCAM: Data buffer	Repeat call/enlarge memory with BCMOD ask software maintenance staff to examine data flow problems
0480021C 0480031C 0480041C 0480051C 0480061C 0480071C 0480081C 0480091C 04800A1C 04800B1C		Reference ACONCB or REQCB APPCB DISTAB Name Manager Interface flow control Bourse ENACB ADDRCB-P CID	If error recurs, inform software maintenance staff
0840011C	1C080000	Warning: BCAM will be terminated	
0C40021C	1C0C0000	BCAM will be terminated	
1082001C	1C100000	Maximum number of predefined applications has been reached	/BCMOD-MAXNPA=
1440001C	1C140000	Waiting for GO signal	
04010020	20000400	Application not yet known to BCAM	Open application with BCAM
08010020	20000800	Application known to BCAM under another task	Repeat call with "correct" task
0C010020	20000C00	Application signed on to BCAM exclusively through another task	
10010020	20001000	Application already signed on with BCAM under this task	
14400220	20001400	Forced closure of the application	Reopen application
18400120	20001800	Forced closure of the application with warning	Reopen application
1C010020	20001C00	Application is not active	/BCACT application name
20010010	20002000	Application may not actively set up connection	

New code	Old code	Meaning	Actions
24820020	20002400	No further connections are allowed for this application	Increase number of connections with the BCMOD command
28820020	20002800	No non-predefined application allowed for this task	Increase with BCMOD MAXNPT
2C010020	20002C00	Application - password error	Correct password
30010020	20003000	Call already issued by another task	
34010020	20003400	Application may not work with high priority	Repeat call with lower priority
04010024	24000004	Partner has no connection	Set up connection
08010024	24000008	Partner already has a connection	
0C010024	2400000C	The application has issued another connection setup request to a partner, but the partner has not yet responded	
10400024	24000010	Partner is already attempting to set up connection	Forget own connection request and accept partner's connection request
14010024	24000014	Connection request did not exist or time elapsed	Accept connection request within the allotted time
18010024	24000018	Partner not known	Generate partner correctly
1C820024	2400001C	Partner processor not known	Check generation of BCIN partner processor
1C820124		Partner processor not active	/BCACT
1C820224		ROUTE not known	Check generation
1C820324		ROUTE not active	/BCACT
	2400002C	Partner does not accept the connection request	Generate DCM.BCAM.CON trace
30010024	24000030	Application may not actively set up a connection	
	24000034	The partner has lost the connection and will set it up again immediately	Wait for connection setup
	24000038	Connection cleared down by the partner	

New code	Old code	Meaning	Actions
	2400003C	Forced closure of the application by the operator	
4001xx24 40010124 40010224 40010324 40010424	24000040	Inconsistent connection parameters: TELAR requested telegrams not permitted predefined connection: call requires GROS=F or THRU=N user data not permitted	Correct connection parameters
00800028	28000000	Attempt input/output later	No acknowledgment for connection log: Look for cause in destination processor, Modify UTM parameters BRETRYWAIT and BRETRYNR in KDCDEF
0040002C	2C000000	Message rejected after timeout	User problem / use more tasks with UTM
0020xx30	30000000	BCAM error	Take DUMP, create ASTRID TOTAL dump
	34000000	Inconsistent network address	Re-establish connection
00800038	38000000	Another input/output is currently running	Repeat call/ if error recurs again turn on DCM.BCAM.CON trace and make copy
0001003C 0001FFFF 0002FFFF 0003FFFF 0004FFFF	3C000000	Illegal call Illegal unit/function Function not available Illegal version Incorrect parameter list	Inform software maintenance staff

K066 Mandatory parameter FILEBASE not specified

K067 Error in interoperation with language connection module &MOD: error code =
&ERCD1 &ERCD2; opcode = &OPCD2

The inserts of the message have the following meaning:

&MOD	Meaning
KDCCC	Connection module for C
KDCCCOB	Connection module for COBOL
KDCCASS	Connection module for ASSEMBLER
KDCCFOR1	Connection module for FORTRAN
KDCCILC	Connection module for ILCS
KDCCSPL	Connection module for SPL
UTMPLI1	Connection module for PL/I
UTMPAS	Connection module for Pascal

&OPCD2	Meaning
INITIALISE	Initialize the language environment
DEACTIVATE	Release the language environment
STXIT-HANDLING	STXIT handling via the language
S; XXXXXXXX	Start the program unit ' XXXXXXXX'
E; XXXXXXXX	Execute the end handling routine
	for the program unit ' XXXXXXXX'

The error code is made up of 4 characters.

&ERCD1	Meaning and any action required
0000	Job has already been executed
0004	Job was not executed correctly
	Compile error documentation and send to Systems Support
0008	Error in application program
00FF	The language connection module for a generated language is not
	linked in at the start of the application

The inserts in &ERCD2 supply additional information on errors that have occurred
and are used for diagnostic purposes. The meaning can be found in the manuals
for the relevant programming languages.

K068 Database connection module &DBCON version &DBV1 cannot interoperate with
KDCDB macro version &DBV2

K069   Disconnection : &PTRM/&PRNM/&BCAP/&LTRM/&REA4/&REA6/&COTM

The insert &REA4 indicates who initiated the connection cleardown. The insert &REAS1 contains the reason for the connection cleardown:

&REA4	Meaning
B	Transport system reports connection lost
L	Connection loss for socket connection
else	Connection was cleared down by openUTM, for reason see &REA6

&REA6	Meaning
X'00'	Reason not specified
X'08'	Resource bottleneck
X'09'	Connection cleared down
X'0A'	Application SHUTDOWN
X'0D'	Negative return code from transport system, see associated K065 message.
X'1A'	Too many invalid sign-on attempts
X'2A'	Connection interrupted after an unencrypted message was received from the terminal on an encrypted connection.
X'2F'	Closure of MUX session after the MUX transport connection had already been cleared down.
X'30'	Connection cleardown by KCSMUX
X'31'	Connection cleardown after KDCOFF
X'32'	Connection cleardown after SIGN OF
X'33'	Connection cleardown by the print administration
X'34'	Connection cleardown by the administration
X'35'	No messages for printer with PLEV > 0
X'36'	Page pool warning level 2 exceeded on receipt of an asynchronous DTP message.
X'37'	Connection cleardown after receipt of an incorrect status message from the terminal
X'38'	Bad return code from VTSU-B or unencrypted message received although permanent encryption was defined for the connection
X'39'	Invalid function key
X'3A'	Connection cleardown by DTP with session not yet established
X'3B'	Connection cleardown by UPIC
X'3C'	Connection cleardown after unsuccessful sign-on attempt
X'3D'	Connection cleardown after unsuccessful sign-on service

&REA6	Meaning
X'3E'	Connection cleardown after timeout
X'3F'	Message fragment received
X'40'	Negative transport or print acknowledgment
X'41'	Unexpected protocol element
X'42'	Inconsistent input message
X'43'	Connection cleardown through new connection setup request
X'44'	Connection cleared down after a negative acknowledgment from RSO.
X'47'	Connection cleared down because IDLE timer timed out.
X'49'	Socket - Connection cleared down because of invalid length of incoming message (negative or >32000)
X'4A'	Socket - Connection cleared down because of invalid version in the protocol header
X'4B'	Socket - Connection cleared down because of invalid type in the protocol header
X'4C'	Socket - Connection cleared down after an error occurred while writing the message in the message queue
X'4D'	Socket - Connection cleared down after timeout
X'4E'	Socket - Connection cleared down because of a length problem: the generated message is longer than the generated maximum length (cf. MAX TRMSGLTH operand in the KDCDEF statement)
X'4F'	Socket - Connection cleared down because of unknown identifier in the protocol header
X'50'	Socket - Connection cleared down because of invalid version minor in the protocol header
X'51'	Socket - Connection cleared down because of invalid flag in the protocol header
X'52'	Socket - Connection cleared down because the maximum possible number of fragments was exceeded
X'53'	Socket - Connection cleared down because of an invalid message type FRAGMENT at the beginning of a message
X'54'	Socket - Connection cleared down because the last part of a fragment was missing.
X'56'	Connection to RSO printer is shut down as the KDCS call FPUT/DPUT RP is not supported by the current RSO version.
X'57'	Connection to client is shut down as after a timeout on the PGWT bourse too few processes are active to continue the procedure normally and this may have caused a loss of sync in the dialog with the client.
X'59'	Connection cleared down by DTP while session established.

&REA6	Meaning
X'60'	Cleardown of a socket connection due to no longer supported protocol version 1.0 (minor version X'00') in the protocol header.

K070 USER / LSES inactive : &USER, &GLOBALSG

A value is only entered for the insert &GLOBALSG 'Cluster Global Sign' in UTM cluster applications. The insert can have the following values:

&GLOBALSG	Meaning
Y	Global sign-off at the cluster
N	Local sign-off at the node
T	The global sign-off at the cluster was not successful because the cluster user file could not be locked in the generated time.
F	The global sign-off at the cluster was not successful because the user was not found in the cluster user file.
S	The global sign-off at the cluster was not successful because the user was not signed on globally at the cluster.
O	The global sign-off at the cluster was not successful because the user was not signed on globally at the cluster at this node.
' '	Sign-off for an LU6.1 session user or a connection user.

The message is output when a user signs off. The message also has the following inserts:

&COTM   Number of seconds since signon of &USER to this connection.

&CPTM   Consumed CPU time in msec of the user ID &USER if only one user was still currently signed on under &USER. If users have previously signed on to several connections under &USER, the value also includes their CPU consumption. Afterwards the value is set back to 0.

  If further users are logged in under &USER, this value is 0.

K071   Internal error in database connection &DBCON: operation=&OPCD1,
           error=&ERCD3, DB-trace-info=&DBTRAC

The inserts of the message have the following meaning:

&DBCON   Name of the connection module to the DB system

&OPCD1   DB operation code, see the "DB-DIAGAREA" on page 107.

&ERCD3   Error code KCRCDC with the possible values KU04 through
            KU24, see page 447ff.

&DBTRAC   DB trace information. The possible values and their meaning are
            described in relation to the relevant database system.
            For UDS-DB system:
            Interpretation of DB trace information, see description of return
            codes in the manual "UDS/SQL Messages".
            For the SESAM-DB system: see the manual "SESAM/SQL
            Database Operation", section "Diagnostic information in the DB-
            DIAGAREA of openUTM" in the chapter "Error recovery"

            If the DB system is connected via the XA interface then the 4 DB
            trace information bytes have the following meanings:

            Byte 1   Resource Manager index (cf sequence in the
                            start parameters)

            Bytes 2 and 3   Operation code of XA call, output hexadecimal
                            and platform-dependent (EBCDIC or ASCII),
                            e.g. in the BS2000 system: (EBCDIC) X'9697'
                            = C'op'
                            op = xa_open, en = xa_end etc.

            Byte 4   Return code of XA call, output hexadecimal
                            and platform-dependent
                            e.g. in the BS2000 system: FD = -3 =
                            XAER_RMERR

K072   There is a mismatch in the number of &STMT entries in ROOT and KDCFILE.

This message is output when it is recognized at the start of the application that the
number of programs and load modules in the ROOT and KDCFILE generation do
not match. The start of the application or the task is terminated with the message
K049, code 40.

Action: Regenerate ROOT and/or KDCFILE

K073 The attribute &ATTR of &STMT &PROG in ROOT and KDCFILE does not match.This message is output, if, at the start of a task it is established that the entries of the LOAD MODULE table or the PROGRAM table in ROOT and in the KDCFILE do not match. The first insert specifies the attribute where the entries differ. The start of the application or the task is terminated with the K049 message, code 41.

Action: Regenerate ROOT and/or KDCFILE

K074 Program exchange completed; &CTYP &PROG &PVER

K075 Program exchange aborted by task &TASK; &CTYP &PROG &PVER

K074 and K075 are issued after a positive or negative termination of a program exchange. The inserts give information as to which part of the application program were exchanged or which part could not be exchanged:

&TASK: TSN of the process for which program exchange was aborted.

&CTYP: APPL, Exchange of the entire application program
LMOD, Exchange of a load module

&PROG: Name of the load module (only when CTYP=LMOD)

&PVER: Version number of the load module (only when CTYP=LMOD)

K076 Error during asynchronous administration with transaction code &ADTC; KCRCCC=&RCCC, KCRCDC=&RCDC

K077 Internal cluster communication with &PTRM/&PRNM/&BCAP/&LTRM successful: &CLSIGT

The message K077 is only output if test mode is active (see).

During internal cluster communication, a &CLSIGT message was sent to another node application. &CLSIGT can have the following values:

&CLSIGT	Meaning
1	Check the availability of the monitored node application (check alive, ping)
2	Check for changes in the cluster configuration
4	Administration job
8	Job to wake up one of the processes waiting at a GSSB or ULS
16	Response to an availability query from the monitoring node application (check alive response, pong)

No action is required.

K078   &ERRNAME &ERRCODE: in &REA3

Message K078 is output by UTM in several different variants, depending on the event that has occurred. The events can be subdivided into the following categories:

–   IICS messages
–   Message on establishing connection to the DB system
–   Messages when loading or exchanging programs
–   STXIT messages
–   Message (warning) on processing of start parameters

IlCS messages

K078   WRONG SPL-VERSION (NO ILCS)

This means that an incorrect SPL version has been linked in ($\leq$ V1.5): SPL does not establish ILCS-compliant stacks.

Response: Link the program again with SPL version $\geq$ 2.0

K078   subcode: ERROR HANDLING IN module, RC = return code

This means that an internal error occurred when managing the ILCS stacks or on calling ILCS-internal functions. The return code is also output if ILCS functions are called.
The *subcode* allows you to identify the phase in which the error occurred:

subcode = IL01 - IL09: Error during initialization

subcode = IL20 - IL29: Error during start/end of TP

subcode = IL30 - IL39: Error during stack management

The following table describes the errors for which *subcode* describes a user error. In all other cases, you must write a problem report (PR) and contact systems support.

subcode	Modul	Return code output?	Precise meaning of &ERCD2
IL02	ITOINIT	yes	Error initializing ILCS. User error if ILCS code is executed before UTM start code. Response: set the ENTRY-POINT of the application to KDCRTST.
IL08	KDCCILCI	no	An incorrect ILCS version was linked in. Version 1.5 or higher is required.

K078   xxxx : UNRECOVERABLE ILCS ERROR OCCURRED - ILCS FUNCTION WITHIN
NON-ILCS PROGRAM UNIT.

xxxx = SEUB/SELB/ASID/AHID

Meaning:

This message is output on termination of a program unit run. In a program unit with
COMP=ASSEMB/COB1, an ILCS object (source module) was compiled by an
ILCS-compiler. However, the ILCS object changes one of the fields for which xxxx
stands in the central ILCS data area (IT0PCD). openUTM can therefore no longer
base itself on this data, and the application program is terminated.

Action:

You must either create an ILCS-capable program unit from your ASSEMB/COB1
program unit or adapt the ILCS-capable object so that it does not use the ILCS.

Variant of the message K078 for the verification of the Oracle interface that is linked
to the application program:

```
"K078 *BIND -- MISMATCH -- : in KDCRTDB:
     old ORACLE IUTMDB interface linked in XA environment"
"K078 *BIND -- MISMATCH -- : in KDCRTDB:
     start of application will be terminated."
```

Meaning:

When a UTM Oracle XA application is started, a check is performed to determine
whether the old IUTMDB interface has been linked by mistake. This check found
the (old) ORADBCON connection module for the IUTMDB interface in the appli-
cation program. openUTM terminates application start.

Action:

The application program must be linked again, see also Oracle® Database User's
Guide (BS2000/OSD).

Message on establishing connection to the DB system

K078   KDCRTDB ERROR : in SESAM : INSTALLATION-UNIT /
         VERSION = db-system / version
K078   GETINSP 00400013: in KDCRTDB: error accessing pathname for logical-id yyyyyy
K078   KDCRTDB ERROR : in KDCRTDB -
         START OF APPLICATION WILL BE TERMINATED!

Error during dynamic loading of the DB connection module if the connection module
was dynamically loaded with LIB=LOGICAL-ID (yyyyyy) via the IMON installation
path. The specified name yyyyyy is incorrect.

K078   *BIND 0C010600: in KDCRTDB ENTRYNAME = entryname
K078   *BIND ERROR---: in KDCRTDB: LOAD-LIBRARY = : library-name
K078   *BIND ERROR---: in KDCRTDB -
         START OF APPLICATION WILL BE TERMINATED!

Error during dynamic loading of the DB connection module if the connection module
was dynamically loaded with LIB=LOGICAL-ID (SYSLNK) via the IMON installation
path. The dynamically loadable module is not present in the library found via
LOGICAL-ID(SYSLNK).

K078   RTIMER event A0 : in KDCRTDI - event in KDCRTDB while connecting to database

The establishment of the connection to the database system is aborted during UTM
application start due to timer expiry (real-time). The task is then terminated.

The message is only output if the start parameter DB-CONNECT-TIME is set
(minimum = 60 sec., maximum = 3600 sec.). In DB-CONNECT-TIME, you specify
the timer which monitors the establishment of the connection to the DB system
during the start phase of a UTM application.

Messages when loading or exchanging programs

K078   xxxxxxxx yyyyyyyy: IN module-name text

This variant is output in the following situations in conjunction with programs and with common memory pools in class 6 memory in the following situations:
– with an application program generated with BLS, and when dynamically loading or exchanging individual programs with BLS
– when exchanging programs with KDCPROG

Meaning of the individual inserts:

xxxxxxxx

contains a short ID of the error which occurred (see table on page 249). For errors with xxxxxxxx = (KDCS00,LDRTnn) the field yyyyyyyy is not output.

yyyyyyyy

contains the return code of the BS2000 macro called. For further information, refer to the BS2000 manual "Executive Macros". This insert does not apply in the case of internal UTM error codes.

module-name

contains the module name of the module, in which this error occurred (for internal diagnostics)

text   contains additional information on certain errors; possible texts:

```
' WRONG OPCODE IN KDCRTLD: WRONG OPCODE AT CALL OF KDCRTLD '
' LMOD: STAT ERR IN KDCRTLD: LMOD: <name> / <version> '
```

These texts refer to internal errors in UTM.

```
' LINK ERR OF AREAS in KDCRTLD: FOR AREA: <areaname> '
```

In this case the specified area has not been linked in by the user (check generation and load module creation).

Action: Link application program correctly.

xxxxxxxxx	Error cause	Action
ASHARE	Error when loading a load module from memory pool. Further information is given by the insert ' LMOD: name / version '	See return code

xxxxxxxx	Error cause	Action
BIND	– Error when loading a load module. Further information is given by the insert ' LMOD: name / version ' or – Error when loading the linkage module to the DB system (see insert ' ENTRYNAME = ') – Error when dynamically loading the root table module (see inserts &object and &lib) – Error when dynamically loading the user-specific message module (see inserts &object and &lib)	See return code.
CSTAT	The status of an MP page cannot be changed to ' read only'. For further information see supplementary insert 'PAGENUMBER = '.	System error
DEQAR	Error when releasing a serialization ID.	See return code (pool name possibly locked; change UTM generation)
DMS	Error when interoperating with the DMS. For further information see supplementary insert 'FILENAME = ' .	See DMS error code
ENAMP	Error when creating a memory pool. SCOPE and NAME of the relevant memory pool are also specified.	See return code, e.g. pool name already in use by another task; change UTM generation
ENQAR	Error when accessing serialization ID.	See return code (e.g. DEQAR), change UTM generation where necessary and increase the value of MAX MP-WAIT.
ITABL	Error when determining addresses of shareable programs. May be caused by unresolved EXTRNS. For more information see supplementary insert ' ENTRYNAME = ' .	System error
JOINFOA	Error when querying the user entry.	See return code
KDCALME	Not possible to provide sufficient memory on request for KL6 memory (INSUFFICIENT MEMORY and number of bytes output).	An attempt may have been made to start the application in the lower address space (insufficient memory!).
KDCS00	CALL "KDCS" in START exit, in SHUT exit or in INPUT exit	Correct program unit.
LDRT00	TABLIB start parameter missing but no ROOT module linked	Adapt start parameter or definitively link ROOT

xxxxxxxxx	Error cause	Action
LDRT01	The version of the UTM macro library does not match that of the UTM modules or the message module.	Regenerate ROOT, link to current library SYSLNK.UTM.nnn or regenerate message module.
LDRT08	The parameter CATID was used even though the filebase operand was specified with catid.	Change start procedure
LDRT12	The root table module is to be dynamically loaded, but no load module has been generated.	Generate the application again (with at least one LOAD MODULE in the SYSLIB library) see ROOT parameter in the KDCDEF gener-ation.
LINK	Error when loading with LINK macro For further information see supplementary insert ' ENTRYNAME = ' .	Check UTM generation
MINF	The memory allocation table is not available.	System error
RELM	Error when releasing class 6 memory For further information see supplementary insert ' PAGENUMBER = '	System error
REQM	Error when allocating a memory pool.	System error
REQMP	Error when requesting class 6 memory. Error code X'0C' can signify overlapping with a memory pool allocated by DSSM and defined with SCOPE=GLOBAL	Check BS2000 generation, possibly reduce UTM application Release area with command: /SET-SUBSYSTEM-OPTIONS SUBSYSTEM-NAME= NON-PRIVILEGED
SETIC	The interval timer could not be set.	System error
STXIT	Execution of the STXIT call failed.	System error
TABLE	Error in macro TABLE	Check KDCSHARE module
TMODE	Error when querying task characteristics.	System error
UNBIND	Error when unloading a load module For further information see supplementary insert ' LMOD: name / version '	See return code
VSVI	Error in VSVI macro	See return code
VSVI1	Error when determining program addresses. For further information see insert ' FOR PROGRAM: name'	See return code
$APCB	Error when transferring the TU-PC to UTM-TPR, possible return codes: 02: $APCB is not supported in this BS2000 version otherwise: see macro return code	See return code UTM error See return code

STXIT messages

K078   STXIT 01 STXIT(ILCS_STXIT_CONTROL) started at xxxxxxxx with IW=yyyyyyyy

K078   STXIT 02 STXIT(STANDARD) started at xxxxxxxx with IW=yyyyyyyy

The messages "STXIT 01" and "STXIT 02" have the following purpose:

They are output by UTM when the UTM STXIT routines start, and they record that an STXIT event has occurred. These messages are output as of openUTM version 5.1.

STXIT 01	The STXIT routine was started for ILCS.
STXIT 02	The standard STXIT routine was started.

It is important for the diagnosis that the P counter of the STXIT routine is specified; a breakpoint can be set at this address, and the error can then be analyzed with AID.

An STXIT event generally causes the UTM service to abort.

STXIT events can occur that do not lead to the abortion of the service. This is the case when an STXIT event occurs, is passed to the ILCS runtime environment and handled there either by ILCS itself or a program unit, and the event is then not passed on to UTM.

If an STXIT event occurs due to the expiry of a timer (e.g. IW=A0), it is possible that the STXIT message is output without any STXIT routine being started since the program unit has already been terminated in the interim. You may therefore need to increase the timer value for this transaction code. The TAC can be taken from the K099 message which is subsequently output:

```
K099 STXIT occurred in TAC: tacname at date: xx-xx-xx time: xx:xx:xx
```

The date and time in the K099 message refer to the start of the program unit.

The event IW=90 (TERM) can occur when one of the application's processes termi-nates.

Action:

If the service aborts, analyze the dump created. If there is an error in UTM, store the documents and write a PR; otherwise, correct the error in the program unit.

If the service does not abort, the message merely indicates that the interruption is being handled by the runtime system or the program unit. There is no error.

Message (warning) on processing of start parameters

K078   WARNING : in KDCRTBF! Syntax-error in SYSPROT-Parameter!
       Default value used!

K078   WARNING : in KDCRTBF! SYSPROT-Parameter: prefix too long!
       Default value used!

In both cases an incorrect SYSPROT start parameter was detected and the default value is effective. You may need to improve the SYSPROT start parameter!

K078   WARNING : in KDCRTBF! No database generated, but startparameters given!

A start parameter was specified for a database system but no database was generated. Adapt the generation or the start parameter!

It is no longer possible to generate the connection to an Oracle database via the IUTMDB interface. The corresponding DATABASE statement is acknowledged with errors as early as the KDCDEF run. If, despite this, the application is started with start parameter ".ORACLE ..." then the start of the application is terminated.

The following K078 message is output:

```
K078 Error in start params : in UTM V6.3 Oracle is supported only via
XA-Interface. Task gets terminated.
```

K079 Accounting problem - reason: &REA2

When attempting to write accounting information, the BS2000 accounting routine reports an error. openUTM thus deactivates the accounting for the application. The application continues to run.

The error cause is given in the insert &REA2:

&REA2	Meaning
04	BS2000 accounting is not active
08	in the BS2000 accounting the corresponding record type is not active (record type UTMA or UTMK)
20	Error in BS2000 accounting: Slot shortage
28	Error in BS2000 accounting write task

If &REA2=X'08', openUTM deactivates the accounting phase, the calculation phase or both, depending on the record type.
In all other cases, openUTM deactivates both phases.

Action: Inform BS2000 system administrator. Once the error has been rectified, the UTM administrator can reactivate the costing and/or accounting phase with the KDCAPPL command.

K080 KDCMON is not active.

K081 Statistics: &IMSG1/&OMSG1/&CONU/&ATAC2/&LWRT/&HITR/&WTBF

Inserts	Meaning	
&IMSG1	Number of terminal input messages	1)
&OMSG1	Number of terminal output messages	1)
&CONU	Number of connected users	
&ATAC2	Number of unprocessed ATACs	1)
&LWRT	Number of log writes	
&HITR	Cache hit rate	1)
&WTBF	Cache waits for buffer	

This message is written every hour on the hour and in the case of normal termination of the application. The values marked with 1) are subsequently reset to 0.

K082 Wrong file &FNAM

K083 File &FNAM has been destroyed.

K084 &OBJ1 &VER1 &OST1 &BMD1 and &OBJ2 &VER2 &OST2 &BMD2 are not compatible.

K085 &FNKT functions are not available.

K086 UTM-D error information: &PTRM, &PRNM, &BCAP, &LTRM, &USER, &SYSD, &USSD, &FMH7, &AGUS

Inserts	Meaning
&PTRM	Name of LU6.1 connection (CON)
&PRNM	Processor name
&BCAP	BCAM application name
&LTRM	Name of LPAP parameter
&USER	User ID
&SYSD	SNA sense code (system)
&USSD	SNA sense code (user)
&FMH7	Error message from the remote application
&AGUS	ID of job-submitting user

The inserts &SYSD and &USSD can have the following values in the case of homogeneous interconnection:

&SYSD&USSD	Meaning
08120000	Resource bottleneck (e.g. page pool full)
08130000 081B0000	Contention: A session reservation request was rejected by the contention loser application because the contention winner application has reserved the session for a job.
08641003	Illegal or locked TAC
0864C5E2	PEND ER by openUTM
0864C5D9	PEND ER by application program unit
0864D9E2	PEND RS by openUTM
0864D9E4	PEND RS by application program unit
0866D9E2	PEND RS by openUTM
0866D9E4	PEND RS by application program unit
10030000	Illegal or locked TAC

K087   Error in interoperation of UTM and format handling system: opcode = &OPCD3, error code = &ERCD4 &ERCD5

&OPCD3	Meaning
STPA	Start parameter input
INIT	Initialize the formatting system
INPI	Create and enlarge the PI (Presentation Image)
ISOUT	Incorporate the output message section in the PI (in MPUT)
CRM	Generate output message (in PEND)
ISINP	Incorporate input message in the PI
RDMP	Read input message section from the PI (in MGET)
REST	Restart formatting for the screen restart
USER	User call to the formatting system
CUR	Cursor positioning
CRMA	Generate asynchronous output message

The insert &ERCD4 contains the error code KCRCDC of the form FHxx:

&ERCD4	Meaning
FH01	An error has occurred in the formatting. See supplementary return code from FHSCON or FHS in field KCRMF.
FH02	Hardware function not available.
FH04	Serious error during initialization of the formatting system FHS. openUTM aborts the start of the application.

The insert in &ERCD5 provides additional information on the errors that have occurred. You can find out the meaning of the supplementary information in the formatting system manual.

K088   UTM-D session start : &LSES/&RSES/&LPAP
SR-STATE: &SRFG PET &PSQN
SAVED : &ESQS &EBSS
ACT. : &ESQR &ESRR &EBSR

UTM-D session start.
The message contains diagnostic information.

Inserts	Meaning
&LSES	LSES name
&RSES	RSES name

Inserts	Meaning
&LPAP	LPAP name
&SRFG	Status bits from the LSES table entry (SR)
&PSQN	PET sequence numbers Two 2 byte numbers: – Sequence number sent saved – Sequence number sent current
&ESQS	Secured sequence numbers Two 2 byte numbers: – SLU TO PLU – PLU TO SLU
&EBSS	Secured bracket state – "BETB" = BETween Bracket – "INBR" = IN Bracket Receive (local side without send authorization) – "INBS" = IN Bracket Send (local side with send authorization)
&ESQR	Current sequence numbers (from STSN request) Total of 5 bytes: – SEC_TO_PRI_SQN (2 bytes) – PRI_TO_SEC_SQN (2 bytes) – ACTION_CODE_SEC_TO_PRI (2 bit)s 00: ignore 01: set7 10: test 11: set and test – ACTION_CODE_PRI_TO_SEC (2 bits) 00: ignore 01: set 10: test 11: set and test – RESERVED (4 bits)

Inserts	Meaning
&ESRR	Current sequence numbers RSP(STSN) bytes 5 Bytes: – SEC_TO_PRI_SQN (2 bytes) – PRI_TO_SEC_SQN (2 bytes) – ACTION_CODE_SEC_TO_PRI (2 bits) 00: ignore 01: set 10: test 11: set and test – ACTION_CODE_PRI_TO_SEC (2 bits) 00: ignore 01: set 10: test 11: set and test – RESERVED (4 bits)
&EBSR	Current bracket state – "BETB" = BETween Bracket – "INBR" = IN Bracket Receive (local side without send authorization) – "INBS" = IN Bracket Send (local side with send authorization)

K089   Request to delete asynchronous message (&GNDATE/&GNTIME) for destination
&DEST accepted.
&GNUSER / &USER / (&DLDATE/&DLTIME) / &CHAIN

&CHAIN	Meaning
NO	No negative follow-up message of the deleted message was present.
DEL	The negative follow-up messages should also be deleted.
ACT	The negative follow-up message should be activated, i.e. inserted in the message chain of its receiver.

Any positive follow-up message is deleted in any case when the job is executed.

K090   Request to delete all asynchronous messages for destination &DEST accepted.
&USER / (&DLDATE/&DLTIME)

K091   Due to a resource bottleneck no sign-on can be processed by application &BCAP
at this time.

K092 Please enter password
> &PAS1 <
If you wish to change the password, please enter new password
> &PAS2 <
and repeat new password
> &PAS3 <

K093 Service cannot be stacked.

See also KDCDEF control statement MAX, operand NRCONV.

K094 Sequence of unsuccessful sign-on attempts

This message triggers a silent alarm to call attention to a succession of unsuccessful sign-on attempts by a user or from a client. This can be dealt with in the application via the MSGTAC program unit for example. The number of unsuccessful attempts after which openUTM is to generate the message K094 can be selected using the statement SIGNON ...SILENT-ALARM = nnn of the KDCDEF utility routine.

The message contains the following inserts:

Inserts	Meaning
&PTRM	Terminal name
&PRNM	Processor name
&BCAP	Application name
&LTRM	Logical terminal name
&USER	User ID
&RCF1B	Reason for the rejection
&REA4	Originator of the rejection (L=LTERM, U=USER, B=BOTH)

The &USER insert contains the user ID of the last unsuccessful sign-on attempt of the sequence. This can be a user ID generated for the application or can be any character string transferred to UTM as user ID.

The three characters in the &RCF1B insert specify the reason for the rejection of the last sign-on attempt of the sequence. This insert is allocated by UTM like the KCRSIGN1 and KCRSIGN2 return fields of the KDCS call SIGN ST.

The &REA4 insert specifies, if the user (U), the client LTERM (L) or both (B) have caused the message. The counter for the unsuccessful sign-on attempts is reset to zero.

K095 Sign-off is effective - please enter <SEND>

K096 Return to interrupted service - please enter <SEND>

K097 Input for new password cannot be used - please sign on

K098 Input exit returned error information &RCF1C &RCF2B - input please

This message is generated when the INPUT exit is called up if errors are detected by UTM or by the exit itself.

Insert &RCF1C indicates what UTM discovered during or after the INPUT exit:

&RCF1C	Meaning
IN00	INPUT exit call okay from UTM standpoint
IN01	Error in calling up INPUT exit via IUTMHLL
IN02	KDCS call in INPUT exit
IN03	Output parameters do not match KCICCD
IN04	Illegal value in KCICCD
IN05	DB USER CALL in INPUT exit
IN06	STXIT in INPUT exit

Insert &RCF2B indicates what the INPUT exit has entered in the KCIERRCD output parameter in the case of KCICCD = 'ER'.

K099 &MSG

This message is output by UTM either when an STXIT event occurs or as a supplement to message K078. In the first case the insert &MSG contains:

```
STXIT occurred in TAC: tacname at date: yy-mm-dd time: hh:mm:ss
```

In the second case the insert &MSG contains additional return codes.

This message is also output as an information message in the case of generated memory pools. In this case, it provides information about the actual allocation of the memory pool by the shared components of the application program loaded in this pool. Two K099 messages are output: The first indicates the relevant memory pool and the second the level of occupancy by generated shared components if this pool is generated with write-protection.

K101   Resource bottleneck - please repeat last input

This message is output in response to any of the following events:

– If a dialog input message should be buffered in the page pool due to a TAC class waiting situation, and there is not enough room in the page pool

– If an input message destined for an asynchronous program, a TAC queue or a temporary queue is to be saved to the page pool and either warn level 2 has been exceeded or there is not enough room in the page pool

– If an input message is intended for an asynchronous program, a TAC queue or a temporary file, and the number of messages saved for this asynchronous TAC, this TAC queue or this temporary queue has already reached or exceeded the generated value of QLEV, and the temporary queues were generated with QMODE=STD

K102   Interrupt weight &CIW at program count &CPC &CDSPL.
Instruction length code : &CILC program mask : &CPM condition code : &CCC
AMODE : &CAMODE
R0 &CR0 R1 &CR1 R2 &CR2 R3 &CR3 R4 &CR4

The messages K102 and K103 are output together (see explanation of the message K103).

K103   R5 &CR5 R6 &CR6 R7 &CR7 R8 &CR8 R9 &CR9
R10 &CR10 R11 &CR11 R12 &CR12 R13 &CR13 R14 &CR14
R15 &CR15

Outputs the context when an STXIT occurs (see also the table in the).

Interrupt weight 48:
If the STXIT event 48 occurs with the program KDCS + 8, this means that an address that cannot be accessed was specified for the KDCS parameter area when the KDCS interface was called.
Action: Change the application program unit.

Interrupt weight 90:
If a termination attempt is made in a program unit (e.g. with the "exit (1)" statement in a C++ or C program), a number of runtime modules are called before openUTM receives control of the application program. For this reason, the most recently available register record is interpreted when the register record is displayed.
You can then find the module responsible for termination via stack chaining if you register 13 points to a valid save area.

K104 UTM-D &UTMDEVT (&RCVDANNO): &LSES , &LPAP , &AGUS
; old state: (&OCVST, &OTAST)
; action: &ACTION
; new state: (&NCVST, &NTAST).

This message is output by UTM-D in response to any of the following events:

During communications via LU6.1, the message is output on the following events:

● On the expiry of one of the following timers:

  – Session occupancy timer for a dialog job (see value time1 in the KDCDEF statement LTAC, parameter WAITTIME).
    In this case, the insert &LSES contains blanks.
    Exception: Timer expiry after allocation of a contention loser session.
    Special case:
    If the session occupancy timer of a contention winner session equals 0 and no session is free at the end of the program unit then this is handled in the same way as a timer expiry.

  – Response timer (see value *time2* in the KDCDEF statement LTAC, parameter WAITTIME).
    In this case; &LSES and &AGUS are assigned different names.

  – Prepare-to-Commit timer (see KDCDEF statement UTMD, parameter PTCTIME). In this case, &LSES and &AGUS are identical.

● If a connection is lost during a session used by

  – a service with an open transaction, or
  – by a service which has initiated the end of the transaction.

● In the case of a session restart, if the session is allocated to a service that has initiated the end of the transaction.

In the case of communication via OSI TP, the message is output when one of the following timers expires. &LSES always contains blanks.

● Association occupancy timer (for dialog jobs, see value *time1* in KCDEF statement LTAC, parameter WAITTIME, for asynchronous jobs, internal value of 60 seconds).
  In the case of an asynchronous job, &ACTION contains the value ASYNCH.

● Response timer (for dialog jobs, see *time2* in KCDEF statement LTAC, parameter WAITTIME, for asynchronous jobs see *time2* in KCDEF statement UTMD, parameter CONCTIME).

In the case of a dialog job, processing waits for the response and in the case of an asynchronous job, processing waits for an acknowledgment from the job receiver. In the case of an asynchronous job, &ACTION contains the value ASYNCH.

● PEND-KP and PGWT-KP timer (see KDCDEF statement MAX, parameter TERMWAIT or PGWTTIME).

● Ready timer (see KDCDEF statement UTMD with parameter PTCTIME or KDCDEF statement MAX with parameter PGWTTIME)

Inserts	Meaning
&UTMDEVT	Event during which the message was output:
	RESTART Session restart
	DISCON Connection loss
	TIMEOUT Timeout
&RCVDANNO	Last announcement received at the UTM exchange. In the case of &UTMDEVT = TIMEOUT, the first two bytes in &RCVDANNO have the following meaning:
	X'F330' X'F331' Timer expiry in OSI TP job-receiving service after sending a message to a job submitter in a transaction (see KDCDEF statement MAX, parameter TERMWAIT or PGWTTIME).
	X'F332' Timer expiry in OSI TP job-submitting service after all job receivers have been requested to initiate the end of the transaction. The timer corresponds to the greater of the generated values of *time* in the KDCDEF statement MAX PGWTTIME= and *time2* in the KDCDEF statement LTAC WAITTIME= of the participating LTACs.
	X'F333' Expiry of the ready timer in the OSI TP job-receiving service (KDCDEF statement UTMD, parameter PTCTIME or KDCDEF statement MAX, parameter PGWTTIME).
	X'F400' Expiry of the OSI TP association occupancy timer for a dialog job (*time1* in KDCDEF statement LTAC, parameter WAITTIME).
	X'F520' Expiry of the OSI TP association occupancy timer for an asynchronous job (internal 60-second timer).
	X'F522' Expiry of the timer which monitors the acknowledgment for a queued message sent via an OSI TP association (*time2* in KDCDEF statement UTMD, parameter CONCTIME).

Inserts	Meaning	
&RCVDANNO (cont.)	X'F534'	Expiry of the OSI TP response timer in the job-submitter service for a dialog service. The timer corresponds to the greater of the generated values of *time* in the KDCDEF statement MAX PGWTTIME= and *time2* in the KDCDEF statement LTAC WAITTIME= of the participating LTACs.
	X'F800'	Expiry of the LU6.1 session occupancy timer for a dialog job (*time1* in KDCDEF statement LTAC, parameter WAITTIME).
	X'F933'	Expiry of the Prepare-to-Commit timer in the LU6.1 job-receiving service (KDCDEF statement UTMD, parameter PTCTIME or KDCDEF statement MAX, parameter PGWTTIME).
	X'F934'	Expiry of the LU6.1 response timer in a dialog service (The timer corresponds to the greater of the generated values for *time* in the KDCDEF statement MAX PGWTTIME= and *time2* in the KDCDEF statement LTAC WAITTIME= of the participating LTACs).
	X'F935'	Expiry of the LU6.1 response timer in a dialog service after re-establishment of the LU6.1 session (The timer corresponds to the greater of the generated values for *time* in the KDCDEF statement MAX PGWTTIME= and *time2* in the KDCDEF statement LTAC WAITTIME= of the participating LTACs).
	X'F936'	Expiry of the LU6.1 response timer on service restart after the establishment of a session (*time2* in KDCDEF statement LTAC, parameter WAITTIME)
&AGUS	Name of the job submitter (user, session or association)	
&ACTION	Action taken by openUTM:	
	COMMIT	Terminate transaction
	RESET	Reset transaction
	WAIT	Nothing
	STPROG	Start of successor program
	ASYNCH	Renewed execution of asynchronous job.

Inserts	Meaning
&OCVST &OTAST &NCVST &NTAST	Service and transaction status of the job submitter specified in &AGUS (before and after the action specified in &ACTION). Possible values: O O Service and transaction are open. O P The service is open and has initiated the end of the trans-action. O C The service is open and the transaction terminated. C C The service and transaction have been terminated. O R The transaction was reset and the service remains open. Z R The transaction has been reset and the service has been terminated.

K105   UTM-D &SYST mismatch; &LSES , &LPAP , &AGUS

       &SYST contains the system with which a mismatch occurred:

       DB   Database
       LPAP   Other application

K106   Output message discarded

K107   Terminal type &TTYP is not supported by VTSU or connection letter is wrong.

K108   Error &KRRC during Kerberos authentication.

       The message contains the following inserts:

Inserts	Meaning
&KRLH	Length of the client or server principal
&KRPR	The first 100 characters of the client or server principal. If the length of the principal is less than 100 then the content of the remaining characters is undefined.
&KRRC	Return code

The return code KRRC is structured in accordance with the BS2000 standard header and has the following significance:

SC2	SC1	MC	Meaning
X'00'	X'00'	X'0000'	Function executed without error
	X'01' X'02'	X'0018'	The output area is not large enough to store the entire Kerberos principal.
		X'FFFE'	Parameter error.
		X'FFFF'	Function not available.
		X'FFFF'	Function not available.
	X'03'	X'FFFF'	Version not supported.
	X'20'	X'00FF'	Internal SECOS error. A SERSLOG entry has been created.
X'00' (cont.)	X'40'	X'0001'	The connection partner ticket is no longer valid. A valid client principal is present in the output area.
		X'0002'	The connection partner ticket is too old. The client is located in the output area.
		X'0003'	The server and client realm are different. However, this is forbidden in the system settings. The client is located in the output area.
		X'0014'	The key table does not contain an entry with the specified ENTRY-ID. No entry in the output area.
	X'41'	X'FFFF'	The SECOS subsystem is not loaded.
	X'42'	X'FFFF'	The SECOS subsystem is not linked.
	X'81'	X'FFFF'	The SECOS subsystem is not currently accessible.
	X'82'	X'FFFF'	The SECOS subsystem has been halted or stopped.
X'01'	X'40'	X'0008'	The connection partner's ticket is invalid. The server's principal is located in the output area.
		X'0009'	The connection partner's ticket is encrypted with an algorithm for which there is no key in the BS2000 system. The server's principal is located in the output area.
		X'000A'	There is no key in the BS2000 system for the server specified in the ticket. The server's principal is located in the output area.
		X'000B'	The key version specified in the ticket is not present in the BS2000 system . The server's principal is located in the output area.

SC2	SC1	MC	Meaning
X'02'	X'40'	X'0010'	Problem with VTSU-B. The possible causes are: – The connection partner does not support any Kerberos authentication. – The user has interrupted the authentication operation using a K key. No entry in the output area.
		X'0011'	The connection partner has supplied a ticket which could not be interpreted. No entry in the output area.
		X'0012'	The terminal emulation uses a protocol version which is not supported. No entry in the output area.
		X'0013'	The connection was interrupted during authentication. No entry in the output area.

K109 Error &KRRC during Kerberos authentication - please sign on

Inserts see message K108.

K110 Invalid Kerberos Principal (&KRRC) - please sign on

Inserts see message K108.

K111 Sign-on not possible because of missing Kerberos support - please sign on

K115 MUX protocol error: &PTRM/&PRNM/&BCAP/&LTRM
&SNPT/&SNPR/&SNLT &CCC &REA1
&ANNO

Inserts	Meaning
&PTRM	PTRM name of the transport connection
&PRNM	Processor name of the transport connection
&BCAP	BCAM application name
&LTRM	LTRM name of the transport connection
&SNPT	PTRM name of the session
&SNPR	Processor name of the session
&SNLT	LTRM name of the session
&CCC	Error context (see table below)
&REA1	Error cause
&ANNO	BCAM announcement

&CCC	Meaning
X'01'	The error can be assigned to an existing session for multiplex connection.
X'02'	Error on BIND request.
X'03'	The error cannot be assigned.

The insert &REA1 is only output for system diagnostics. The fields &SNPT, &SNLT and &SNPR can only be filled if the error can be assigned to a current session (&CCC=X'01') or if a session could be identified at the session setup (i.e. possibly with &CCC=X'02').

K116 MUX UNBIND: &PTRM/&PRNM/&BCAP/&LTRM
&SNPT/&SNPR/&SNLT/&USER &REA1

&REA1	Meaning
X'01'	The terminal user has entered the KDCOFF command.
X'02'	The terminal user has entered the KDCOFF BUT command.
X'03'	A UTM-internal event caused the session to be cleared down (e.g. timeout for the session, administration, ...).
X'04' thru X'07'	Protocol error
X'0E'	Message could not be read.

The insert &USER denotes the user to whom the session is assigned.

The other inserts have the same meaning as for message K115.

K117 MUX BIND rejected: &PTRM/&PRNM/&BCAP/&LTRM
&SNPT/&SNPR/&SNLT/&USER &REA1

&REA1	Meaning
X'01'	The specified interface version is not supported.
X'02'	The partner is responsible for sign-on handling. However, the sign-on data supplied was incomplete or incorrect.
X'03'	The maximum number of sessions is reserved, i.e. no further sessions can be reserved using this multiplex connection.
X'04'	The BIND request was rejected (general error code).
X'05'	The specified station is already signed on.
X'06'	The specified user is already signed on.
X'08'	The station information in the BIND is not consistent with the generation.
X'09'	An incorrect password is supplied with the BIND.
X'0A'	The BIND request could not be read (LETTER_LOST).
X'0B'	Protocol error.
X'0C'	The specified station is not supported by the communication partner (UTM: DTP station, no dialog station, PTYPE=APPLI, ANNOAMSG=N).
X'0D'	The specified station is not supported by the communication partner (UTM: PROTOCOL=N is generated).
X'0E'	Protocol error.
X'10'	Incorrect ID information with BIND.
X'11'	Invalid USER specified with BIND.
X'12'	No KSET for station.
X'13'	USER locked.
X'15'	Protocol error.
X'16'	Protocol error.
X'17'	Release pending, i.e. the session cleardown is in progress, but not yet completed.
X'18'	Transaction recovery - no sign-on for the user
X'1C'	Password is being checked
X'1D'	The new password is not sufficiently complex
X'1E'	Error in new password: new password = old password
X'1F'	The password is required
X'20'	KDCSIGN not permitted at present; user is already signed on

&REA1	Meaning
X'21'	No further users may sign on; the permitted number (MAX CONN-USERS) has been reached
X'22'	No new password specified
X'23'	Validity of password has lapsed
X'24'	The new password is too short
X'25'	Resource bottleneck
X'26'	Application is being terminated; SHUT WARN was issued
X'27'	The LTERM partner of the terminal does not have the correct key (KDCDEF statement) to continue the service
X'28'	After transfer of the password by KDCUPD, the password is no longer sufficiently complex or is too short
X'29'	PTERM name not known
X'2A'	Processor name not known
X'2B'	STATUS=OFF for this PTERM
X'2C'	STATUS=OFF for this LTERM
X'2D'	Connection cleardown in progress
X'2E'	The LTERM/connection is generated on a different BCAM application
X'2F'	Processor name not known and also not a LTERM pool name
X'30'	No LTERM pool exists for station type or PTERM name not known
X'31'	No further free entry exists in LTERM pool
X'32'	Terminal already connected to the LTERM pool
X'33'	The LTERM pool is generated on a different BCAM application
X'34'	Connection request rejected due to contention
X'35'	Connection request rejected by VTSU, e.g. because of unsupported PTYPE or incorrect connection letter
X'36'	BCAM letter length too small
X'37'	BCAM return code not in order

In the &USER insert, in some cases the USER specified in the BIND request is entered; otherwise, &USER contains blanks.

The meaning of the remaining inserts is the same as for message K115.

K119   OSI-TP error information: &OSLPAP, &USER, &TAC, &DIA1, &DIA2, &DIA3

The explanations in the tables below refer to (primitive) log elements at the XAP-TP interface. These data elements have the following meaning.

Data element	Meaning
APM_ALLOCATE_CNF	Response (confirmation) to an association's request for an OSI TP dialog. A negative response means that no association could be provided.
TP_BEGIN_DIALOGUE_REQ	Request to the partner to start an OSI TP dialog.
TP_BEGIN_DIALOGUE_IND	Indication that the partner wants to start an OSI TP dialog.
TP_BEGIN_DIALOGUE_CNF	Response to the request to start an OSI TP dialog. A negative response means that the partner has rejected the dialog.
TP_END_DIALOGUE_REQ	Request to the partner to terminate the OSI TP dialog.
TP_END_DIALOGUE_IND	Indication that the partner wants to end the OSI TP dialog.
TP_U_ABORT_IND	Indication that the remote TP Service User[1] has interrupted the OSI TP dialog.
TP_P_ABORT_IND	Indication that the remote TP Service Provider[2] has interrupted the OSI TP dialog.
TP_DATA_IND	Indication that data has been passed by the partner.
TP_U_ERROR_IND	Indication that the partner has reported an error.
TP_HANDSHAKE_IND	Indication of a synchronization (handshake) request by the partner.
APUTM_ABORT_REQ	Request to abort a connection.

[1] TP Service User: UTM application

[2] TP Service Provider: System component which provides the OSI TP service. In the case of openUTM, this is the XAP-TP component.

The insert &DIA1 contains the case for the output of the message K119:

&DIA1	Meaning
1	A negative APM_ALLOCATE_CNF was received.
2	A negative TP_BEGIN_DIALOGUE_CNF was received.
3	A TP_BEGIN_DIALOGUE_IND was rejected by openUTM.
4	A TP_U_ABORT_IND or TP_P_ABORT_IND was received.
5	A TP_U_ERROR_IND was received.
6	openUTM does not support reception of TP_HANDSHAKE_IND for a dialog message.

&DIA1	Meaning
7	openUTM does not support reception of TP_END_DIALOGUE_IND from the client for a dialog message. A dialog RTAC may have been generated for an asynchronous LTAC.
8	With the unchained transactions functional unit, openUTM requires that the transaction starts with a dialog.
9	The association was rejected with APUTM_ABORT_REQ.
10	Resource bottleneck in UTM

The inserts &DIA2 and &DIA3 contain the following additional information depending on the value of &DIA1:

&DIA1	&DIA2 / Meaning	&DIA3 / Meaning
1	Cause of rejection:	Diagnostic information:
	0 ACSE Service User 1 ACSE Service Provider	-1 No reason given 2 Application Context Name is not supported 3 Job submitter's Application Process Title (APT) is unknown 4 Job submitter's Application Identifier (APID) is unknown 5 Job submitter's Application Entity Qualifier (AEQ) is unknown 6 Job submitter's Application Entity Identifier (AEID) is unknown 7 Job receiver's Application Process Title (APT) is unknown 8 Job receiver's Application Identifier (APID) is unknown 9 Job receiver's Application Entity Qualifier (AEQ) is unknown 10 Job receiver's Application Entity Identifier (AEID) is unknown
	2 Presentation Service Provider	– Not relevant
	6 TP Service Provider	-1 No reason given. 0 CCR version 2 not available. 1 Incompatible TP protocol version 2 No contention winner association available. Possible cause: The number of associations in the partner application is smaller than in the local UTM application.

&DIA1	&DIA2 / Meaning		&DIA3 / Meaning	
1	7	Association Pool Manager	-1	No reason given.
			4	Association Pool Manager cannot construct any further associations.
			6	Local or remote Application Entity Title (AET) is unknown.
			7	No matching association found for the specified AET.
2	Cause of rejection:		Diagnostic information:	
	02	TP Service Provider	0	No reason given.
	03	TP Service User	1	TPSU title not known.
			2	TPSU title permanently unavailable.
			3	The possible reasons are: – TPSU title temporarily unavailable. – No free OSI TP instance table for the ACCESS-POINT is available in the partner application. – No free dialog table available. – There is another resource bottleneck in openUTM. Message K119 with &DIA1=10 at the job receiver contains more detailed information about the type of bottleneck. Action: Generate a larger number of associations for the ACCESS-POINT's OSI-LPAP partner at the partner application.

&DIA1	&DIA2 / Meaning	&DIA3 / Meaning

2 (cont.)	02 TP Service Provider
	03 TP Service User
		4 No TPSU title specified.
		5 Functional unit not supported.
		6 Combination of functional units not supported.
		7 Association reserved for partner.
		8 Partner AEI unknown.
		11 Functional Unit Shared Control not supported by openUTM.
		12 TPSU title cannot be decoded.
		13 Type of TPSU title not supported.
		14 TPSU title is too long.
		15 Decoding error for user data.
		16 Decoding error for security data.
		17 Abstract syntax (UTMSEC) unknown or not generated.
		18 User ID unknown or partner application generated without USER or user ID rejected, e.g. due to incorrect password.
		19 Partner not available, e.g. locked (STATUS=OFF) or due to QUIET command.
		
		20 User ID or password too long.
		21 Warning level 2 for page pool exceeded.
		22 The possible reasons are:
		  – Invalid transaction code
		  – Transaction code locked
		  – No administration rights.
		23 Transaction code not generated with CALL=FIRST.
		24 Asynchronous service cannot be started. The queue level generated for the asynchronous TAC (maximum number of messages in the message queue) has already been reached.
		25 Dialog service cannot be started because the transaction code has been generated as an asynchronous TAC (KDCDEF statement TAC TYPE=A).
		26 Combination of functional unit and restart functionality (KDCDISP) is not supported.

&DIA1	&DIA2 / Meaning	&DIA3 / Meaning
3	Cause of rejection:	Diagnostic information:
	3 TP Service User	04 No TPSU title specified 11 Functional Unit Shared Control not supported by openUTM. 12 TPSU title cannot be decoded. 13 Type of TPSU title not supported. 14 TPSU title is too long. 15 Decoding error for user data. 16 Decoding error for security data. 17 Abstract syntax (UTMSEC) unknown or not generated. 18 User ID unknown or partner application generated without USER or user ID rejected, e.g. due to incorrect password. For more detailed information, see the previous message K147. 19 Partner not available, e.g. locked (STATUS=OFF) or due to QUIET command. 20 User ID or password too long. 21 Warning level 2 for page pool exceeded.
4	Cause of rejection:	Diagnostic information:
	2 TP Service Provider 3 TP Service User	0 No reason given. 1 A permanent error has occurred. 2 TP_BEGIN_TRANSACTION_REQ rejected. 3 A temporary error has occurred. 4 A protocol error has occurred. 5 Collision of two TP_END_DIALOGUE_REQ primitives. 6 Collision of TP_BEGIN_TRANSACTION_REQ and TP_END_DIALOGUE_REQ.
9	Cause of rejection:	Diagnostic information:
	3 TP Service User	21 Warning level 2 for page pool reached. The queued message could not be saved. 31 The KDCFILE file is full: The message could not be saved. 32 Decoding error for the UDT data of a TP_DATA_IND primitive The association is interrupted in all cases.

&DIA1	&DIA2 / Meaning	&DIA3 / Meaning
10	Affected resources:	Diagnostic information:
	11 No free OSI TP instance table for this ACCESS-POINT is available in the application of the job-submitting service. The cause is a message to an OSI TP partner when all OSI TP instance tables are occupied. The message is entered in a queue. Action: Generate a larger number of associations for the access point's OSI-LPAP partners.[1]	Not relevant
	12 No free OSI TP instance table for this OSI-LPAP partner is available in the application of the job-submitting service. The cause is a message to the OSI TP partner when all OSI TP instance tables are occupied. The message is entered in a queue. Action: Generate a larger number of associations for the access point's OSI-LPAP partners.[1]	OSI-LPAP index
	13 The instance is occupied in XAPTP in the application of the job-submitting service. The message is entered in a queue. Action: Generate a larger number of associations for the access point's OSI-LPAP partners.[1]	UTM instance index
	14 In the application of the job-submitting service, an error occurred while reading the queued message to be sent. The message is entered in a queue.	Not relevant

&DIA1	&DIA2 / Meaning	&DIA3 / Meaning
10 (cont.)	15 No free OSI TP node table available. This error can occur in both an application of a job-submitting service and in an application of a job-receiving service. In an application of a job-submitting service, the message is entered in a queue. Action: Generate a larger number of associations for the OSI-LPAP partners.[1]	Not relevant
	16 There is no free OSI TP service available in the application of the job-receiving service. Action: Generate a larger number of associations for the OSI-LPAP partners.[1]	Not relevant
	17 There is no free OSI TP user table for the OSI-LPAP partner in the application of the job-receiving service. Possible cause: A dialog program unit whose association has been interrupted in the interim is still active for the OSI-LPAP partner.	OSI-LPAP index

[1] KDCDEF generation OSI-LPAP ASSOCIATIONS=

K120 Password expired

K121 Please enter password
> &PAS1 <
Your password is valid for &NUMDAYS more day(s) only!
Please enter new password
> &PAS2 <
and repeat new password
> &PAS3 <

K122 Your password is valid for &NUMDAYS more day(s) only

K123 LTERM does not have the rights to continue the service
 - please sign on

K124 Error: &RCXAPTP at startup of XAP-TP occurred in phase: &PHAXAPTP

This message is output as a diagnostic aid for errors which occur when starting XAP-TP.

Meaning of the inserts:

&PHAXAPTP Phase during start of XAP-TP. Possible values:
INIT (passing of the generation values to XAP-TP)
START/RECOVERY (OPEN and BIND of the instances and recovery of incomplete transactions).

&RCXAPTP Return code specifying the error in more detail. Depending on the phase (value of &PHAXAPTP), the return codes have different meanings. If the return code is not listed in the following table, the error is internal (in this event, you must write an error message).

&PHAXAPTP	&RCXAPTP	Meaning
INIT	18	The OSS version is lower than 4.0A20.
	20	KDCFILE and ROOT are not compatible. Possible cause: A new KDCFILE was created with KDCDEF (new: with OSI TP, old: without OSI TP); the UTM application was not relinked or the ROOT source was not recompiled.
	16 60 85	Not enough memory space could be allocated to cover the OSI TP-specific generation values.
	21	The KCSALME call to create the transfer buffer for the transfer syntaxes returned a bad return code.
	41	The "apext_init" sequence has already been executed.
	44	The KCSALME call to create the transfer buffer for the abstract syntaxes returned a bad return code.
	63	The KCSALME call to create the transfer buffer for the access points returned a bad return code.
	99	A function called by "apext_att" returned a bad return code. This return code can also mean that the address space that the OSS wants to allocate for its shared memory in a follow-up task is already being used as shared memory (e.g. by KAA, XAPTP global memory). Action: Try to start additional tasks.
	102	Error on attach. Action: Check whether an application is already signed on under this name at this host.
	106	XAPTP "apext_att" call Return code: APEXT_ATTACH_INVEREF Meaning: The waiting point reference specified in the OSS o_attach call is invalid or is not (or no longer) recognized. Possible cause: Follow-on task cannot be started because the application has already been terminated.

K125 Password not complex enough - contact administrator or sign off

The complexity of the password is inadequate. It does not correspond to the complexity level generated in the KDCDEF statement USER...,PROTECT-PW=(..,level,..).

Action: Administrator must assign new password

K126   SAT error information: &SATRC

The &SATRC insert contains the reason for outputting message K126.

The hexa characters of the return code (XXSSMMMM) have the following meaning (with SATCP V1.0A and later)

SS	Meaning
01 - 40	System error
41 - 42	SATCP not available
80 - 82	SAT function currently not executable

Additional return codes with SATCP V2.0A and later

SS	MMMM	Meaning
00	0001	SAT log record buffered
00	0101	SATCP in start phase
00	0102	SATCP in hold mode
00	0103	SATCP in shutdown phase
00	0104	Event not logged
00	0105	Event rejected by system exit routine #110

The BS2000 security officer should be informed if there are no system errors.

K127   Internal error in UTM - ROOT: &ERCD6

The &ERCD6 insert contains the reason for outputting message K127.

&ERCD6	Meaning
RT04	Errors in action index in KDCRTMM
LD01	Illegal load mode of load module
LD02	Illegal operation code in KDCRTLD call
LD03	Illegal load status of load module
LD04	Area table in Root is overwritten
LD05	Area in a load module that is not loaded (ONCALL)

K128   UTM-D job rejected: &CON/&PRNM/&BCAP/&LPAP &LSES &REA1 &RCDC &TAC

If the job is an OSI TP UTM-D job, the inserts have the following meanings:

&CON: OSI-CON name

&PRNM: Eight blanks

&BCAP   ACCESS-POINT name

&LPAP   OSI-LPAP name

The &REA1 insert contains the reason for outputting message K128.

&REA1	Meaning
X'01'	Illegal transaction code – LU6.1: DPN or PRN in FMH-5 or FMH-6 or the first 8 characters of the message – OSI TP: Recipient-tpsu-title in TP-BEGIN-DIALOGUE-RI &RCDC contains a – KCRCDC error code
X'02'	Transaction code not generated with CALL=FIRST – LU6.1: DPN or PRN in FMH-5 or FMH-6 or the first 8 characters of the message – OSI TP: Recipient-tpsu-title in TP-BEGIN-DIALOGUE-RI
X'03'	An asynchronous service should be started, TAC is generated with TYPE=D – LU6.1: Message with begin bracket and end bracket – OSI TP: TP-END-DIALOGUE-RI log element received
X'04'	A dialog service should be started, TAC is generated with TYPE=A – LU6.1: Message with begin bracket and change direction – OSI TP: TP-GRANT-CONTROL-RI or TP-HANDSHAKE-AND-GRANT CONTROL-RI received.
X'05'	An asynchronous service should be started, but the QLEV of the asynchronous is already reached. – LU6.1: Connection cleardown. – OSI TP: Connection cleardown.
X'06'	Only with a heterogeneous link via LU6.1: An asynchronous message was received with RQE. UTM expects RQD2. Response: NOCHECK and PROTECT must be specified in the start command for CICS.
X'07'	Only for heterogeneous connection using LU6.1: A message was received with EC and RQD2 but neither CD nor EB is set. Response: modify CICS or IMS program.

K129 &CMD - start parameter not allowed at this moment! Statement ignored

K130 Task priority &TPRIO not allowed for task &TASK! Priority not changed.

K132 &OBJ1 and &OBJ2 have not been generated by the same KDCDEF run

K133 Program for &EXIT missing!

K134 Message to &DEST has been placed into dead letter queue. &NMSG message(s) in dead letter queue

As soon as the generated threshold is reached, this message informs the administrator of asynchronous messages which have been sent to the dead letter queue after abnormal processing.

The inserts have the following meaning:

Insert	Meaning
&DEST	Original destination of the message which resulted in the dead letter queue threshold being reached.
&NMSG	Generated threshold value for the number of messages in the dead letter queue.

K135 UPIC message :
&PTRM/&PRNM/&BCAP/&LTRM/&UPCREAS/&UPCSTAT/&UPCPROT/&UPVENC1/&UPPENC2

A problem has occurred during interoperation with a UPIC partner.

The value of the insert &UPCREAS indicates the cause of the problem. The possible values for &UPCREAS and their meanings are described in the following table.

&UPCREAS	Meaning	Reason / Action
01	UPIC partner sends without send right	Error in UPIC or openUTM
02	UPIC partner sends protocol which is too short	Error in UPIC
03	UPIC partner uses unsupported protocol version	UPIC version incompatible with UTM version
04	UPIC partner sends invalid protocol	UPIC error
05	UPIC partner has initiated abortion of the conversation	Effect of the UPIC function Deallocate() (CMDEAL())
06	UPIC partner has not sent a TAC	UPIC error

&UPCREAS	Meaning	Reason / Action
07	The TAC sent by the UPIC partner: – is not generated – for which the LTERM/USER does not have key – is an administration TAC and USER is not an administrator – exists and program is missing	User error (Side info or SETTP)
08	UPIC partner has sent a TAC generated with CALL=NEXT	User error
09	UPIC partner has sent a TAC which is locked/deleted, or has too low encryption	User error (Side info or SETTP)
0A	UPIC partner has sent a TAC which is too short	UPIC error
0B	UPIC partner has sent a TAC when a service is open	UPIC error
0C	UPIC partner has sent a TAC for a continuation section	UPIC error
0D	UTM page pool is too small for input message	Page pool for application generated too small (statement MAX PGPOOL)
0E	UPIC partner grants only send right when a message is open	UPIC error
0F	UPIC partner grants send right when a message is open without closure of current message section	UPIC error
10	UPIC partner has sent a TAC generated with TYPE=A (asynchronous TAC)	User error
11	UPIC partner is assigned to an LTERM with STATUS=OFF (locked) on connection setup	Normal behavior
12	UPIC partner is assigned to an LTERM with STATUS=OFF (locked) on message receipt	Normal behavior Administrator has locked LTERM
13	UPIC partner has sent USER with invalid length	UPIC error
14	UPIC partner has sent invalid USER or PASSWORD	User error
15	UPIC partner has sent invalid data for service restart, e.g. KDCDISP without USER or with additional user data.	User error

&UPCREAS	Meaning	Reason / Action
16	UPIC partner has sent invalid data for service restart, e.g. KDCDISP for USER with RESTART=NO	User error
17	Invalid length of the protocol token received.	UPIC error
19	Invalid protocol token has been received.	UPIC error
1A	Overall length of the received data is inconsistent.	UPIC error or transport system error
1B	The protocol token for data was not the last protocol token.	UPIC error
1C	Two protocol tokens for the service were received.	UPIC error
1D	Two protocol tokens for the user ID were received.	UPIC error
1E	Two protocol tokens for the password were received.	UPIC error
1F	Two protocol tokens for the form were received.	UPIC error
20	Two protocol tokens for the function key were received.	UPIC error
21	No protocol tokens for data were received.	UPIC error
22	Inconsistent sign-on data was received.	UPIC error
23	A protocol token was received for a user ID, but not for a service.	UPIC error
24	Invalid function key.	UPIC error
25	Protocol token for a form was received in a data fragment.	UPIC error
26	Protocol token for a function was received in a data fragment.	UPIC error
27	Invalid encryption level	UPIC or UTM error
28	Length of the received user message is invalid	UPIC error
29	An error occurred during the protocol discussion	UPIC or UTM error
2A	Two protocol tokens were received for encrypted data	UPIC error

&UPCREAS	Meaning	Reason / Action
2B	Two protocol tokens were received for one encrypted password	UPIC error
2C	Two protocol tokens were received for one RSA key	UPIC error
2D	Two protocol tokens were received for one DES key	UPIC error
2E	Two protocol tokens were received for one cursor	UPIC error
2F	Two protocol tokens were received for one protocol discussion	UPIC error
30	No protocol tokens were received for encrypted data	UPIC error
31	One protocol token for one password and one encrypted password were received	UPIC error
32	An error occurred while decrypting the password	UPIC or UTM error
33	An error occurred while encrypting the password. The encrypted password is too long.	UPIC or UTM error
34	An error occurred while reading the DES key	UPIC or UTM error
35	The RSA key cannot be sent	UPIC error
36	An error occurred while reading the RSA key	UPIC error
37	An error occurred while encrypting the data	UPIC error
38	An error occurred while decrypting the data	UPIC or UTM error
39	Inconsistent message and conversation characteristics have appeared	UPIC error
3A	The protocol discussion could not be completed	UPIC error
3B	An error occurred while exchanging keys	UPIC error
3C	The UPIC partner does not support encryption	Normal behavior, change generation
3D	Two protocol tokens were received for the transaction status	UPIC error

&UPCREAS	Meaning	Reason / Action
3E	Two protocol tokens were received for the client context	UPIC error
3F	The length of the user message in the connection letter is not compatible with the protocol version	UPIC error
40	openUTM does not support encryption	Normal behavior, change generation
41	Two protocol tokens received for new password	UPIC error
42	Two protocol tokens received for encrypted new password	UPIC error
43	Protocol tokens received for new password and encrypted new password	UPIC error
44	Impermissible operation code occurred with the call of the internal function PASSWD_ENCRYPT_PROC to decrypt a password.	UPIC error
45	Two protocol tokens received for the client type.	UPIC error
46	No RSA key could be found.	Activate one of the RSA keys

The inserts &UPCSTAT and &UPCPROT are used for diagnosis in error situations. &UPCSTAT contains the UPIC state bits in the USRTNSR table. &UPCPROT contains the UPIC protocol.

Inserts &UPVENC1 and &UPPENC2 are used for diagnosing data encryption.

The first byte of &UPVENC1 contains the ENCRYPTION_LEVEL of the message, the second byte of &UPVENC1 contains the ENCRYPTION_LEVEL of the conversation.

The first byte of &UPPENC2 contains the ENCRYPTION_LEVEL of the session, the second byte of &UPPENC2 contains the ENCRYPTION_LEVEL of the partner.

K136 (First) SYSLOG file is &FNAM

openUTM outputs this message in the start phase. &FNAM contains the name of the SYSLOG file. If the SYSLOG is created as a file generation group, &FNAM then contains the name of the first file generation which is written by openUTM.

K137 SYSLOG switched to file &FNAM

openUTM has successfully switched to a new SYSLOG file generation. The switchover was initiated by the administration or by the automatic size monitoring facility. &FNAM contains the name of the new SYSLOG file generation.

K138 SYSLOG file &FNAM closed

Two situations should be differentiated:

– the SYSLOG is maintained as a single file:
The last UTM task of the application has closed the SYSLOG file. &FNAM contains the name of the SYSLOG file.

– the SYSLOG is maintained as a SYSLOG FGG:
A SYSLOG file generation was closed by the last UTM task (i.e. completely). This file generation is now freely available. It is no longer required by openUTM. &FNAM contains the name of the closed file generation.

K139 Switching SYSLOG failed! Still using file &FNAM

The attempt to switch to a new SYSLOG file generation has failed. openUTM continues working with the file generation &FNAM. It may be possible to ascertain the reason for the error occurring on switchover from the DMS error code in the preceding message K043.

K140 There is no supported MUX protocol version in the range from &MXP1 to &MXP2

K141 The MUX protocol version &MXP1 is not supported

K142 Relaese pending timeout for session. PTERM: &PTRM MUX-PTERM: &MXPT

K143 UTM-D: STSN sequence numbers response differ from request. Request: &STS1, &STS2 Response: &STS3, &STS4

K144 VTSU-B return code &VTRC &CBRC when processing asynchronous message with editprofile or format &FMTN and coded character set name &CCSN to LTERM = &LTRM

K145 Due to a transaction recovery no sign-on can be processed by user &USER at this time - please sign on

K146 Monitoring BCAM waiting time. OPCODE=&BCMOPCD, RTCODE= &BCMRTCD, standard header= &STDHEAD, TSN= &TASK, BCAM appliname= &BCAP

This message is produced when:

– openUTM activates or deactivates monitoring of the BCAM waiting time, or

– BCAM reports an error on reading the BCAM waiting time. In this case, openUTM generates a UTM dump as diagnostic documentation and continues processing without monitoring the BCAM waiting time.

The values in OPCODE and RTCODE have the following meanings:

OPCODE	Meaning
00000000	Activation of time monitoring
00000001	Deactivation of time monitoring
00000002	Read waiting time of a message

RTCODE	Meaning
00000000	Job executed by BCAM
00000001	Functions for time monitoring not available
00000002 or greater	BCAM reports error, cause in openUTM or BCAM

K147 Sign-on for &USRTYPE user &USER not successful.
&PTRM/&PRNM/&BCAP/&LTRM reason: U&REA7

&USRTYP contains the following values:
– CONNECTION at sign-on of the connection user ID of a connection to a UPIC or TS client
– CLIENT at sign-on of a genuine user ID via TS application, UPIC client or OSI TP partner.

The values in &REA7 have the following meanings:

&REA7	Meaning
U1	The specified USER does not exist.
U2	The specified USER is locked.
U3	Somebody has already signed on with this USER name.
U4	The "old" password specified is incorrect.
U5	The new password specified cannot be used.
U6	There is no card reader available.
U7	The card information is incorrect.

&REA7	Meaning
U8	It is not possible to sign on at the moment: – because of a resource bottleneck. – because the maximum number of users who can be signed on simultaneously has already been reached. – because a password could not be changed, since an inverse KDCDEF is currently running.
U9	Sign-on not possible due to lack of Kerberos support.
U10	The current LTERM partner is not authorized to continue the service.
U11	The password is no longer valid. The password must be changed by the administrator.
U12	The new password does not fulfill the requirements of the generated complexity level.
U13	The new password is too short.
U14	The password passed by KDCUPD does not fulfil the generated complexity requirements or is too short.
U15	A transaction restart is necessary for the specified user.
U16	The open service cannot be continued from this partner type.
U17	The administrator entered SHUT WARN; normal users can no longer sign on to the application (an administrator can still sign on).
U18	The encryption mechanism required for continuing the open service is not available on the connection
U19	The validity period of the password has expired. Because Grace Sign-On is generated, sign-on can be repeated by transferring a new password.
U22	The specified USER does not exist in the cluster user file.
U23	Someone else has already signed on to another node with this USER.
U24	It is currently not possible to sign on because the cluster user file could not be locked in the generated time (CLUSTER statement, FILE-LOCK-TIMER-SEC parameter, FILE-LOCK-RETRY parameter).
U25	It is not possible to sign on at this node application because the user has a service that is bound to another node application and may not be terminated.
U26	Sign-on rejected because the open service belonging to the user has a transaction in the PTC state but no service restart has been requested.

K149 Internal diagnostic information &DIA5

An event has occurred which is logged with diagnostic information. &DIA5 is the internal UTM diagnostic information.

K150   RSO error &RSORC &RSO2RC &RSOMSG when calling subsystem RSO with action &RSOOPC

The insert &RSOOPC describes the RSO call in which the error occurred and has the following meaning:

RSOOPC	Meaning
X'01'	Reserve Printer
X'02'	Print Message
X'03'	Release Printer
X'04'	Invalidate Bourse
X'05'	Receive Info: Information on the printer or printer jobs
X'06'	Receive OLTP interfaces version number

The inserts &RSORC, &RSO2RC and &RSOMSG describe the synchronous and asynchronous return code and the RSO error message. The RSO error message can be obtained from the RSO manual.

If the asynchronous return code is not X'00000000', then the error occurred during the execution of the preceding job. In this case, the operation code of the preceding call can be found in byte 4 and the return code in bytes 5 to 8 of the insert RSOANNO:

X'56'	undef	undef	Opcode	R e t u r n c o d e

If the synchronous return code is not X'00000000' then the error occurred in the current &RSOACT call. The content of the insert &RSOANNO is then irrelevant.

If both the synchronous and the asynchronous return codes are X'00000000' then the RSO error message indicates a problem.

The return code (SC2 SC1 MC) consists of the components subcode 2 (SC2, always X'00'), subcode 1 (SC1) and main code (MC).

SC1	MC	Meaning
X'00'	X'0000'	The function was executed without error.
X'01'	X'0003'	Function call with invalid action.
	X'0004'	Length of the RSO macro OLTPRQX passed in the FPUT/DPUT RP by the UTM program unit is incorrect.
	X'0005'	Version of the RSO macro OLTPRQX passed in the FPUT/DPUT RP by the UTM program unit is incorrect.
	X'0006'	Function/unit of the RSO macro OLTPRQX passed in the FPUT/DPUT RP by the UTM program unit is incorrect.

SC1	MC	Meaning
X'01'	X'0011'	Character set name (CCSNAME) incorrect
	X'000A'	RSO buffer size too small (see RSO command MODIFY-SPOOL-PARAMETER).
	X'FFFF'	Unit or function not supported.
X'02'	X'0006'	Printer in release process. It will soon be available.
	X'FFFF'	Function not available.
X'03'	X'FFFF'	Version not supported.
X'20'	X'0001'	An error was reported during a memory request.
	X'0003'	Printer is already reserved by another application (as of RSO V3.2A).
	X'0004'	This printer is already reserved.
	X'0005'	An error occurred when creating the OLTP controller.
	X'0009'	An error occurred when using the bourse interface.
	X'000F'	An error occurred when using the memory pool.
	X'0010'	An inconsistency in the internal tables was detected.
X'40'	X'0007'	The printer is currently being used.
	X'0012'	Character set name (CCSNAME) unknown
	X'0013'	XHCS error in RSO
	X'000B'	An error occurred on accessing the printer.
	X'000D'	UTM does not have any access rights for the device (see the MODIFY-PRINTER-OUTPUT command).
X'41'	X'0002'	Printer not reserved by UTM or not started (see RSO command START-PRINTER-OUTPUT).
	X'0006'	The printer is currently being released because the STOP-PRINTER-OUTPUT command was issued.
	X'0008'	Printer released internally as a result of an internal error (recovery).
	X'000C'	Printer cannot be reserved due to recovery.
	X'FFFF'	RSO subsystem not loaded.
X'42'	X'FFFF'	RSO subsystem not connected.
X'80'	X'000E'	An error occurred on accessing the printer.
X'81'	X'FFFF'	RSO subsystem temporarily unavailable.
X'82'	X'FFFF'	RSO subsystem halted or stopped.

K151 Run of inverse KDCDEF terminated. Return code: &IDEFRC &DMSE &FNAM

The insert &IDEFRC contains a 16-digit return code for the inverse KDCDEF. The return code comprises two 8-digit printable numbers.

The first 8 digits have the following meaning:

First 8 digits	Meaning
00000000	The inverse KDCDEF was terminated normally.
00000001	A file handling error occurred.
00000002	A memory bottleneck occurred.
00000003	No KDCDEF statements were generated.

Digits 9 - 16 are only relevant if the first 8 digits contain 00000001. In this event, digits 9 - 16 contain a more detailed description of the error which occurred.
In all other cases, digits 9 - 16 contain 00000000.

Digits 9 - 16	Meaning
00000001	Invalid name for an output file.
00000002	Output file could not be created.
00000003	Output file could not be opened.
00000004	Output file could not be written.
00000005	Output file already exists and is not of the type SAM.

The insert &DMSE contains the printable DMS error code if an error occurs during file handling and digits 9 - 16 of the insert &IDEFRC contain one of the values 00000002, 00000003 or 00000004.

The insert &FNAM contains the name of the file for which the error occurred.

K152 Heuristic report: &COND &MTYPE &OSLPAP &USER &LTAC &AAIS &AAID

The inserts have the following meaning

&COND	Meaning
MIX	The server reported data inconsistency. This is not possible with an asynchronous message to a UTM application.
HAZ	Possible data inconsistency. Connection to OSI TP job receiver (Subordinate) was lost after a PREPARE was sent and the OSI TP job submitter (Superior) had not carried out a backup. The transaction is rolled back on the job submitter and a asynchronous message is sent again. The transaction is also reset in a UTM job receiver, or an asynchronous message is rejected (immediately after a restart), so that the data is consistent.

&MTYPE	Meaning
DIAL	A dialog message was sent.
ASYN	An asynchronous message was sent.

&OSLPAP: Name of the partner application.

&USER: Name of the user who issued the job.

&LTAC: for an asynchronous message: LTAC name of the job for a dialog massage: TAC of the transaction for which inconsistency is possible

&AAIS: Size of the atomic action identifiers in bytes.

&AAID: contains the encoded atomic action identifier (up to 64 bytes).

K153 Info for FHS message: &PTRM/&PRNM/&BCAP/&LTRM/&USER

This message is generated if FHS requests openUTM to output a message to SYSOUT. The message allows you to assign the user and LTERM partner to the FHS message. This information can be required in the event of problems with OMNIS (with and without MUX connections) in order to identify where the problem originated.

K154 SOCKET message: &PTRM/&PRNM/&BCAP/&LTRM &TCPCL &TCPRC

The inserts of the message have the following meaning:

Insert	Meaning
&TCPCL	Internal socket function or internal socket event
&TCPRC	DIAGNOSTIC WORD

&TCPCL is output in printable form:

&TCPCL	Meaning
OPEN_EVENTGROUP	Open the event group
ATTACH IPv4	Create a socket for the address family AF_INET
ATTACH IPv6	Create a socket for the address family AF_INET6
ATTACH AMBIGUOUS	A socket with the same port number has already been generated.
EVENT	Pick up an event
CONNECTION_REQ	Request the establishment of a connection
CONNECTION_RSP	Confirm the establishment of a connection
DISCONNECT_REQ	Clear connection
DATA_REQ	Send data
DATA_IND	Receive data
GET_HOST_BY_ADDR	Determine computer name of an IPv4 address
GET_IPNODE_BY_ADDR	Determine host name corresponding to an IPv6 address
GET_HOST_BY_NAME	Determine IPv4 address of a computer
GET_IPNODE_BY_NAME	Determine IPv6 address corresponding to a host name

The values of the diagnostic word are 4 byte long and are specified in hexadecimal notation. They correspond to the return codes of the "YDBSOC" interface.

General return codes

The following return codes can occur at every call of the TCP/IP functions:

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	01	FF	FF	Invalid unit or function in the standard header
00	02	FF	FF	Requested function not available
00	03	FF	FF	Invalid version number in standard header
00	41	FF	FF	Subsystem not present or not started
00	20	00	30	Internal BCAM error
00	82	00	04	BCAM is not active
0C	40	02	1C	BCAM shutdown running
14	40	02	20	SOCKET of BCAM administrator closed
00	04	00	44	Parameter list not accessible

&TCPCL = OPEN_EVENTGROUP

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	OPEN-EVENTGROUP executed successfully
04	80	06	1C	No memory available for SUB-TCB
04	80	0C	1C	No memory available for layer 4 CB
04	80	10	1C	No memory available for local event group control block
04	80	11	1C	No memory available for global event group control block
04	80	14	1C	No memory available for event group name
08	40	01	1C	BCAM shutdown announced
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing
00	01	02	51	Event group already opened by this task
00	01	10	51	USERID of the owner of the event group could not be deter-mined
00	01	11	51	Specification of EVENTGROUP_NAME is not permitted for a TU user

&TCPCL = CLOSE_EVENTGROUP

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	01	00	00	CLOSE_EVENTGROUP executed successfully
04	80	0C	1C	No memory available for layer 4 CB
08	40	01	1C	BCAM shutdown announced
00	80	00	28	Event group cannot be closed at the moment (call can be repeated later)
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing
00	01	04	51	Task not connected to the event group
00	01	07	51	Invalid EVENTGROUP_ID
00	01	13	51	Event group cannot be closed at the moment because it is still being used

&TCPCL = ATTACH IPv4 or ATTACH IPv6

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	SOCKET executed successfully
00	01	33	08	Specified port number already being used
04	80	04	1C	No memory available for APPCB
04	80	06	1C	No memory available for SUB-TCB
04	80	09	1C	No memory available for ENACB
04	80	0C	1C	No memory available for layer 4 CB
04	80	0D	1C	No free ABID available
04	80	0E	1C	No free port number available
04	80	13	1C	No name server entry available
08	40	01	1C	BCAM shutdown announced
10	82	00	1C	Global limit value for the number of open SOCKETs reached
10	01	00	20	SOCKET already opened by this task
18	40	01	20	SOCKET is currently being forcibly closed by the BCAM administrator
28	82	00	20	Local task limit value for the number of open SOCKETs
00	80	00	28	SOCKET currently not possible (call can be repeated later)
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing
00	01	01	50	Unknown host
00	01	02	50	Host not active
00	01	04	51	Task not connected to the event group
00	01	07	51	Invalid EVENTGROUP_ID
00	01	10	53	USER_ID could not be determined

&TCPCL = DETACH

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	CLOSE_SOCKET executed successfully
00	01	03	08	SOCKET_ID not specified
04	08	0C	1C	No memory available for layer 4 CB
04	01	00	20	SOCKET not opened by this task
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing

&TCPCL = EVENT

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	GET-EVENT executed successfully
00	80	00	28	GET-EVENT currently cannot be executed (call can be repeated later)
00	01	04	51	Task not connected to the event group
00	01	07	51	Invalid EVENTGROUP_ID
00	80	09	51	No event occurred

&TCPCL = CONNECTION_REQ

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	CONNECT executed successfully
00	01	03	08	SOCKET_ID not specified
04	80	02	1C	No free transport reference available
04	80	03	1C	No memory available for ACONCB
04	80	0A	1C	No memory available for ADDRCB-P
04	80	0B	1C	No free CONNECTION_ID available
04	80	0C	1C	No memory available for layer 4 CB

Return code				Meaning
Subcode		Main code		
2	1	2	1	
04	80	0E	1C	No free port number available
04	80	15	1C	No memory available for EVOL
08	40	01	1C	BCAM shutdown announced
04	01	00	20	SOCKET not opened by this task
18	40	01	20	SOCKET currently being closed forcibly by the BCAM administrator
20	01	00	20	CONNECT not permitted for SOCKET
24	82	00	20	No further connections permitted for this SOCKET
08	01	00	24	Connection already established
0C	01	00	24	Connection is already being established
18	01	00	24	Partner not known
1C	82	00	24	Partner processor not known
1C	82	01	24	Partner processor not active
1C	82	04	24	Partner IP address not known
1C	82	06	24	Request to establish connection to broadcast address
30	01	00	24	SOCKET not authorized for connection establishment
40	01	05	24	Required interface functionality not supported
40	01	07	24	Interface functionality of partners does not match
40	01	08	24	Level 4 address not available
60	40	00	24	Port number already being used
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing
00	01	04	51	Task not connected to the event group
00	01	07	51	Invalid EVENTGROUP_ID

&TCPCL = CONNECTION_RSP

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	ACCEPT executed successfully
00	01	07	08	CONNECTION_ID not specified
04	80	0C	1C	No memory available for layer 4 CB
08	40	01	1C	BCAM shutdown announced
04	01	00	20	SOCKET not opened by this task
18	40	01	20	SOCKET currently being closed forcibly by the BCAM administrator
04	01	00	24	Invalid CONNECTION_ID
14	01	00	24	No CONNECTION request pending
40	01	05	24	Required interface functionality not supported
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND process.
00	01	04	51	Task not connected to the event group
00	01	07	51	Invalid EVENTGROUP_ID

&TCPCL = DISCONNECT_REQ

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	CLOSE-CONNECTION executed successfully
00	01	07	08	CONNECTION_ID not specified
04	80	0C	1C	No memory available for layer 4 CB
04	01	00	20	SOCKET not opened by this task
18	40	01	20	SOCKET currently being closed forcibly by the BCAM administrator
04	01	00	24	Invalid CONNECTION_ID
64	40	00	24	Connection already being cleared down
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND process.

&TCPCL = DATA_REQ

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	SEND executed successfully
14	00	00	00	SEND executed successfully, but further sending prevented by interface flow control
00	01	07	08	CONNECTION_ID not specified
00	01	26	08	LENGTH_OF_DATA invalid
00	01	13	08	User buffer length = 0
00	01	31	08	User buffer not accessible
00	01	00	14	User data facility too large
04	80	01	1C	No memory available for data buffer
04	80	07	1C	Sending prevented by interface flow control
14	40	00	1C	Waiting for DATA_GO_INDICATION
04	01	00	20	SOCKET not opened by this task
04	01	00	24	Invalid CONNECTION_ID
48	01	00	24	Connection is not in the data transfer phase (not yet fully established)
00	80	00	28	SEND currently cannot be executed (call can be repeated later)

&TCPCL = DATA_IND

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	READ executed successfully
00	01	07	08	CONNECTION_ID not specified
00	01	26	08	length_of_data invalid
00	01	2D	08	No data available
00	01	31	08	User buffer cannot be accessed
00	40	00	10	No data received
04	01	00	20	SOCKET not opened by this task
04	01	00	24	Invalid CONNECTION_ID
48	01	00	24	Connection is not in the data transfer phase (not yet fully established)
00	80	00	28	READ currently cannot be executed (call can be repeated later)
00	40	00	2C	User data lost

&TCPCL = GET_HOST_BY_ADDR

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	GET_HOST_BY_ADDR executed successfully
04	80	0C	1C	No memory available for layer 4 CB
08	40	01	1C	BCAM shutdown announced
00	80	00	28	GET_HOST_BY_ADDR currently not possible (call can be repeated later)
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing
00	01	01	53	Invalid IP address

&TCPCL = GET_HOST_BY_NAME

Return code				Meaning
Subcode		Main code		
2	1	2	1	
00	00	00	00	GET_HOST_BY_NAME executed successfully
00	01	30	08	Invalid socket host name
04	80	0C	1C	No memory available for layer 4 CB
08	40	01	1C	BCAM shutdown announced
00	80	00	28	GET_HOST_BY_NAME currently not possible (call can be repeated later)
00	20	01	30	System error at startup of CONHAND processing
00	20	02	30	System error waiting for the termination of CONHAND processing

&TCPCL = GET_IPNODE_BY_ADDR

Return code				Meaning
Subcode		Maincode		
2	1	2	1	
00	00	00	00	GET_IPNODE_BY_ADDR performed successfully
04	80	0C	1C	No memory available for Layer 4 CB
08	40	01	1C	BCAM shutdown announced
00	80	00	28	GET_IPNODE_BY_ADDR not currently possible (repeat call later)
00	20	01	30	System error on starting CONHAND processing
00	20	02	30	System error on waiting for termination of CONHAND processing
00	01	01	53	Invalid IPv6 address

&TCPCL = GET_IPNODE_BY_NAME

Return code				Meaning
Subcode		Maincode		
2	1	2	1	
00	00	00	00	GET_IPNODE_BY_NAME performed successfully
00	01	30	08	Invalid socket host name
04	80	0C	1C	No memory available for Layer 4 CB
08	40	01	1C	BCAM shutdown announced
00	80	00	28	GET_IPNODE_BY_NAME not currently possible (repeat call later)
00	20	01	30	System error on starting CONHAND processing
00	20	02	30	System error on waiting for termination of CONHAND processing

K155   Your password is expired!
Please enter expired password
> &PAS1 <
Please enter new password
> &PAS2 <
and repeat new password
> &PAS3 <

K156   UTM-UPIC encryption function available: &RSLT

K157 Error &STDRC &SSORC when calling subsystem SRPMSSO

The &STDRC insert describes the return code in the standard BS2000 header of the function called.

The return code is structured in accordance with the standard BS2000 header and has the following meaning:

SC2	SC1	MC	Meaning
X'00'	X'00'	X'0000'	Function executed and information available for the connection
		X'0001'	Function executed and no information available for the connection
		X'0002'	Parameter error
		X'0003'	Internal error
	X'01'	X'FFFF'	Function not supported
	X'02'	X'FFFF'	Function not available
	X'03'	X'FFFF'	Version not supported
	X'41'	X'FFFF'	Subsystem SRPMSSO not loaded
	X'42'	X'FFFF'	Subsystem SRPMSSO not connected
	X'81'	X'FFFF'	Subsystem SRPMSSO temporarily not available
	X'82'	X'FFFF'	Subsystem SRPMSSO halted or stopped

The SSORC insert describes the internal return code.

Value and meaning of the &SSORC return code:

Value	Meaning
X'00'	Information supplied by SPRMSSO for the specified connection
X'01'	No information supplied by SPRMSSO for the specified connection
X'02'	The SRPMSSO call returned an error
X'03'	The SRPMSSO call returned a serious error
X'04'	The SRPMSSO call returned a parameter name that is too long
X'05'	The SRPMSSO call returned an invalid connection type
X'06'	The SRPMSSO call returned an invalid connection ID

K158   Inconsistent value for CPU time used: &CPUTEXT

The calculation of the CPU time used by the user resulted in an inconsistent value.

The message contains the following inserts:

Insert	Meaning
PTRM	Name of the current PTERM
PRNM	Name of the current processor
BCAP	Name of the current BCAM application
LTRM	Name of the current LTERM
USER	Name of the current user
CPUTEXT	NEGATIVE: A negative value was observed OVERFLOW: Too much time used by the user
CPUBEGIN	CPU time used at the beginning of the period monitored
CPUEND	CPU time used at the end of the period monitored
CPUUSED	CPU time used calculated for the period monitored
CPUCLNT	CPU time actually used by the current user
CPUREAS	Internal diagnostic value

K159   Password for User &USER changed

The password has been changed for the specified user. The message is sent every time the password is changed independently of how the password was changed (administrator, via SIGN CP, etc.)

The message possesses the following inserts:

Insert	Meaning
USER	Name of the user whose password was changed.
ENCPW	Modified password in encrypted form.

This message can be used to transfer modified user passwords to a standby application. To this end, the message destination MSGTAC or USER-DEST must be defined for this message in the private message module. The new user password can be set in encrypted form in the standby application by defining pw_encrypted='Y' via programmed administration KC_MODIFY_OBJECT, object type USER.

Notes:

– You should note that no UTM-D communication is permitted in the MSGTAC program unit. The modified password cannot therefore be sent directly from the MSGTAC program unit via a UTM-D connection to the standby application.

– If the application terminates during productive operation and password changes are made during this period then it is possible that the corresponding messages to the MSGTAC program will not be received but will instead be lost.

– UTM generates the new message for each type of password change and in particular if the password is modified in the standby application. You should make sure that the standby application does not transfer the password back to the productive application in order to avoid any ping-pong effect.

– You should note that UTM uses different encryption algorithms on different platforms. As a result, the function cannot be used on a cross-platform basis.

– In UTM(BS2000), the same host code should be used on the participating hosts so that special characters in passwords always have the same EBCDIC code.

– The passwords should be generated with the same level of complexity in the involved applications since UTM does not check the complexity when entering the encrypted password.

K160 The &TACNTR. transaction of service &TCVG has been rolled back by
&RBCAUSER (&RCCC/&RCDC)

The message has the following inserts:

Insert	Meaning
PTRM	Name of the current PTERM
PRNM	Name of the current processor
BCAP	Name of the current application
LTRM	Name of the current LTERM
USER	Name of the current user
TCVG	Name of the service TAC in which the transaction was reset
TAC	Name of the TAC in which the transaction was reset
TACNTR	Number of the reset transaction within the service
RBCAUSER	Initiator of the reset. There are the following values for the initiator, – User PEND RS: Reset was triggered by a PEND RS in the program unit. – System PEND RS: Reset was triggered by a PEND RS initiated by UTM. – User PGWT RB: Reset by triggered by a PGWT RB in the program unit. – System PGWT RB: Reset was triggered by a PGWT RB initiated by UTM. – User PEND ER: Reset was triggered by a PEND ER/FR in the program unit. – System PEND ER: Reset was triggered by a PEND ER initiated by UTM. – DBSTATUS ERROR: Error when querying the database status during node recovery. – DBSTATUS OPEN: The reset was triggered during the warm start due to an unsuccessful database status query. – DB CONFIRM PTC: Error on XA database call CONFIRM PREPARE TO COMMIT STATE during node recovery. – DB CATA VTV: Error on database call CANCEL TRANSACTION during node recovery. – JOURNAL: The reset was triggered during the warm start due to an incompletely written journal record.
RCCC	Value of the compatible KCRCCC return code
RCDC	Value of the incompatible KCRCDC return code

K161 Synchronous Periodic Write started for task &TASK

K162 Long IO for task &TASK: &IOPG pages, &IOMS milliseconds

K163 Long Periodic Write for task &TASK: &IOPG pages, &IOMS milliseconds

K164 System file SYSOUT will be switched to file &FNAM.

K165 System file SYSLST will be switched to file &FNAM.

K166 Error &RCHX when switching system file &SYSFILE to file &FNAM.

K167 System files SYSOUT und SYSLST successfully switched.

If the UTM application was started in a dialog then message K167 confirms the successful switching of the SYSLST files for all the application's tasks. SYSOUT continues to be assigned to the terminals unchanged.

K169 action: &ACTION; node: &NNM1/&HST1/&STATE; node monitoring this node: &NNM2/&HST2; node monitored by this node: &NNM3/&HST3

The message is output when a node application signs on or off at a cluster or when a node application fails. It provides information during the monitoring relationships in the UTM cluster application.

The message possesses the following inserts:

Insert	Meaning
&ACTION	Describes the situation in which the message is output. It contains one of the following values: – ADD – REMOVE – CHANGE – FAIL
&NNM1, &NNM2, &NNM3	Contain the node names.
&HST1, &HST2, &HST3	Contain the computer names.
&STATE	Contains the status of the node application that has been added or removed. If the node application has been added then the old status of this node application is output; if it has been removed then its new status is output. The insert &STATE can have the following values: G: generated Node application has not yet been started R: registered Node application is active T: terminated Node application was terminated normally A: abterm Node application was terminated abnormally F: failure Node is marked as failed

K170   Cluster: new KDCFILE detected; generation time: &DTTM, state: &STATE

The message is output if the generation time reveals a newly generated KDCFILE which is defined as the basis for future starts of application instances.

The message possesses the following inserts:

Insert	Meaning
&DTTM	Generation time of the new KDCFILE
&STATE	Start type of the local application. The insert &STATE can have the following values: G: generated Node application has not yet been started R: registered Node application is active T: terminated Node application was terminated normally A: abterm Node application was terminated abnormally F: failure Node is marked as failed

K171   Cluster: Node &HST1 is not a member of a XCS group or is member of an other XCS group (&XCS1,&XCS2).

The insert &XCS1 contains the name of the XCS cluster for node &HST1. If &XCS1 contains blanks then the node &HST1 does not belong to any XCS cluster.

The insert &XCS2 contains the name of the XCS cluster corresponding to the cluster. If the XCS cluster is not active then the XCS status is returned in the form Status: n in &XCS2. The values for n have the following meanings;

1:      XCS_INACTIVE
3:      XCS_RECONF_JOIN
4:      XCS_RECONF_LEAVE
5:      XCS_RECONF_FAIL
6:      XCS_INTERRUPTED
7:      XCS_SUSPENDED

If &XCS2 contains blanks then the name of the XCS cluster has not yet been entered in the cluster configuration file since no node applications are running yet.

K174 Configuration mismatch: &DIA1

The configuration of the starting node application does not match the configuration of the running UTM cluster application.

The insert &DIA1 describes the cause of the error:

– Values $\leq$ 10: Runtime configuration error
– Values > 10: KDCDEF configuration error

Values of the insert &DIA1 and their meanings:

&DIA1	Meaning
1	Mixed system log file SYSLOG as single file and file generation group (FGG)
2	The encryption capability of the starting node application is different from that of the running node applications.
11	Mixture of UTM-F and UTM-S applications (MAX APPLIMODE)
12	Mixture of single and duplicate file maintenance (second parameter in MAX KDCFILE, MAX USLOG)
13	Mixture of applications with and without users
14	Mixture of applications with and without multiple sign-on permission (SIGNON MULTI-SIGNON)
15	Mixture of applications with and without formatting system
16	The number of database systems has been modified (DATABASE, RMXA).
17	The sequence of database statements (DATABASE) has been modified.
18	The password history has been modified (SIGNON PW-HISTORY)
19	The number of LSSBs has been changed (MAX LSSB)
20	The number of GSSBs has been changed (MAX GSSB)
21	The number of ULS has been changed (ULS)
22	A ULS from the old KDCFILE no longer exists in the new KDCFILE.
23	The maximum number of services that a user is permitted to stack has been reduced (MAX NRCONV)
24	The maximum number of asynchronous services that may be open simultaneously has been reduced (2nd value in MAX ASYNTASKS)
25	The maximum length of the card information has been reduced (MAX CARDLTH)
26	The maximum length of the Kerberos principal has been reduced (MAX PRINCIPAL-LTH)
27	The size of the page pool has been reduced (1st parameter value in MAX PGPOOL)
28	The size of the process-specific buffer for the restart data has been reduced (2nd parameter value in MAX RECBUF)

&DIA1	Meaning
29	The length of the communication area has been changed (MAX KB)
30	The length of the default primary work area has been reduced (MAX SPAB)
31	The length of the message area has been reduced (MAX KB)
32	The maximum length of physical output messages has been changed (MAX TRMSGLTH)
33	The maximum length of the user data in LPUT records has been reduced (MAX LPUTLTH)
34	The value of the parameter ABORT-BOUND-SERVICE in the CLUSTER statement has been changed.

K175 File &FNAM created

K176 Procedure &PRCN started. &MSG2 RC = &RCHX

The configured procedure was started on the recognition of a node application failure.

The insert &PRCN contains the started procedure including the passed parameters.

The insert &MSG2 contains additional return information. In the BS2000 system, &MSG2 contains the printable return code of the CMD macro.

Values of the insert &RCHX1 and their meanings:

&RCHX1	Meaning
0	The procedure could not be started successfully
-1	The procedure could not be started. &MSG2 contains additional information.

K178 Cluster journal files: action &ACTION; states (&STATE,&STA2); used pages (&PGS1,&PGS2); number switches &SWNR

Diagnostic information is output for the two files in the administration journal.

The message possesses the following inserts:

Insert	Meaning
&ACTION	Action of the administration journal:
	CHANGE The administration journal file to which data is not currently written has been fully incorporated by all running node applications and an online copy of the administration data has been created; the entries in the file are no longer needed.
	CREATE The administration journal files have been (re)generated. This is performed on the start of the first node application if the node application was previously regenerated.
	EXTEND The administration journal file to which data is currently written has been extended because the other administration journal file has not yet been incorporated by all running node applications or the online copy of the administration data has not yet been completed.
	OPEN The administration journal files have been opened. Output on the start of every UTM process.
	SWITCH The administration journal file to which data is currently written has been switched.
&STATE / &STA2	Status of an administration journal file: At any time, only one file may have the status C; the other file must have the status O or R .
	C (Current) This is the file to which data is currently written.
	O (Old) This file is still being incorporated by running node applications or the online copy of the administration data is not yet terminated.
	R (Reusable) The data in this file is no longer required and can be overwritten after the administration journal files have been switched.
&PGS1 / &PGS2	Valid pages of an administration journal file: Outputs the number of logically valid pages of an administration journal file: When the file to which data is currently written is switched, it contains only one valid page: the first page containing the control information. Other pages containing old, invalid journal records are not physically released but continue to be occupied.
&SWNR	Number of times the administration journal is switched (ACTION SWITCH)

K179 Data to import completely imported by Online Import

K180   Data to import partially imported by Online Import

K181   No data to import existing for Online Import in file &FNAM.

K182   Abnormal termination of Online Import from file &FNAM.

K183   Normal termination of Online Import from file &FNAM.

K184   There is already an Online Import or a run of KDCUPD active for the old KDCFILE.

K185   The Online Import states that the generation times of the KDCFILEs are not equal

K186   The source application is not part of the cluster application.

K187   Start of Online Import from file &FNAM.

K188   Journal creation time is different. In &OBJ1: &DTTM; in &OBJ2: &DTM2

In the case of UTM cluster applications, this warning is output for diagnostic purposes if different administration journal generation times have been found in files.

The message possesses the following inserts:

Insert	Meaning	
&OBJ1 / &OBJ2	File ID.	
	jkaa	JKAA file
	journal	JRN1/JRN2 file
	kdcfile	KDCFILE
&DTTM / &DTM	Journal generation time found in file 1 or file 2.	

K189   Signon rejected - service restart in node application on host &HST1 required

K190 Diagnostic information &DIA1 for cluster &SUFF-file: &INF1: &INF2

The insert &SUFF designates the file to which the message refers.

The insert &DIA1 describes the cause of the error; the inserts &INF1 and &INF2 provide additional information.

The **Grp.** (Group) column in the tables below describes the reason group to which the error code belongs. The following groups exist:

A   The cause is a user error, e.g. an error in
        – generating and administering UTM applications
        – operating UTM applications
        – generating the system (e.g. division of the address space)

U   The cause is an error in the UTM code.

S   The cause is an error in another system component (software or hardware).

M   The cause is a memory bottleneck.

I   The message is for information only.

Message K190 for the cluster configuration file

If &DIA1 is in the range 1000 to 1099, then &SUFF has the value 'CFG' and the message refers to the cluster configuration file.

The table below provides an overview of the values of the insert &DIA1 for the cluster configuration file and their meanings, together with the values and meanings of the inserts &INF1 and &INF2 which may occur in combination with &DIA1:

&DIA1	Grp.	Meaning	&INF1	&INF2
1001	A	APPLNAME_MISMATCH Different application names in the cluster configuration file and KDCFILE.	Application name from the cluster configuration file	Application name from the KDCFILE
1002	A	HOSTNAME_NOT_FOUND The host computer name was not found in the cluster configuration file.	Local host name	Host name of the cluster node or blank
1003	A	INVALID_KDCDEF_TIME The generation time of the KDCFILE is too old; another node application with a more recent KDCFILE has already been started.	KDCDEF time from the cluster configuration file	KDCDEF time from the KDCFILE

&DIA1	Grp.	Meaning	&INF1	&INF2
1004	I	OLD_STATE_INVALID (warning) The old status of the node is "registered" and the start was not a warm start.	G generated: The node has never been started. R registered: The node is active. T terminated: The node was terminated normally. A abterm: The node was terminated abnormally. F failure: The node is marked as failed.	D: Start after KDCDEF run C: Cold start W: Warm start U: Start after KDCUPD run
1006	A	CLUSTER_FILE_CHANGED The cluster configuration file was regenerated during the application run.	Generation time from the cluster configuration file or KDCFILE filebase from the cluster configuration file.	Generation time from the KDCFILE or KDCFILE filebase from the KDCFILE.
1008	U	NODE_NOT_REGISTERED The local application is not entered as registered in the KAA.		
1010	AS	NODE_START_SERIALIZATION The lock for the serialization of the start of different node applications could not be initialized or could not be set.	INIT: Error when initializing lock LOCK: Error when setting lock	1: Timeout 2: Locked 3: Try later
1014	A	INVALID_FILE_VERSION The file version of the cluster configuration file does not match the UTM system code.	Expected version of the cluster configuration file.	Actual version of the cluster configuration file.
1015	ASU	CLUSTER_FILE_CORRUPTED The start and end markers of the cluster configuration file could not be found; the file is probably not a cluster configuration file.		

&DIA1	Grp.	Meaning	&INF1	&INF2
1016	A	NODE_KDCFILE_MISSMATCH A node application has already been started on another host computer with this KDCFILE.	Name from the cluster configuration file of the host computer on which a startup has already been performed using this KDCFILE.	Name from the cluster configuration file of the host computer on which a startup is now to be performed using this KDCFILE.
1017	A	NODE_GT_KDCFILE_START The node application was not last started with this KDCFILE.	Time of the last start with this KDCFILE.	Time of the last start of this node application.
1018	A	NODE_START_GT_KDCFILE_DEF The newly generated KDCFILE was generated before the last start of the node application.	Generation time of the KDCFILE.	Time of the last start of this node application.
1019	A	NODE_NAME_OF_OWN_HOST Node recovery of the node application at the local computer is not permitted.		
1020	A	NODE_NAME_NOT_FOUND Node name for node recovery does not exist in the cluster configuration file.	Node name in the cluster configuration file	Node name in the KDCFILE
1021	A	NODE_NAME_INCONSISTENCY The sequence of node names in the cluster configuration file and the KDCFILE do not match.		

Message K190 for the cluster user file

If &DIA1 is in the range 1100 to 1199, then &SUFF has the value 'USER' and the message refers to the cluster user file. In this case, the message can also be output by the KDCDEF utility program.

The table below provides an overview of the values of the insert &DIA1 for the cluster user file and their meanings:

&DIA1	Grp.	Meaning
1101	AS	TIMEOUT_FOR_USERFILE A timeout occurred on the request for a file lock when opening or closing the cluster user file.
1102	A	NO_USER_FILE The first page in the file is not a CONS page of a cluster user file.
1103	A	INVALID_FILE_VERSION The version number in the cluster user file does not match the version number in the UTM system code.

&DIA1	Grp.	Meaning
1104	A	APPLINAME_MISMATCH Different application name in the cluster user file and the KDCFILE.
1105	A	GEN_TIME_MISMATCH The generation time of the cluster user file specified in the cluster user file is not the same as that specified in the KDCFILE.
1106	A	CLUSTER_FILEBASE_MISMATCH The cluster filebase specified in the cluster user file is not the same as that specified in the KDCFILE.
1107	ASU	LAST_PAGE_CORRUPTED The last page in the file is not a CONS page of a cluster user file.
1108	U	PAGE_COUNTERS_INCONSISTENT The counters for the free and used entries in the cluster user file are inconsistent.
1109	U	FREE_ANCHOR_INCONSISTENT The free chain administration fields in the cluster user file are inconsistent.
1110	U	ENTRY_NOT_FOUND An error occurred while addressing an entry in the cluster user file.
1111	U	FREE_ANCHOR_BROKEN The chaining of the first entry in the free chain is inconsistent.
1112	U	USER_STATE_INVALID The status of a user entry in the free chain is invalid.
1113	U	ENTRY_NOT_FREE An entry in the free chain is not identified as free.
1114	U	END_OF_CHAIN_NOT_FOUND The last element was not found when the free chain was searched.
1115	U	FREE_CHAIN_BROKEN The number of elements in the free chain does not correspond to the counter in the control page.
1116	AU	MAX_FILE_SIZE_REACHED The cluster user file cannot be extended any further since it already contains the maximum number of entries.
1118	A	VERSION_MISMATCH The cluster user file and KDCFILE come from different UTM versions.

Message K190 for the cluster JKAA file

If &DIA1 is in the range 1200 to 1399, then &SUFF has the value 'JKAA' and the message refers to the cluster JKAA file.

The table below provides an overview of the values of the insert &DIA1 for the cluster JKAA file and their meanings:

&DIA1	Grp.	Meaning
1202	AU	FILE_IS_EMPTY The JKAA file is empty.
1203	AU	FILE_DOES_NOT_EXIST The JKAA file does not exist.
1204	AU	DMS_ERROR DMS error for the cluster JKAA file Insert &INF1 contains the DMS error code.
1205	M	MEMORY_INSUFFICIENT Error allocating memory for FCB and read buffer. Action: Increase the virtual memory.
1206	AU	FILE_OPEN_ERROR DMS error when opening the JKAA file. Insert &INF1 contains the DMS error code.
1207	ASU	LOCK_INIT_TIMEOUT Timeout when initializing file lock (KCCGFLI).
1208	SU	LOCK_INIT_ERROR Error when initializing file lock (KCCGFLI).
1209	ASU	FILE_LOCK_TIMEOUT Timeout when requesting file lock (KCCGFLK).
1210	SU	FILE_LOCK_ERROR Error when requesting file lock (KCCGFLK).
1211	AU	INVALID_FILE_TYPE The file is not a JKAA file.
1212	AU	FILE_CORRUPTED The second CONS page has an invalid format.
1213	ASU	FILE_UNLOCK_TIMEOUT Timeout when releasing file lock (KCCGFLK).
1214	SU	FILE_UNLOCK_ERROR Error when releasing file lock (KCCGFLK).
1215	ASU	DESTROY_LOCK_TIMEOUT Timeout when destroying file lock (KCCGFLK).
1216	SU	DESTROY_LOCK_ERROR Error when destroying file lock (KCCGFLK).

&DIA1	Grp.	Meaning
1217	SU	FILE_CLOSE_ERROR DMS error when closing the JKAA file.
1218	AU	READ_FILE_ERROR DMS error when reading the KAA pages of the JKAA file. Insert &INF1 contains the DMS error code.
1219	A	VERSION_MISMATCH The KDCFILE and JKAA file have different UTM versions. Insert &INF1 contains the UTM version of the KDCFILE. Insert &INF2 contains the UTM version of the JKAA file.
1221	A	APPLINAME_MISMATCH The KDCFILE and JKAA file have different application names. Insert &INF1 contains the application name of the KDCFILE. Insert &INF2 contains the application name of the JKAA file.
1223	A	DATETIME_NOT_EQUAL The KAA generation time of the JKAA file is not the same as the KAA generation time of the KDCFILE even though the node application that is starting has the same generation time as the running UTM cluster application. Insert &INF1 contains the KAA generation time of the KDCFILE. Insert &INF2 contains the KAA generation time of the JKAA file.
1224	A	DATETIME_NOT_LATER The KAA generation time of the KDCFILE is not later than the KAA generation time of the JKAA file even though the starting node application has been newly generated. Insert &INF1 contains the KAA generation time of the KDCFILE. Insert &INF2 contains the KAA generation time of the JKAA file.
1225	A	KCSANT_ERROR The KCSANT component returned an incorrect return code on the inclusion of a KSET for generation in the object tree of the local application.
1226	A	OBJECT_TYPE_ERROR On the generation of a user (USER) or an LU6.1 session in the local application, it was found that an LU6.1 session or user already exists under the index in the local application.
1301	A	BLKSIZE_ERROR The block size of the KDCFILE is not equal to 4K. Action: Generate KDCFILE with MAX BLKSIZE=4K.
1302	A	BUFFER_RQ_ERROR Error when requesting write/read buffer. Action: Increase the virtual memory.

&DIA1	Grp.	Meaning
1303	A	FCB_RQ_ERROR Error requesting memory for FCB. Action: Increase the virtual memory.
1304	AU	FILE_CMD_ERROR DMS error when setting up the JKAA file. Insert &INF1 contains the DMS error code.
1305	AU	OPEN_ERROR DMS error when opening the JKAA file. Insert &INF1 contains the DMS error code.
1306	U	WRITE_NSR_1_ERROR DMS error when writing the first NSR page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1307	U	WRITE_NSR_ERROR DMS error when writing the follow-up NSR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1308	U	WRITE_NSR_END_ERROR DMS error when writing the last block of NSR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1309	AU	PWRT_LOCK_ERROR Error when requesting periodic write lock.
1310	U	READ_DSR_1_ERROR DMS error when reading the first DSR page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1311	U	WRITE_SR_ERROR DMS error when writing SR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1312	U	WRITE_SR_END_ERROR DMS error when writing the last block of SR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1313	U	READ_SR_1_ERROR DMS error when reading the first SR page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.

&DIA1	Grp.	Meaning
1314	U	READ_SR_ERROR DMS error when reading the follow-up SR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1315	U	SR_PAGE_TYPE_ERROR Read SR page has incorrect page type. Insert &INF1 contains the type of the read page. Insert &INF1 contains the half page number.
1316	U	READ_DSR_ERROR DMS error when reading the follow-up DSR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1317	U	DSR_PAGE_TYPE_ERROR Read DSR page has incorrect page type. Insert &INF1 contains the type of the read page. Insert &INF1 contains the half page number.
1318	U	WRITE_DSR_ERROR DMS error when writing DSR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1319	U	WRITE_DSR_END_ERROR DMS error when writing the last block of DSR pages. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1320	U	READ_CONS_ERROR DMS error when reading the first CONS page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1321	AU	WRITE_CONS_1_ERROR DMS error when writing the first CONS page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1322	AU	WRITE_CONS_2_ERROR DMS error when writing the last CONS page. Insert &INF1 contains the DMS error code. Insert &INF1 contains the half page number.
1323	U	CLOSE_ERROR DMS error when closing the JKAA file. Insert &INF1 contains the DMS error code.
1324	SU	GFLI_ERROR Error when initializing file lock KCCGFL. Insert &INF1 contains the KCCGFLI return code.

&DIA1	Grp.	Meaning
1325	ASU	GFLI_TIMEOUT Timeout when initializing file lock KCCGFLI. Insert &INF1 contains the KCCGFLI return code.
1326	SU	GFLK_LOCK_ERROR Error when requesting file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1327	AU	GFLK_LOCK_TIMEOUT Timeout when requesting file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1328	SU	GFLK_UNLOCK_ERROR Error when releasing file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1329	ASU	GFLK_UNLOCK_TIMEOUT Timeout when releasing file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1330	SU	GFLK_DESTROY_ERROR Error when destroying file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1331	ASU	GFLK_DESTROY_TIMEOUT Timeout when destroying file lock KCCGFLK. Insert &INF1 contains the KCCGFLK return code.
1332	U	JFCT_START_ERROR Error when writing the copy status (Started) to the journal file.
1333	U	JFCT_END_ERROR Error when writing the copy status (Completed) to the journal file.

Message K190 for the administration journal files

If &DIA1 is in the range 1400 to 1499, then &SUFF has the value 'JRN' and the message refers to the administration journal.

The table below provides an overview of the values of the insert &DIA1 for the administration journal files and their meanings:

&DIA1	Grp.	Meaning
1401	AU	One or both journal files do not exist. See previous K043 message(s) Action: Regenerate the application.
1402	AU	Error when opening a journal file. See previous K043 message.
1403	AU	Error when creating a journal file. See previous K043 message.
1404	SU	Error when setting up lock for the journal files.
1405	SU	Error when closing a journal file. See previous K043 message.
1406	ASU	Error when releasing reserved pages in a journal file. See previous K043 message.
1407	A	Errored journal file(s): Incorrect ID. &INF1/&INF2: ID in file with suffix JRN1/JRN2. Action: Delete files, regenerate the application.
1408	A	Errored journal file(s): Incorrect file ID. &INF1/&INF2: ID in file with suffix JRN1/JRN2.
1409	A	Errored journal file(s): Incorrect UTM application name. &INF1/&INF2: Application name in JRN1 file / KDCFILE.
1410	A	Errored journal file(s): Incorrect UTM application name. &INF1/&INF2: Application name in JRN2 file / KDCFILE.
1411	A	Errored journal file(s): Incorrect cluster filebase. &INF1/&INF2: Cluster filebase in JRN1 file / KDCFILE.
1412	A	Errored journal file(s): Incorrect cluster filebase. &INF1/&INF2: Cluster filebase in JRN2 file / KDCFILE.
1413	A	Errored journal file(s): Different creation time. &INF1/&INF2: Creation time of file with suffix JRN1/JRN2.
1414	AU	Errored journal file(s): Invalid file status. &INF1/&INF2: File status in file with suffix JRN1/JRN2.
1415	SU	Error when destroying lock for the journal files.
1416	AU	Errored journal file(s): Incorrect file sequence numbers. &INF1/&INF2: File sequence numbers in JRN1/JRN2 file

&DIA1	Grp.	Meaning
1417	A	Errored journal file(s): Incorrect file version. &INF1/&INF2: Version of JRN1 file / expected version.
1418	A	Errored journal file(s): Incorrect file version. &INF1/&INF2: Version of JRN2 file / expected version.

Message K190 for the cluster page pool control file

If &DIA1 is in the range 1500 to 1599, then &SUFF has the value 'CPMD' and the message refers to the cluster page pool control file.

The table below provides an overview of the values of the insert &DIA1 for the cluster page pool control file and their meanings:

&DIA1	Grp.	Meaning
1501	A	Control file has incorrect file version. &INF1/&INF2: Version of CPMD file / expected version. Action: Regenerate all the UTM cluster files.
1502	A	A cluster update with transfer to the CPMD file was terminated abnormally. Action: Regenerate the cluster configuration files.
1503	A	The size of the cluster page pool has been reduced. &INF1/&INF2: Size in CPMD file / size in KDCFILE. Action: Regenerate all the UTM cluster files.
1504	A	The number of files in the cluster page pool has been changed. &INF1/&INF2: Number in CPMD file / number in KDCFILE. Action: Regenerate all the UTM cluster files.
1505	A	Error when trying to increase the size of the files in the cluster page pool. &INF1: 4K page number that could not be written. &INF2: Generated size of the cluster page pool in 4K pages. Action: Make sufficient disk space available for the UTM cluster files.
1506	AU	The size of the free page quota in the cluster page pool is invalid. &INF1: Size of quota as number of UTM pages. Action: Regenerate all the UTM cluster files or write problem report.
1507	AU	The number of pages in the free page quota in the cluster page pool is invalid. &INF1: Number of pages for the full free page quota. Action: Regenerate all the UTM cluster files or write problem report.
1508	AU	Error in management of the free page quota in the cluster page pool. Action: Regenerate all the UTM cluster files or write problem report.
1509	AU	Invalid ID for file size increase in the cluster page pool. &INF1: ID for file size increase. Action: Regenerate all the UTM cluster files or write problem report.

&DIA1	Grp.	Meaning
1510	AU	Invalid ID for running a cluster update with transfer to the CPMD file. &INF1: ID for running cluster update. Action: Regenerate all the UTM cluster files or write problem report.
1511	A	MAX APPLIMODE of the node application has been changed. To change between Secure and Fast, it is necessary to recreate all the UTM cluster files. &INF1/&INF2: APPLIMODE in CPMD file / in KDCFILE
1512	A	MAX VGMSIZE of the node application has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: VGMSIZE in CPMD file / in KDCFILE
1513	A	MAX KB of the node application has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: KB in CPMD file / in KDCFILE
1514	A	MAX NB of the node application has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: NB in CPMD file / in KDCFILE
1515	A	MAX LSSBS of the node application has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: LSSBS in CPMD file / in KDCFILE
1516	A	MAX TRMSGLTH of the node application has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: TRMSGLTH CPMD file / KDCFILE.
1517	A	The number of generated databases has been changed. All UTM cluster files must be regenerated. &INF1/&INF2: Number in CPMD file / KDCFILE.
1518	A	Invalid combination with and without user IDs. All UTM cluster files must be regenerated.
1519	A	Invalid combination with and without user formatting system. All UTM cluster files must be regenerated.
1520	A	Change of CLUSTER ABORT-BOUND-SERVICE. All UTM cluster files must be regenerated.
1521	A	Change of generated database systems. All UTM cluster files must be regenerated.

Message K190 for the cluster ULS file

If &DIA1 is in the range 1600 to 1699, then &SUFF has the value 'ULS' and the message refers to the cluster ULS file in which the administration data for ULS areas is stored in UTM cluster applications.

The table below provides an overview of the values of the insert &DIA1 for the cluster ULS file and their meanings:

&DIA1	Grp.	Meaning
1601	AU	The file is not a cluster ULS file or the file is corrupt.
1602	A	The version of the cluster ULS file is invalid.
1603	A	The application names are different in the cluster ULS file and KAA.
1604	AU	The last page of the cluster ULS file is destroyed.
1605	U	Generation error: The number of ULS blocks from the cluster ULS file is different from the number of ULS blocks in the KDCFILE.
1606	U	Generation error: A ULS from the cluster ULS file is missing in the KDCFILE.
1607	ASU	A timeout occurred while requesting a lock for the cluster ULS file.

Message K190 for the cluster lock file

If &DIA1 is in the range 1700 to 1799, then &SUFF has the value 'LOCK' and the message refers to the cluster lock file for locking global resources in UTM cluster applications (GSSB and ULS).

The table below provides an overview of the values of the insert &DIA1 for the cluster lock file and their meanings:

&DIA1	Grp.	Meaning
1700	AU	An error occurred while opening the existing cluster lock file. See previous K043 message.
1701	AU	An error occurred while creating the cluster lock file. See previous K043 message.
1702	SU	Error while writing the cluster lock file. See previous K043 message.
1703	SU	Error while closing the cluster lock file. See previous K043 message.
1704	SU	Error while opening the cluster lock file. See previous K043 message.
1705	SU	Error while replacing the cluster lock file. See previous K043 message.
1706	I	Information message - the cluster lock file has been extended. The number of new pages is output.
1707	SU	Error while closing the cluster lock file. See previous K043 message.

Message K190 for the cluster GSSB file

If &DIA1 is in the range 1800 to 1899, then &SUFF has the value 'GSSB' and the message refers to the cluster GSSB file in which the administration data for GSSB areas is stored in UTM cluster applications.

The table below provides an overview of the values of the insert &DIA1 for the cluster GSSB file and their meanings, together with the values and meanings of the inserts &INF1 and &INF2 which may occur in combination with &DIA1:

&DIA1	Grp.	Meaning	&INF1	&INF2
1800	ASU	GSSBFILE_LOCK_ERROR An error or timeout occurred when requesting or releasing a lock for the cluster GSSB file.	LOCK: Error when requesting lock LOCK: Error when releasing lock	TIMEOUT: Timeout (KCSGFLK return code) TRY LATER: Try later (KCSGFLK return code) LOCKED: File is already locked by own task. NOT LOCKED: File is not locked.
1801	A	INVALID_FILE_VERSION The version of the cluster GSSB file is invalid.	File version in the GSSB file	Expected file version
1802	AU	HEADER_PAGE_CORRUPTED The control page of the cluster GSSB file is destroyed.	Incorrect field content in the GSSB file	Expected field content
1803	AU	FREE_PAGES_ANCHOR_CORRUPTED The free page chaining of the cluster GSSB file is corrupt.		
1804	AU	NO_GSSBFILE The file is not a cluster GSSB file or the file is corrupt.	File type on the CONS page of the GSSB file	Expected file type
1805	AU	PAGE_COUNTER_INCONSISTENT The page counters of the cluster GSSB file are corrupt.		
1806	AU	LAST_PAGE_INCONSISTENCY The file sizes on the CONS and control pages of the cluster GSSB file are different.		
1808	SU	CONS_PAGE_READ_ERROR Error when reading the first CONS page of the cluster GSSB file.		
1809	SU	CONS_PAGE_WRITE_ERROR Error when writing the first CONS page of the cluster GSSB file.		

&DIA1	Grp.	Meaning	&INF1	&INF2
1810	SU	READ_ERROR_GSSBFILE Error when reading a page of the cluster GSSB file.		
1811	SU	WRITE_ERROR_GSSBFILE Error when writing a page of the cluster GSSB file.		
1812	U	RELEASE_BUFFER_ERROR Error when releasing a page of the cluster GSSB file.		
1814	U	FREE_ENTRY_CHAIN_CORRUPTED The chaining of the free GSSB entries in the cluster GSSB file is corrupt.		
1815	U	HASH_CHAIN_CORRUPTED The hash chaining of the GSSB entries in the cluster GSSB file is corrupt.		
1816	U	FREE_PAGE_CHAIN_CORRUPTED The free page chaining of the cluster GSSB file is corrupt.		
1817	U	GSSB_CNTR_ERROR The counter for the occupied GSSB entries in the cluster GSSB file is corrupt.		
1818	AU	MAX_GSSB_ERROR The value of MAX GSSB in the KDCFILE is different from the value in the cluster GSSB file.	MAX_GSSB in the GSSB file	MAX_GSSB in the KDCFILE

Message K190 for the UTM cluster files

If &DIA1 is in the range 1900 to 1949 then the message refers to the request for a lock for a UTM cluster file and is used for diagnostic purposes only.

The table below provides an overview of the values of the insert &DIA1 and their meanings:

&DIA1	Grp.	Meaning
1900	I	The lock request could not be executed in the generated time (operand FILE-LOCK-TIMER-SEC in the CLUSTER statement). The lock request is repeated in accordance with the value generated by the FILE-LOCK-RETRY operand in the CLUSTER statement. &INF1: Internal error code of the lock request. &INF2: Either the status of the XCS cluster or FILE LOCKED if the lock is maintained, see table below.
1901	I	This message is issued on the release of the file lock if the lock has been maintained by this process for at least half the time generated in the operand FILE-LOCK-TIMER-SEC in the CLUSTER statement. &INF1: Status of the requested lock. &INF2: Lock maintenance period in seconds.

&SUFF specifies the suffix of the UTM cluster file for which the lock was requested. Exception: The value SLCK does not stand for a file but for a logical lock in order to serialize the starts of the node applications.

The table below provides an overview of the values of the insert &INF2 and their meanings for &DIA1=1900:

&INF2	Meaning
FILE LOCKED	The UTM cluster file is currently locked by another process.
1	XCS_INACTIVE
2	XCS_ACTIVE
3	XCS_RECONF_JOIN
4	XCS_RECONF_LEAVE
5	XCS_RECONF_FAIL
6	XCS_INTERRUPTED
7	XCS_SUSPENDED

Message K190 on the failure of a node application

If &DIA1 is in the range 2001 to 2099, then &SUFF has the value 'KDCA' and the message refers to the KDCFILE of a node application. This message is output if a node failure is detected.

The table below provides an overview of the values of the insert &DIA1 and their meanings:

&DIA1	Grp.	Meaning
2001	I	The application is not active.
2002	S	An error occurred while requesting memory.
2003	ASU	An error was reported by the file system; see also the message K043
2004	SU	An error occurred while initializing the lock.
2005	U	The node could not be found in the UTM cluster file

&INF1 contains the node number of the node whose failure was detected.

&INF2 contains the file name of the KDCFILE of the node whose failure was detected.

K191 Dump will be written without &SUFF-file because of request memory error (&RQM bytes requested).

K192 Node recovery for node &NNM1 on host &HST1, RESET-PTC=&RSPTC

This message is output at SYSOUT and SYSLST at the start of a node recovery.

K193 PTC found : ID=&PTCID, USER=&USER, LPAP=&LPAP, LSES=&LSES, USER-type=&USTYPPTC

On a node recovery, this message is output once for every transaction with PTC status.

K194 After node recovery: Number of still locked GSSB: &GBLNBR, number of still locked ULS: &ULLNBR

This message is output at the end of node recovery.

If the failed node application for which a node recovery has been performed still has GSSB locks, then this can impair the running UTM cluster application.

K201 XA(&TSNPID) Resource Manager support using X/Open &XASPEC

&XASPEC identifies the version of the XA connection. The insert can have the values XA-CAE-Spec or XA-P-Spec.

K202 XA(&TSNPID) RM instance &INSTNUM, &TEXT32, &RMSTAT

The message provides information about the status of the XA connection.

The inserts have the following meanings:

&INSTNUM   Serial number of the Resource Manager instance, starting with 1. The numbering corresponds to the sequence of the start parameters.

&TEXT32   Name of the Resource Manager.

&RMSTAT   Status of the connection. The following values are possible:

&RMSTAT	**Meaning**
opened	The Resource Manager has been opened
reopened	The Resource Manager has been opened after a restart
closed	The Resource Manager has been closed

K203 XA(&TSNPID) RM &TEXT32,&INSTNUM - recovered transactions: &RTAANZ

The inserts have the following meanings:

&TEXT32   Name of the Resource Manager.

&INSTNUM   Serial number of the Resource Manager instance, starting with 1. The numbering corresponds to the sequence of the RMXA start parameters.

&RTAANZ   Number of recovered transactions.

K204 XA(&TSNPID) precommit requires global rollback - reason: &XATXT
TA=&INTTAID

Precommit failed, global transaction will be rolled back.

The inserts have the following meanings:

&XATXT   Reason for precommit failure. The following values are possible:

&XATXT	**Meaning**
XA_RBROLLBACK	Rollback for unspecified reason
XA_RBCOMMFAIL	Rollback due to a internal communication error in the Resource Manager
XA_RBDEADLOCK	Rolback due to a deadlock
XA_RBINTEGRITY	Rollback due to a resource inconsistency
XA_RBOTHER	Rollback for unspecified reason

&XATXT	**Meaning**
XA_RBPROTO	Rollback due to an internal protocol error in the Resource Manager
XA_RBTIMEOUT	Rollback due to transaction period timeout
XA_RBTRANSIENT	Rollback due to a temporary error

&INTTAID   Description of the UTM transaction which triggered the global rollback of the other transactions.

Byte 0 contains the instance number of the Resource Manager.

If the XID (= XA transaction identifier) was created in an OSI TP job-receiver transaction then the AAID (Atomic Action IDentifier) is added here.
Otherwise the subsequent bytes have the following content:

Byte	**Meaning**
1 - 8	Host name
9 - 16	Application name
17	Type of BS2000 hardware platform (7 ≙ /390, 8 ≙ RISC, 9 ≙ SPARC)
18	Bit mode (1 ≙ 32-bit platform)
19	Byte format (2 ≙ big-endian)
20	Number of the openUTM session
21 - 22	Number of the transaction in the service
23 - 24	Reserved
25 - 28	Service number
29 - 32	Service index

K205 XA(&TSNPID) transaction committed - reason: &XATXT
      TA=&INTTAID

The inserts have the following meanings:

&XATXT   Reason for committing the transaction. Possible reasons: "Recovery" or "Int.Event".

&INTTAID   Description of the UTM transaction that was committed (for contents, see K204).

K206 XA(&TSNPID) transaction rolled back - reason: &XATXT
TA=&INTTAID

The inserts have the following meanings:

&XATXT   Reason for the rollback. Possible reasons: "Recovery" or
"Int.Event".

&INTTAID   Description of the UTM transaction that was committed (for
contents, see K204)

K207 XA(&TSNPID) transaction is unknown to the RM&INSTNUM
TA=&INTTAID

The inserts have the following meanings:

&INSTNUM   Instance number of the Resource Manager, see K203.

&INTTAID   Description of the UTM transaction that is unknown to the RM (for
contents, see K204)

K210 XA(&TSNPID) Error: &XATXT - open RM: &TEXT32,&INSTNUM

K211 XA(&TSNPID) Error: &XATXT - close RM: &TEXT32,&INSTNUM

The inserts in the messages K210, K211 and K216 have the following meanings:

&TEXT32   Name of the Resource Manager.

&INSTNUM   Instance number of the Resource Manager, see K203.

&XATXT   Keyword for the return code from the called XA function. Possible
values:

&XATXT	**Meaning**
XA_OK	Normal execution
XA_NOMIGRATE	The transaction was suspended but the context cannot be migrated and the transaction can only be continued in the same process.
XA_HEURHAZ	The transaction may have been heuristically terminated due to a possible internal RM error.
XA_HEURRB	The transaction has been internally rolled back due to a heuristic Resource Manager decision.
XA_HEURMIX	The transaction has been internally partially committed and partially rolled back due to a heuristic Resource Manager decision.
XA_RETRY	The call to the routine must be repeated
XA_RDONLY	The transaction was "read_only" and was committed
XA_RBROLLBACK	Rollback for unspecified reason
XA_RBCOMMFAIL	Rollback due to a communication error

&XATXT	**Meaning**
XA_RBDEADLOCK	Rollback due to a deadlock
XA_RBINTEGRITY	Rollback due to a resource inconsistency
XA_RBOTHER	Rollback for unspecified reason
XA_RBPROTO	Rollback due to an internal RM protocol error
XA_RBTIMEOUT	Rollback due to a transaction period timeout
XA_RBTRANSIENT	Rollback due to a temporary error
XAER_ASYNC	An asynchronous operation is still outstanding
XAER_RMERR	A non-recoverable error has occurred in the Resource Manager. Possible cause: The Resource Manager was not initialized or was incorrectly initialized
XAER_NOTA	The transaction identifier is invalid or is unknown to the Resource Manager
XAER_INVAL	Invalid function arguments were specified. It is possible that the Open or Close strings contain incorrect parameters.
XAER_PROTO	The routine was called internally in the RM in the wrong context.
XAER_RMFAIL	The Resource Manager is no longer available
XAER_DUPID	The transaction identifier already exists in the Resource Manager. openUTM terminates the service. To eliminate this residual XID left by XAER_DUPID from the Resource Manager, the database administrator should remove this transaction. Possible action: shut down and restart the database.
XAER_OUTSIDE	The Resource Manager is operating outside of the transaction.

K212   XA(&TSNPID) xa_start(&XAFLAG) - return code: &XATXT
 TA=&INTTAID

K213   XA(&TSNPID) xa_end(&XAFLAG) - return code: &XATXT
 TA=&INTTAID

K214   XA(&TSNPID) xa_commit() - return code: &XATXT
 TA=&INTTAID

K215 XA(&TSNPID) xa_rollback() - return code: &XATXT
 TA=&INTTAID

The inserts in the messages K212 to K215 have the following meanings:

&XATXT   Keyword for the return code from the called XA function. Possible
            values, see K211.

&XAFLAG   Keyword that describes the flags on an xa_start() or xa_end() call.

&INTTAID   Description of the UTM transaction during which the return code
            occurred (for contents, see K204)

K216 XA(&TSNPID) return code: &XATXT - recover PTC list, RM: &TEXT32,&INSTNUM

For the meaning and content of the inserts, see K211.

K217 XA(&TSNPID) xa_prepare() - return code: &XATXT
 TA=&INTTAID

For the meaning of the inserts, see K211 (&XATXT) and K215 (&INTTAID).

K218 XA(&TSNPID) xa_forget() - return code: &XATXT
 TA=&INTTAID

For the meaning of the inserts, see K211 (&XATXT) and K215 (&INTTAID).

K220 XA(&TSNPID) Error: xa_switch definition not found for specified RM: &TEXT32

A Resource Manager (RM) for which the RMXA start parameter was specified was
not defined in the KDCDEF generation.
&TEXT32 contains the name Resource Manager.

K221 XA(&TSNPID) Error: Start parameters not found for defined RM: &TEXT32

No start parameters are specified for a Resource Manager (RM) present in the
KDCDEF generation.
&TEXT32 contains the name Resource Manager.

K222 XA(&TSNPID) Error: Linked RM is not &XASPEC compatible: &TEXT32

The Resource Manager (RM) does not use the XA interface that was specified The
Resource Manager (RM) does not use the XA interface that was specified during
KDCDEF generation.
&TEXT32 contains the name of the Resource Managers and &XASPEC the XA
interface specified during generation.

K223 XA(&TSNPID) Syntax error in start parameters:

The following line contains the incorrect line from the start parameter file.

K224   XA(&TSNPID) &XACALL - return code &XATXT from RM instance &INSTNUM, &TEXT32 is not XA(CAE) compliant

The Resource Manager (RM) returns an unexpected return code.

The meaning of the inserts is as follows:

&XACALL   Name of the call to the XA interface, e.g. xa_start().

&XATXT   Keyword for the return code from the called XA function (see K211).

&INSTNUM   Instance number of the Resource Manager, see K203.

&TEXT32   Name of the Resource Manager.

K225   XA(&TSNPID) recursive call of XA module: function: &XADBC1 - error/signal in DB/XA connection for &XADBC2

The meaning of the inserts is as follows:

&XADBC1   Name of the called function in the UTM/XA connection module.

&XADBC2   Name of the function in which an error was detected in the UTM/XA connection module.

K230   XA(&TSNPID) Int. error: &TEXT32

K231   XA(&TSNPID) Int. error: PETA not supported

&TEXT32   Specifies an internal error. This may, for example, indicate a lack of memory space for malloc().

K231   XA(&TSNPID) Int. error: PETA not supported

Preliminary end of transaction not supported.

K232   XA(&TSNPID) Int. error: DBSTAT secondary opcode inconsistent

The secondary operation code of the primary operation code DBSTAT is inconsistent.

K233   Heur. decision in task / process &TSNPID, RM=&INSTNUM in
&XACALL(&DBCALL), xa-rc=&XATXT, LTHGTRID=&LTHGTRID, GTRID=&GTRID

The meaning of the inserts is as follows:

&INSTNUM   Instance number of the Resource Manager (RM), see K203.

&XACALL   Name of the call to the XA interface, e.g. xa_start().

&DBCALL   Name of the called UTM-DB interface function

&XATXT   Keyword for the return code from the called XA function (see K211).

&GTRID   Global transaction ID (gtrid) of the UTM transaction in accordance
with the XA/CAE specification (for contents, see K204).

&LTHGTRID   Length of the global transaction ID.

K234   LMS error (PLA&PLARC, DMS&DMSRC) for
library: &FNAM
element: &ELEM
version: &EVER
type: &LMSTYPE

The meaning of the inserts is as follows:

&PLARC   PLAM return code. The help text can also be queried using /HELP
PLA&PLARC.

&DMSRC   DMS return code. The help text can also be queried using /HELP
DMS&DMSRC.

&FNAM   Name of the PLAM library to which the message refers.

&ELEM   Name of the LMS element to which the message refers.

&EVER   LMS version of the LMS element.

&LMSTYPE   LMS type of the LMS element.

K235   Name resolution for &PRNM lasts &TCPMS milliseconds (socket call: &TCPCL,
returncode: &TCPRC, IP address: &IPADDR).

K251   Version V&IMPVER of filebase &FBASUPD does not match current version
V&DEFVER.

K252   &UPDERR error

If &UPDERR is set to the value PARAM, then this points to a syntax error. KDCUPD
aborts and sets switch 3 to ON.
Action: Enter the correct command!
Otherwise an error will occur when requesting memory, when reading in data, etc.
Action: Write a problem report with documentation.

K255   DMS error &DMSE on file &FNAM

Action: Depends on DMS error message.

K256   File type of &FNAM is not a valid UTM type

Action: Make correct file available.

K257   Application run was not terminated correctly

K258   File &FNAM has already been used

K260   Unknown version &DEFVER of filebase &FBASUPD

Action: Make correct file available or problem report.

K261   Inconsistent file &FNAM

Action: Restore file or repeat generation.

K262   Wrong file &FNAM

Action: Make correct file available or repeat generation.

K263   File &FNAM has not yet been used

No start-up has yet been performed with the file `&FNAM`.

K269   &OBJ1 &OST1 &BMD1 and &OBJ3 &OST2 &BMD2 are not compatible

K273   Error &TRMA in module &UPDMODUL

The KDCUPD run was aborted. The error code &TRMA provides information on the cause.

&TRMA	Cause	Action
BFMMER	Error initializing buffer management (cluster)	if necessary, increase virtual address space or PM
CFGERR	Error processing the cluster config-uration file	previous diagnostic messages, or if there are none: inconsistent node name sequence
CONSER	Internal error	Write PR and compile documentation
DMSERR	A DMS error has occurred	See K255
DxxxRD	DMS error xxx when reading the first page	Depends on DMS code
DxxxWR	DMS error xxx when writing the first page	Depends on DMS code
xxxx01	DMS error xxx when reading the first KAA page	Depends on DMS code
GMDTER	Error checking the cluster GSSB file	see previous diagnostic messages

&TRMA	Cause	Action
INCONS	Error during consistency checks	See additional message K261/K262
LOOKFL LOOKFU	Error on searching for originator (LTERM, USER)	Write PR and compile documentation
LOOKLT	Error on searching for LTERM	Write PR and compile documentation
LOOKTC	Error on searching for TAC	Write PR and compile documentation
LOOKUS	Error on searching for USER	Write PR and compile documentation
READxx	DMS error	See start error code xx in K049
REQKTA	'Request memory' error for KTA	Insufficient storage space available, try again with reduced system load
REQM01	'Request memory' error for 1st page	Regenerate system if necessary
REQM02	Memory shortage when creating memory pool for UTM cache	Check BS2000 generation or reduce the CACHESIZE operand
OPWRON	Internal error	Write PR and compile documentation
PPMMER	Cluster page pool management file incorrect.	see previous diagnostic messages
UMDTER	Error checking the cluster ULS file	see previous diagnostic messages
UPDSTA	File has already been processed with KDCUPD	Assign correct file
USFERR	Error opening the cluster user file	see previous diagnostic messages
WRDB01	Abort in KDCUPD write module. Cause: The number of DB systems in the new KDCFILE is smaller than in the old KDCFILE. Can occur only if old and new KDCFILE are generated with UTM-S. K307 is also output.	Modify the generation of the new KDCFILE
WRDB02	Abort in KDCUPD write module. Cause: The type of database(s) in the new KDCFILE is different from the type in the old KDCFILE. Can occur only if old and new KDCFILE are generated with UTM-S. K307 is also output.	Modify the generation of the new KDCFILE

&TRMA	Cause	Action
WRDB03	Abort in the KDCUPD write module. Cause: at least 1 DB system is in the old KDCFILE and there are more DB systems in the new KDCFILE than in the old KDCFILE. Can occur only if old and new KDCFILE are generated with UTM-S. K307 is also output.	Modify the generation of the new KDCFILE
WRFORM	Abort in the KDCUPD write module. Cause: the old KDCFILE was generated with formatting and the new KDCFILE without formatting, both KDCFILEs are generated with the variant UTM-S.	Modify the generation of the new KDCFILE
WRITxx	DMS error	See start error code xx in K049
...GSB ....GB	Error during GSSB handling	Write PR and compile documentation
...TLS ....TL	Error during TLS handling	Write PR and compile documentation
...ULS ....UL	Error during ULS handling	Write PR and compile documentation
..DIAL	Error during transfer of chained services	Write PR and compile documentation
...ASY ....AS	Error during handling of FPUTs or DPUTs at LTERM	Write PR and compile documentation
...IMS ....IM	Error during handling of FPUTs or DPUTs at TAC	Write PR and compile documentation
...UPI ....UP	Error while processing a dialog message for UPIC	Write problem report and create documentation
...SOC ....SO	Error while processing a dialog message for SOCKET	Write problem report and create documentation
...UMS ....UM	Error while processing a user queue message	Write problem report and create documentation
...QMS ....QM	Error while processing a queue message	Write problem report and create documentation
...LSB	Error during LSSB handling	Write PR and compile documentation

In addition, the error codes indicating abnormal termination of UTM may be output (for description see section "UTM dump error codes" on page 185). In this case, you should always write an error report and compile relevant documentation.

K274 Terminated with a dump

K277 At least one node application was not terminated correctly.

For more information, see SYSOUT protocol.

For more information, see SYSOUT protocol.

K278 At least one node application has already been started.

K279 Error in cluster-handling, see (possible) diagnostic message before

K300 &UPDPRO percent of &PGPOOL used.

K303 &UKCOP data transferred. KCRN = &UKCRN, type = &UPDTYP, KCLM = &UKCLM.

K304 Service data transferred for user &USER. Service type = &TACTYPE, height of stack = &UKCHSTA

K305 Used pages of &PGPOOL for &UPDTYP: &PGS1 overall, &PGS2 for &UKCRN.

K306 Used pages of &PGPOOL for &UPDTYP: &PGS1.

K310 * &UPDTYP &UKCRN not found

K311 * &UPDTYP &UKCRN not found. No data transferred

K314 * Warning: LPUT data cannot be transferred

K317 * &UKCOP data not transferred. KCRN= &UKCRN, Type = &UPDTYP, KCLM = &UKCLM, KCRCCC = &RCCC, KCRCDC = &RCDC . Caused by LTERM = &LTRM , user = &USER

K318 * &UPDTYP &UKCRN : sender of asynchronous message not found. LTERM = &LTRM, user = &USER

K320 * Service data for user &USER not transferred. Service type = &TACTYPE, reason: &UERCODE, &UERINFO, &RCDC

The insert &UERRCODE indicates why the service data for USER &USER was not transferred. The insert &UERRINFO contains additional information.

&UERRCODE	&UERRINFO	Meaning
EDIT	name	The edit profile name is not generated.
KBLTH	bytes	The generation parameter MAX KB in the new KDCFILE is too small. The service requires the number of bytes specified for "bytes".
LOCK ERR	tac	The user does not have the necessary key for the transaction code "tac" in his/her key set
LSSBS	number	The number "number" of LSSBs in a service of the user is larger than the generation value (MAX LSSBS =).

&UERRCODE	&UERRINFO	Meaning
LTERM	lterm	The sender LTERM "lterm" is not defined.
MISSMATCH		The user is generated in the new KDCFILE with RESTART=NO or the user type has changed (connection user ID yes/no).
NRCONV	number	The number "number" of stacked services of the user is larger than the generation value (MAX NRCONV =).
PAGEPOOL		The page pool is too small (MAX PAGEPOOL =).
RESSHORT		The UTM table entries for administrating service data are not sufficient. More user IDs, sessions and/or tasks were generated in the new KDCFILE than in the old or the value of ASYNTASKS is less than in the old KDCFILE.
TAC	name	The transaction code "name" is invalid.
TRMSGLTH		The message for a continuation program is too large (MAX TRMSGLTH =).
VGMSIZE	size	The service memory (size in KB) is larger than the generation value in the new KDCFILE (MAX VGMSIZE=).

5.2 Messages of the XAP-TP provider

The messages from the XAP-TP provider all start with the letter "P". The values for the inserts are either described following the message or (if the insert occurs a number of times) in the section "General inserts for the XAP-TP messages" on page 365.

P001 Error on OSS call (&XPFUNC): &ACPNT, &XPRET, &XPERR, &XP1INFO, &XP2INFO

This message is output if a call to an OSS function (*&XPFUNC*) returns an error. If the error has been reported by the transport system, message P012 is also output.

The inserts have the following meaning:

&XPFUNC   Name of the OSS function

&ACPNT   Name of the local ACCESS-POINT

&XPRET   See table on page 365

&XPERR   See table on page 365

&XP1INFO   Supplementary OSS information

&XP2INFO   Supplementary OSS information

P002 Error on association establishment (&XPFUNC): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

This message is issued if the call to an OSS function (*&XPFUNC*) required to establish an association returns an error. If the error has been reported by the transport system, message P012 is also output. If the error has not been reported by the transport system, the application is terminated with "Termapplication".

The inserts have the following meaning:

&XPFUNC   Name of the OSS function

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPRET   See table on page 365

&XPERR   See table on page 365

&XP1INFO   Supplementary OSS information

&XP2INFO   Supplementary OSS information

P003   Association rejected (a_assin()):&ACPNT, reason: &XPRJCT, length: &XPLTH

This message is issued if a request to establish an association was rejected from outside.

The inserts have the following meaning:

     &ACPNT   Name of the local ACCESS-POINT

     &XPLTH   Incorrect length

     &XPRJCT   See table on

P004   Association rejected (a_assin()):&ACPNT, &OSLPAP, reason: &XPRJCT

This message is issued if a request to establish an association was rejected from outside.

The inserts have the following meaning:

     &ACPNT   Name of the local ACCESS-POINT

     &OSLPAP   Name of the partner in the local application

     &XPRJCT   See table on

Possible causes in the case of &XPRJCT = 34 or &XPRJCT = 35
(NO_MORE_CONTENTION_LOSER / WINNER_ASSOCIATIONS,):

–   Associations have been disconnected without the local UTM application being informed of this. The local UTM application considers that these associations still exist. The OSI TP partner application attempts to re-establish these associations. However, these cannot be re-established until either the affected associations have been disconnected due to the expiry of the idle timer or all associations to the relevant partner have been disconnected at the administrative level.

–   More associations are generated in the OSI TP partner application than in the local UTM application.

–   Different numbers of contention losers/winners in the local and OSI TP partner applications.

P005 Association rejected (a_assin()):&ACPNT, reason: unknown partner
N-SEL: &XPNSEL, T-SEL: &XPTSEL
S-SEL: (&XPLSSEL,&XPCSSEL,&XPHSSEL)
P-SEL: (&XPLPSEL,&XPCPSEL,&XPHPSEL)

This message is issued if a request to establish an association was rejected from outside because the remote partner is not known to the local application.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&XPNSEL   Network selector (BCAM processor name) of the remote partner

&XPTSEL   Transport selector (BCAM application name) of the remote partner

&XPLSSEL   Length of the session selector of the remote partner

&XPCSSEL   Session selector (printable) of the remote partner

&XPHSSEL   Session selector (hexadecimal) of the remote partner

&XPLPSEL   Length of the presentation selector of the remote partner

&XPCPSEL   Presentation selector (printable) of the remote partner

&XPXPSEL   Presentation selector (hexadecimal) of the remote partner

P006 Association rejected (a_assin()): &ACPNT, &OSLPAP, reason: wrong application context name (&XP0OBID, &XP1OBID, &XP2OBID, &XP3OBID, &XP4OBID, &XP5OBID, &XP6OBID, &XP7OBID, &XP8OBID, &XP9OBID)

This message is issued if a request to establish an association was rejected from outside. The application context name for the remote partner does not match the application context name generated for this partner in the local application.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XP0OBID - &XP9OBID

           These are (up to) ten elements of the object identifier which form the application context name of the remote partner.
           -1 is output for elements which do not have a value assigned.

P007   Error on association establishment (a_assrs()): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

This message is output when a call to the OSS function *a_assrs()* to respond to a request to establish an association from outside returns an error. If the error has been reported by the transport system, message P012 is also output.
The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPRET   See table on page 365

&XPERR   See table on page 365

&XP1INFO   Supplementary OSS information

&XP2INFO   Supplementary OSS information

P008   Association established: &ACPNT, &OSLPAP

This message is issued when an association has been established.
The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

P009   Association rejected (a_asscf()): &ACPNT, &OSLPAP, reason: &XPRJCT, length: &XPLTH

This message is issued when active establishment of an association is rejected because the confirmation from the partner cannot be accepted.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPRJCT   See table on page 368

&XPLTH   Possible incorrect length

P010 Association rejected (a_asscf()): &ACPNT, &OSLPAP, reason: unknown partner
N-SEL: &XPNSEL, T-SEL: &XPTSEL
S-SEL: (&XPLSSEL,&XPCSSEL,&XPHSSEL)
P-SEL: (&XPLPSEL,&XPCPSEL,&XPHPSEL)

This message is issued when active establishment of an association is rejected,
because the remote partner confirms establishment of an association with an
address (&XPADDR) which is unknown to the local application.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPNSEL   Network selector (BCAM processor name) of the remote partner

&XPTSEL   Transport selector (BCAM application name) of the remote partner

&XPLSSEL   Length of the session selector of the remote partner

&XPCSSEL   Session selector (printable) of the remote partner

&XPHSSEL   Session selector (hexadecimal) of the remote partner

&XPLPSEL   Length of the presentation selector of the remote partner

&XPCPSEL   Presentation selector (printable) of the remote partner

&XPXPSEL   Presentation selector (hexadecimal) of the remote partner

P011 Association rejected (a_asscf()): &ACPNT, &OSLPAP, reason: wrong application
context name (&XP0OBID, &XP1OBID, &XP2OBID, &XP3OBID, &XP4OBID,
&XP5OBID, &XP6OBID, &XP7OBID, &XP8OBID, &XP9OBID)

This message is issued when active establishment of an association is rejected,
because the remote partner confirms establishment of an association with an appli-
cation context name other than the one configured for this partner in the local appli-
cation.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XP0OBID - &XP9OBID

These are (up to) ten elements of the object identifier which form the
application context name of the remote partner.
-1 is output for elements which do not have a value assigned.

P012   CMX diagnostic information: &XPCTYPE, &XPCCLS, &XPCVAL, &XPBCAM

This message is issued if a preceding message is issued as a result of an error reported by the transport system. The diagnostic code of the transport system is print-edited. The following table describes a number of values for &XPCTYPE, &XPCCLS and &XPCVAL. The CMX header file `cmx.h` contains a complete list.

XPCTYPE	Meaning (CMX error type)
0	T_CMXTYPE: CMX error detected by the CMX library
2	T_DSTEMPERR: Temporary TNS error
3	T_DSCALL_ERR: TNS call error
4	T_DSPERM_ERR: Permanent TNS error
5	T_DSWARNING: TNS warning
>15	CMX error on the basis of error codes from the transport system

XPCCLS	Meaning (CMX error class, valid for &XPCTYPE < 15)
0	T_CMXCLASS: CMX class
2	T_DSNOT_SPEC: TNS class not specified
3	T_DSPAR_ERR: TNS parameter error
4	T_DSILL_VERS: Invalid TNS version
5	T_DSSYS_ERR: TNS system error
6	T_DSINT_ERR: Internal TNS error
7	T_DSMESSAGE: TNS note

XPCVAL	Meaning (CMX error value)
0	T_NOERROR: No error
5	T_EIO: Temporary bottleneck or error in the transport system
14	T_EFAULT: IO_Area not allocated
100	T_UNSPECIFIED: Unspecified error, generally a system call error
101	T_WSEQUENCE: Invalid call sequence
103	T_WPARAMETER: Invalid parameter
104	T_WAPPLICATION: The application is not known to TNS of the task is not authorized to sign on to the application or the application has already been opened by this task.
105	T_WAPP_LIMIT: The limit for the number of simultaneously active applications has already been reached.

XPCVAL	Meaning (CMX error value)
106	T_WCONN_LIMIT: The limit for the number of simultaneously active connections has already been reached.
107	T_WTREF: Invalid transport reference or the transport connection has already been established.
111	T_NOCCP: The transport system does not support the requested application or connection.
114	T_CCP_END: The transport system has been terminated or the application was closed by the administrator.
255	T_WLIBVERSION: No connection to the CMX subsystem possible.
-100	T_INVREF: Invalid evid. CMX cannot assign the call to a wait point.

&XPBCAM   if the value of &XPCTYP is greater than 15, the BCAM info word is output with &XPBCAM.

CMX return code

Refer to the appendix of the CMX manual for details on the structure of the CMX return code. Two different situations must be taken into account when evaluating the CMX return code:

1. The CMX error type has a value less than 15: a system error has occurred.

2. The CMX error type has a value larger than 15: the transport system has reported an error and the BCAM return code is contained in the CMX return code. Bits 0 through 7 of the CMX return code contain the CMX error value (refer to the CMX manual for the value). The BCAM return code can be derived from the rest of the CMX return code using the following method. SC1 of the BCAM return code (S-RTC1) must always be set to 00.
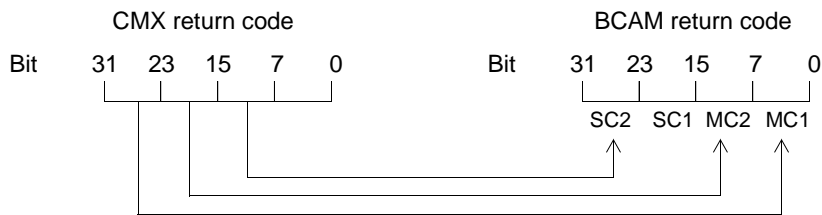
Where:

SC1/2 = S-RTC1/2 (Subcodes)
MC1/2 = M-RTC1/2 (Maincodes)

Bits 24 through 31 of the CMX return code are passed in the field MC1 of the BCAM return code.

Bits 16 through 23 of the CMX return code are passed in the field MC2 of the BCAM return code.

Bits 8 through 15 of the CMX return code are passed in the field SC2 of the BCAM return code.

Refer to the appendix of the CMX manual for the meaning of the diagnostic code.

P013   Association rejected (a_asscf()): &ACPNT, &OSLPAP, reason: &XPCRES,
           &XPSRC, &XPNDIA
           CCR V2 = &XP1BOOL, Version Incompatibility = &XP2BOOL
           ContWin Assignment rejected = &XP3BOOL
           Bid mandatory rejected = &XP4BOOL, No reason = &XP5BOOL

This message is issued when active establishment of an association is rejected by the remote partner. The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPCRES   Specifies whether the rejection is temporary or permanent:
                0= permanent reject
                1= transient reject

&XPCSRC   Specifies who has rejected establishment of the association:
                0 = ACSE service user
                1 = ACSE service provider
                2 = Presentation service provider

&XPNDIA   See table on

&XP1BOOL - &XP5BOOL
                These inserts can take the values TRUE or FALSE. Values of TRUE indicate the reasons the partner reported for rejecting the request to establish an association:

                &XP1BOOL: CCR Version 2 is not available
                &XP2BOOL: The TP protocol versions are not compatible
                &XP3BOOL: The contention winner assignment has been rejected
                &XP4BOOL: The specification "Bidding is mandatory" or "Bidding is not mandatory" has been rejected
                &XP5BOOL: No reason is specified

Possible causes for the rejection of the contention winner assignment (&XP3BOOL=TRUE):

– Fewer associations are generated in the OSI TP partner application than in the local UTM application.

– Different number of contention losers/winners in the local and the partner application.

– Associations have been disconnected without the OSI TP partner application being informed of this. The OSI TP partner application rejects the establishment of associations since it considers that they are still established.
  In this situation, the message P004 with &XPRJCT = 34 or &XPRJCT = 35 is generated in the OSI TP partner application
  (NO_MORE_CONTENTION_LOSER / WINNER_ASSOCIATIONS,).

P014 Error on association disconnection (&XPFUNC): &ACPNT, &OSLPAP, &XPRET, &XPERR, &XP1INFO, &XP2INFO

This message is issued if the call to an OSS function (*&XPFUNC*) required to establish an association returns an error. If the error has been reported by the transport system, message P012 is also output. If the error has not been reported by the transport system, the application is terminated with "Termapplication".

The inserts have the following meaning:

&XPFUNC	Name of the OSS function
&ACPNT	Name of the local ACCESS-POINT
&OSLPAP	Name of the partner in the local application
&XPRET	See table on page 365
&XPERR	See table on page 365
&XP1INFO	Supplementary OSS information
&XP2INFO	Supplementary OSS information, currently always set to zero.

P015 Association disconnected (&XPFUNC): &ACPNT, &OSLPAP, &XPLNK, &XPSRC, &XPNDIA, &XPINI, &XP1INFO, &XP2INFO

This message is issued when an association is cleared.

The inserts have the following meaning:

&XPFUNC	Name of the OSS function
&ACPNT	Name of the local ACCESS-POINT
&OSLPAP	Name of the partner in the local application
&XPLNK	Represents the internal status of the association 0 = Association not linked 1 = Association linked to channel 2 = Association linked to instance
&XPCSRC	Originator of clear-down 0 = ACSE service user 1 = ACSE service provider 2 = Presentation service provider
&XPNDIA	See table on page 372
&XP1INFO	Supplementary OSS information
&XP2INFO	Supplementary OSS information
&XPINI	See table below

XPINI	Meaning
0	Association was cleared internally.
401	O_LOC_TRAN The originator is the local transport system. &XP1INFO contains the CMX return code. This is output in detail in the subsequent message P012.
402	O_REM_TRAN The originator is the remote transport system. &XP1INFO contains the reason for the CMX event t_disin. The values are defined in cmx.h. Below are some of the possible values for &XP1INFO and their meanings. The following abbreviations are used: – CCP (**C**ommunication **C**ontrol **P**rogram) for the application program that controls communication – TSAP (**T**ransport **S**ervice **A**ccess **P**oint) for the access point to the transport service, – NSAP (**N**etwork **S**ervice **A**ccess **P**oint) for the access point to the network service. – PDU (**P**rotocol **D**ata **U**nit) for data elements in a protocol layer. &XP1INFO can, for example, take on the following values: 0 (T_USER) The communication partner cleared the association, possibly as a result of a user error on the partner side. 1 (T_RTIMEOUT) The connection was cleared locally by CMX because the connection had been inactive for too long according to the t_timeout parameter. 2 (T_RADMIN) The connection was cleared locally by CMX because the administrator closed down CCP. 3 (T_CCPEND) The connection was cleared locally by CMX because CCP failed. 256 (T_RUNKOWN) Either the partner or CCP cleared the connection. No reason was given. 257 (T_RSAP_CONGEST) The partner CCP cleared the connection because of a TSAP-specific bottleneck. 258 (T_RSAP_NOTATT) The partner CCP cleared the connection because the addressed TSAP was not registered there. 259 (T_RUNSAP) The partner CCP cleared the connection because the addressed TSAP was not known there.

XPINI	Meaning
402 (cont.)	261 (T_RPERMLOST) The connection was cleared by the network administrator or the partner CCP administrator.
	262 (T_RSYSERR) Error in the network
	385 (T_RCONGEST) The partner CCP cleared the connection as a result of a resource bottleneck.
	386 (T_RCONNFAIL) No connection could be established. The partner CCP aborted the attempt to do so.
	387 (T_RDUPREF) The partner CCP cleared the connection because a second connection reference was assigned for an NSAP pair (system error).
	388 (T_RMISREF) The partner CCP cleared the connection because a connection reference could not be assigned (system error).
	389 (T_PROTERR) The partner CCP cleared the connection because of a protocol error (system error).
	391 (T_PREFOFLOW) The partner CCP cleared the connection because of a connection reference overflow.
	392 (T_RNOCONN) The partner CCP rejected the request to establish a network connection.
	394 (T_RINLNG) The partner CCP cleared the connection because of an incorrect length header or parameter (system error).
	448 (T_RLCONGEST) The local CCP cleared the connection because of a resource bottleneck.
	449 (T_RLNOQOS) The local CCP cleared the connection because the "quality of service" could not be maintained.
	451 (T_RILLPWD) Invalid connection password.
	452 (RNETACC) Access to the network was refused.
	464 (T_RLPROTERR) The local CCP cleared the connection because of a transport protocol error (system error).

XPINI	Meaning
402 (cont.)	465 (T_RLINTIDU) The local CCP cleared the connection because it received an interface data unit which was too long (system error).
	466 (T_RLNORMFLOW) The local CCP cleared the connection because of an infringement of the flow control rules for normal data (system error).
	467 (T_RLEXFLOW) The local CCP cleared the connection because of an infringement of the flow control rules for expedited data (system error).
	468 (T_RLINSAPID) The local CCP cleared the connection because it received an invalid TSAP identification (system error).
	469 (T_RLINCEPID) The local CCP cleared the connection because it received an invalid TCEP identification (**T**ransport **C**onnection **E**nd **P**oint, end point of a transport connection) (system error).
	470 (T_RLINPAR) The local CCP cleared the connection because of an invalid parameter value (e.g. user data too long or expedited data not permitted).
	480 (T_RLNOPERM) The administrator of the local CCP prevented establishment of a connection.
	481 (T_RLPERMLOST) The administrator of the local CCP cleared the connection.
	482 (T_RLNOCONN) The local CCP could not establish the connection because no network connection is available.
	483 (T_RLCONNLOST) The local CCP cleared the connection because the network connection was lost. Most common cause: generation error on the CCP and PDN side, e.g. incorrect link addresses. Other possible causes: partner is not available, modem is faulty or has been set incorrectly, data transfer connection not plugged in, data transfer card faulty.
	484 (T_RLNORESP) The local CCP cannot establish the connection because the partner does not respond to the connection request (CONRQ).
	485 (T_RLIDLETRAF) The local CCP cleared the connection because the connection was lost (Idle Traffic Timeout).
	486 (T_RLRESYNC) The local CCP cleared the connection because it was not possible to resynchronize (more than ten attempts were made).

XPINI	Meaning
402 (cont.)	487 (T_RLEXLOST) The local CCP cleared the connection because the expedited data channel is faulty (more than three attempts were made).
403	O_LOC_SESS The originator is the local session provider. &XP1INFO can take the following values: 4 (S_PROTERROR) Protocol error: Incorrect establishment of the session PDU or incorrect SPDU parameter 16 (S_PICSREST) Violation of implementation-specific restrictions.
404	O_REM_SESS The originator is the remote session provider. &XP1INFO can take the following values: 1 (S_TCDISCON) Disconnection of the transport connection (transport disconnect) 4 (S_PROTERROR) protocol error 8 (S_UNDEFINED) undefined 16 (S_PICSREST) violation against implementation-related restrictions
405	O_LOC_PRES The originator is the local presentation provider. &XP1INFO can take the following values: 0 (P_ARRNO) reason not specified – A decoding buffer requested internally cannot be provided due to a lack of memory. – Overflow of the internal data buffer when reassembling fragmented messages – An unknown session event was reported. – System bottleneck or system error. 1 (P_ARNRPDU) unrecognized PPDU – No session user data is available or the presentation part of the session user data cannot be decoded (system error). 4 (P_ARNRPAR) unrecognized PPDU parameter – Error on decoding the ACSE, presentation or user syntax.

XPINI	Meaning
405 (cont.)	5 (P_ARNEPAR) unexpected PPDU parameter – PPDU parameter not in normal mode. 6 (P_ARNIPAR) invalid PPDU parameter – Invalid context identifier on decoding. – Invalid PPDU parameter, e.g. incorrect length. This "abort" can be triggered by the UTM user by specifying invalid presentation or session selectors.
406	O_REM_PRES The originator is the remote presentation provider. &XP1INFO can take the following values: -1 (O_NOVALUE) Optional parameter is not present 0 (P_ARNNO) Reason not specified 1 (P_ARNRPDU) Unrecognized PPDU 2 (P_ARNEPDU) Unexpected PPDU 3 (P_ARNESSP) Unexpected session service primitive 4 (P_ARNRPAR) Unrecognized PPDU parameter 5 (P_ARNEPAR) Unexpected PPDU parameter 6 (P_ARNIPAR) Invalid PPDU parameter value
407	O_LOC_ACSE The originator is the local ACSE provider &XP1INFO always has the following value: 1 (A_ABSASP) ACSE service provider initiated the abort The instance is specified which initiated the abort ("abort source") from the point of view of ACSE.

XPINI	Meaning
408	O_REM_ACSE The originator is the remote ACSE service provider. &XP1INFO can take the following values: 0 (A_ABSASU) ACSE service user initiated the abort 1 (A_ABSASP) ACSE service provider initiated the abort

P016   Association disconnected (a_relin()): &ACPNT, &OSLPAP, &XPLNK, &XPNDIA

This message is issued if an association is cleared because a "release indication" was received.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPLNK   Represents the internal status of the association
              0 = Association not linked
              1 = Association linked to channel
              2 = Association linked to instance

&XPNDIA   See table on page 372

P017 OSS decoding error: &XPPDU, &XP1DIA, &XP2DIA, &XP3DIA

This message is issued if OSS detects an error on decoding a TP PDU, CCR PDU or user data PDU. The insert &XPPDU indicates the type of PDU in question.

The inserts have the following meaning:

XPPDU	Meaning
0	Unknown PDU type (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
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

XPPDU	Meaning
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	Meaning
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   Corresponding index in the syntax table

P018   FSM protocol error: &ACPNT, &OSLPAP, &XPPTYP, &XPFSMN

This message is issued when the finite state machine reports an error.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPPTYP   Type of the service protocol element

&XPFSMN   Name of the finite state machine

P019   APDU contains invalid value: &ACPNT, &OSLPAP, &XPAPDU, &XP3INFO

This message is issued if an invalid APDU is received.

The inserts have the following meaning:

&ACPNT   Name of the local ACCESS-POINT

&OSLPAP   Name of the partner in the local application

&XPAPDU   Type of the APDU

&XP3INFO   Supplementary information on the error

P020   OTRACE implicitly switched off. Reason: &XPTRFAIL

This message is issued when an attempt to write a trace record fails. The OSS trace is deactivated implicitly as a result of the error. After the error has been corrected, the administrator can reactivate the OSS trace.

The inserts have the following meaning:

XPTRFAIL	Meaning
1	The OSS function o_wutr() issued the return code O_ERROR. The preceding P001 message provides further information on the error.
2	The OSS function o_wutr() issued the return code O_INVEREF.
3	The OSS function o_wutr() issued an unknown return code.

P021 Unexpected event &XPEVT occurred for association, event discarded: &ACPNT, &OSLPAP, &XPOSAS, &XPASST

This message is output if an event occurs which does not match the current status of the association. XAPTP does not take account of this event.

The inserts in this message have the following meaning:

&XPEVT   Type of event that has occurred.

&ACPNT   Name of the local access point.

&OSLPAP   Name of the partner in the local application.

&XPOSAS   Index of the relevant association.

&XPASST   Status of the relevant association.

5.2.1   General inserts for the XAP-TP messages

XPRET	Meaning
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. are invalid
-5	Invalid waiting point reference
-6	Invalid application reference
-7	Waiting period to obtain a lock on a shared association expired

XPERR	Meaning
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	Meaning
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	Meaning
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	Meaning
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	Meaning
0	NO_REJECT
1	APPLICATION_CONTEXT_NAME_TOO_LONG The object identifier received from the partner, which forms the application context name, contains more elements than supported by UTM.
2	CALLING_APT_TOO_LONG A length was specified for the application process title in the association indication which is not supported by UTM.
3	CALLING_AEQ_TOO_LONG A length was specified for the application entity qualifier in the association indication which is not supported by UTM.
4	CALLED_APT_TOO_LONG The application process title which was called is longer than that supported by UTM.
5	CALLED_AEQ_TOO_LONG The application entity qualifier which was called is longer than that supported by UTM.
6	CONTEXT_DEFINITION_LIST_TOO_LONG More abstract syntaxes were passed for the association indication than are supported by UTM.

XPRJCT	Meaning
7	CONTEXT_RESULT_LIST_TOO_LONG The list of supported abstract syntaxes passed when establishing an association (association indication or confirmation) contains more elements than are supported by UTM.
9	ADDRESS_NO_PSAP_INFO The address passed for association indication or confirmation does not contain any information on PSAP.
10	ADDRESS_NO_INFO_VERS_0_PSAP The address passed for association indication or confirmation contains an incorrect version for the PSAP information.
11	ADDRESS_INVALID_P_SEL_LENGTH The address passed for association indication or confirmation contains an invalid length for the presentation selector.
12	ADDRESS_NO_SSAPINFO The address passed for association indication or confirmation does not contain any information on SSAP.
13	ADDRESS_NO_INFOVERS_0_SSAP The address passed for association indication or confirmation contains an incorrect version for the SSAP information.
14	ADDRESS_INVALID_S_SEL_LENGTH The address passed for association indication or confirmation does not contain a valid part for the session selector.
15	ADDRESS_NO_PARTNER_MODE The address passed for association indication or confirmation does not contain a valid part for the network and transport selector.
16	ADDRESS_TNSX_ERROR The address passed for association indication or confirmation has been rejected by TNS.
17	UNKNOWN_PARTNER The address passed for association indication or confirmation is not known in the local application.
18	WRONG_APPLICATION_CONTEXT_NAME The application context passed for association indication or confirmation does not correspond to the application context name generated in the local application.
19	ABSTRACT_SYNTAX_MISSING The association indication or confirmation supports less abstract syntaxes than are generated in the local application.
20	OSITP_SYNTAX_MISSING The association indication or confirmation does not support the abstract syntax for OSI TP.

XPRJCT	Meaning
21	NO_TP_INITIALIZE No TP-INITIALIZE-RI/RC PDU was passed with the association indication or confirmation.
22	OSITP_NO_VERSION_1 The partner does not support Version 1 of the OSI TP protocol.
23	OSITP_RCH_WRONG_LENGTH The recovery context handle passed with the TP-INITIALIZE-indication or TP-INITIALIZE-confirmation is of a length not supported by UTM.
24	NO_CCR_INITIALIZE The CCR-INITIALIZE-RI PDU is missing.
25	CCR_NOT_VERSION_2 The partner does not support Version 2 of the CCR protocol.
26	SESSION_NO_FDX Session functionality "full duplex" has not been set.
27	SESSION_NO_DATA_SEPARATION Session functionality "data separation" has not been set although CCR is in the context.
28	SESSION_NO_TYPED_DATA Session functionality "typed data" has not been set although CCR is in the context.
29	SESSION_NO_MINOR_SYNCHRONIZE Session functionality "minor synchronize" has not been set although CCR is in the context.
30	SESSION_NO_RESYNCHRONIZE Session functionality "resynchronize" has not been set although CCR is in the context.
31	TOKEN_CONTENTION_WINNER_AND_NO_TOKEN The local application is the contention winner, but does not possess the "token" (only if CCR is in the context).
32	TOKEN_CONTENTION_LOSER_AND_TOKEN The local application is the contention loser, but possesses the "token" (only if CCR is in the context).
33	INITIAL_SYNC_POINT_SERIAL_NUMBER_NOT_SET The initial syncpoint serial number is not set, although CCR is in the context.
34	NO_MORE_CONTENTION_LOSER_ASSOCIATIONS The request to establish an association from outside is rejected because all the contention loser associations have already been established in the local application.
35	NO_MORE_CONTENTION_WINNER_ASSOCIATIONS The request to establish an association from outside is rejected because all the contention winner associations have already been established in the local application.

XPRJCT	Meaning
36	CCR_BUT_NO_PARTNER_AET The partner did not specify an application entity title, although CCR is in the context.
37	CCR_BUT_NO_OWN_AET No application entity title is specified in the local application, although CCR is in the context.
38	RESPONDING_APT_TOO_LONG The application process title specified in the association confirmation is longer than that supported by UTM.
39	RESPONDING_AEQ_TOO_LONG The application entity qualifier specified in the association confirmation is longer than that supported by UTM.
40	ASS_ESTABLISHMENT_TIMEOUT The establishment of an association started by the local application cannot be completed in the specified time.
41	PARTNER_IS_IN_QUIET_STATE The request for establishment of an association will be rejected because the partner in the local application has been set to *Quiet*.
42	NO_SPACE_FOR_RCH The *PutElement* call for storing the recovery context handle returned a bad value.
43	REMOTE_AET_2_BIG The application entity title of the partner is longer than that supported by UTM.
44	REMOTE_AET_CHANGED When establishing parallel associations to a partner the partner did not provide the same application entity title as for the first association established.
45	NO_SPACE_FOR_REMOTE_AET The *PutElement* call for storing the application entity title of the partner returned a bad value.
46	PARTNER_HAS_STATUS_OFF Establishment of the association is rejected because the partner is locked in the local UTM application (STATUS=OFF set).

XPNDIA	Meaning
0	NO_REASON_GIVEN
1	NO_COMMON_ACSE_VERSION The partner rejected the request to establish an association because there is no common ACSE version.
2	APPL_CONTXT_NAM_NOT_SUPPORTD The partner rejected the request to establish an association because it does not support the application context name.
3	CALLING_AP_TITLE_NOT_RECON The partner rejected the request to establish an association because the sender is not generated correctly at the partner (e.g. incorrect N-SEL). or (for heterogeneous connections only): The partner rejected the request to establish an association because it does not know the calling application process title.
4	CALLING_AE_QUALI_NOT_RECON The partner rejected the request to establish an association because it does not know the calling application entity qualifier.
5	CALLING_AP_INVOC_ID_NOT_RECON The partner rejected the request to establish an association because it does not know the calling application process invocation identifier.
6	CALLING_AE_INVOC_ID_NOT_RECON The partner rejected the request to establish an association because it does not know the calling application entity invocation identifier.
7	CALLED_AP_TITLE_NOT_RECON The partner rejected the request to establish an association because it does not know the called application process title.
8	CALLED_AE_QUALI_NOT_RECON The partner rejected the request to establish an association because it does not know the called application entity qualifier.
9	CALLED_AP_INVOC_ID_NOT_RECON The partner rejected the request to establish an association because it does not know the called application process invocation identifier.
10	CALLED_AE_INVOC_ID_NOT_RECON The partner rejected the request to establish an association because it does not know the called application entity invocation identifier.
11	PERMANENT_FAILURE The partner cleared the association because a permanent error occurred.
12	BEGIN_TRANSACTION_REJECT The partner cleared the association because it rejected the start of a transaction.

XPNDIA	Meaning
13	TRANSIENT_FAILURE The partner cleared the association because a temporary error occurred.
14	PROTOCOL_ERROR The partner cleared the association because a protocol error occurred.
15	UNRECOGNIZED_PDU The association was cleared from outside with P-ABORT because the presentation layer received an unknown presentation PDU.
16	UNEXPECTED_PDU The association was cleared from outside with P-ABORT because the presentation layer received an unexpected presentation PDU.
17	UNEXPECTED_SESSION_SERVICE_PRIMITIVE The association was cleared from outside with P-ABORT because the session layer received an unexpected session service primitive.
18	UNRECOGNIZED_PDU_PARAMETER The association was cleared from outside with P-ABORT because the presentation layer received an unknown PPDU parameter.
19	UNEXPECTED_PPDU_PARAMETER The association was cleared from outside with P-ABORT because the presentation layer received an unexpected PPDU parameter.
20	INVALID_PPDU_PARAMETER_VALUE The association was cleared from outside with P-ABORT because the presentation layer received an invalid PPDU parameter value.
21	RELEASE_NORMAL The association was cleared by the partner with release. The partner specified release normal as the reason.
22	RELEASE_URGENT The association was cleared by the partner with release. The partner specified release urgent as the reason.
23	RELEASE_USER_DEFINED The association was cleared by the partner with release. The partner specified user defined as the reason
24	IDLE_TIMEOUT_ABORT The association was cleared by the local application because the association was not used in the time generated with IDLETIME.
25	ABORT_BY_ASS_INDICATION The association was terminated by the local UTM application because a request to establish an association was received and no suitable, non-established association was available.

5.3 Messages from the KDCDEF generation tool

K400 KDCDEF &DEFTYP &VERS started

K401 Please enter control statements

K402 Syntax error: &SYNERR.

K403 Illegal value entered for parameter &PARAM15.

K404 DMS error &ERROR for file &FNAM.

K405 &ERROR error.

K406 Illegal value entered for operand &PARAM15. &PARVAL8 assumed

K407 Unable to create entry for user "&PARVAL8" in cluster user-file

K408 &STMTNAME name &UTMNAME must not start with "KDC".

K409 LTERM parameter is mandatory for PTERM statements with PTYPE=APPLI, SOCKET or UPIC-R.

K410 Warning: LTERM is assigned to a PTERM with PTYPE=APPLI, SOCKET or UPIC-R and an illegal USER or USAGE=O was specified.

K411 Mandatory parameter &PARAM15 has not been declared in any &PARAM15 control statement.

K412 DESTADM &UTMNAME is not defined as asynchronous TAC or as LTERM. No DESTADM assumed

K413 Ambiguous definition of &STMTNAME &OPERAN32 &STRING64

K414 No LSES control statement assigned to LPAP &UTMNAME.

K415 Illegal parameter for MAX DPUTLIMIT1 or DPUTLIMIT2. Default assumed

K416 &PARAM30 defined more than once.

K417 &PARAM10 but type is not "&PARAM30".

K418 CID parameter is mandatory for a PTERM controlled by a printer control terminal.

K419 Printer pool defined but type is not "OUTPUT".

K420 LTERM with USAGE=D must not be assigned to a PTERM with USAGE=O.

K421 &PARAM30 and &PARAM50 are mutually exclusive.

K422 Mandatory parameter &PARAM10 has not been specified.

K423 Warning: size page pool (PGPOOL) is not greater than size LPUT buffer (LPUTBUF).

K424 Invalid key value: &KEYVALUE. Valid key values are: 0 <= key <= MAX KEYVALUE.

K425 &STMTNAME name &PARAM32 has not been defined.

K426 &STMTNAME control statement missing

K427 Event exit &EXITNAME has already been defined.

K428 Invalid CARD parameter: &CARDINFO or POSITION = 0.

K429 Too many database types

K430 Too many &STMTNAME control statements

K431 User "&PARVAL8" from cluster user-file is missing in KDCFILE.

K432 No &STMTNAME with &PARAM32 .

K433 Filebase name is too long. The configuration run will be aborted.

K434 Minimum length of buffer for recovery data, RECBUF parameter, is 1024. 1024 assumed

K435 The event exit &PARAM8 must be assigned to an ASSEMBLER program unit.

K436 MAX statement is missing. The configuration run will be aborted.

K437 Program error in KDCDEF; reason: &TERMREAS.

TERMREAS describes the cause that lead to the abnormal termination of the utility.

The description of UTM message K060 also helps to clear the error.

Action: Proceed as described for K060.

In the other cases, there is an error in KDCDEF. Store the documents and write a PR.

K438 More LSES statements than CON statements have been assigned to LPAP &UTMNAME.

K439 In parameter KDCFILE the file base name was specified with a catalogue ID.

K440 &PARAM8 is not allowed in &STMTNAME statements with &PARAM10.

K441 SHARE table &UTMNAME is empty.

K442 Inclusion of &PARAM8 is only meaningful if the &STMTNAME is assigned to a &PARVAL8 program unit.

K443 Inclusion of ULS control statements is only meaningful if USER statements were generated.

K444 OPTION control statement is only effective when reading from SYSDTA.

K445 &PARAM30 no longer supported.

K446 Parameter ignored.

K447 Control statement ignored.

K448 KDCFILE generated with warnings; KAA size: &KAASIZE K

K449 There was at least one ERROR. Configuration run aborted.

K450 KDCFILE generated; KAA size: &KAASIZE K

K451 File &FNAM generated.

K452 Too many &PARAM10 names. The configuration run will be aborted.

K453 BCAMAPPL name different from APPLINAME is only allowed for a &STMTNAME with PTYPE=APPLI, SOCKET or UPIC-R.

K454 Length of LTERM parameter &UTMNAME incompatible with NUMBER parameter.

K455 TAC assigned to STACK parameter may not be defined with TYPE=A or CALL=NEXT.

K456 It is not possible to assign a group-LTERM to a PTERM

K457 Warning: &PARAM10 larger than &PARAM10A

K458 LTERM &PARAM8 has been assigned as CTERM for LTERM &PARAM8A. "USAGE = OUTPUT" is not meaningful.

K459 The LTERM specified at GROUP must not be a group-LTERM itself.

K460 File &FNAM processed.

K461 &STMTTP30 &STMTNM30 not supported by &UTMVAR

K462 Only one OSI-CON connection group is allowed to be active with local partner application &PARAM8.

K463 &PARAM30 set to &PARAM30.

K464 Length of session names + length of associations > 8.

K465 &STMTNAME control statement is mandatory if &PARAM30 is generated.

K466 Parameter &PARAM30 is mandatory if &PARAM30 is generated.

K467 Error: Multiple entries for user "&PARAM8" found in cluster user-file

K468 Password parameter does not fit security level.

K469 Parameter &PARAM32 only allowed if APPLIMODE = SECURE.

K470 No user is able to call KDCSHUT.

K471 &OPERAND&PARAM10 already set.

K472 &PARAM32 too long.

If the Principal parameter is too long then the user is generated without a Principal entry.

K473 Parameter &PARAM10 used more than once for &PARAM30

K474 Parameter &PARAM30 is illegal if &PARAM30 is generated.

K475 Parameter &PARAM30 not needed if &PARAM30 is generated.

K476 In &STMTNAME control statements &OPERAND is given more than once for PRONAM=&PARAM8 and BCAMAPPL=&PARAM8 .

K477 Parameter &PARAM30 is illegal if any &PARAM30 is given.

K478 Control statement &STMTNAME is illegal if &PARAM30 is given.

K479 &STMTNAME &PARAM50 not used.

K480 Too many &STMTNAME control statements for this &PARAM30.

K481 The primary LTERM of a group must not be slave of a bundle.

K482 EXIT and TAC must be in the same LOAD-MODULE if the exit program is referencing a LOAD-MODULE with LOAD-MODE=ONCALL.

K483 &PARAM32 must not reference a &STMTNAME with &PARAM32 .

K484 &PARAM32 with &PARAM32 not generated.

K485 Control statement too long.

K486 Different libraries given for same DATABASE TYPE.

K487 Cannot assign a master LTERM of bundle to PTERM.

K488 All slave LTERMs of bundle must be assigned to PTERMs with identical PTYPE (APPLI or SOCKET).

K489 Master LTERM of bundle must not be a slave itself.

K490 Master LTERM must be generated with QAMSG=YES and RESTART=YES. Parameters changed.

K491 &OPERAND&PARAM32&PARAM20 will not be supported in the next UTM version.

K492 Note: multiple ACCESS-POINT statements may cause problems.

K493 The creation of the TNS input file failed.

K494 Operand &PARAM30 is mandatory if &STMTNAME control statement is given more than once.

K495 Standard message module not found in any MESSAGE control statement or not defined by MAX control statement.

K496   File &FNAM corrupted or not a KDCFILE.

K497   KDCDEF &VER1 / &OST1 / &BMD1 cannot read KDCFILE &VER2 / &OST2 / &BMD2

K498   There are no &STMTNAME control statements to be created.

K499   &PARAM30 in &STMTNAME control statement and &PARAM30 do not match.

K500   No file(s) generated.

K501   Illegal character "&CHAR1" in string: &STRING64

K502   Number of free &PARAM10 entries:&PARAM11

K503   The referenced ACCESS-POINT &PARAM8 must include an APPLICATION-
           ENTITY-QUALIFIER because the referenced OSI-LPAP &PARAM8 includes an
           APPLICATION-CONTEXT with the ABSTRACT-SYNTAX CCR.

K504   OSI-CON &PARAM8 references same OSI-LPAP but different ACCESS-POINT.

K505   The operand &PARAM32 is mandatory if the referenced APPLICATION-CONTEXT
           includes the ABSTRACT-SYNTAX CCR.

K506   An APPLICATION-CONTEXT including the ABSTRACT-SYNTAX CCR must not be
           used because the UTMD control statement does not specify an APPLICATION-
           PROCESS-TITLE.

K507   Too many &PARAM32

K508   Error occurred during generation of RSA-Keys.

K509   OSI-CON &PARAM8 references same LOCAL-ACCESS-POINT and an OSI-LPAP
           with identical APPLICATION-ENTITY-QUALIFIER and APPLICATION-PROCESS-
           TITLE.

K510   Operand &PARAM32 requires at least &PARAM8 parameters.

K511   At most &PARAM11 concurrent stacked services will be possible.

K512   At most &PARAM11 concurrent sign on and/or stacked services will be possible.

K513   KDCDEF will now generate new RSA-Keys. Depending on key length and system
           configuration this process can take a considerable amount of time.
           Please wait ...

K514   Generation of RSA-Keys completed

K515 Warning: For proper operation the application needs RSA-keys. However, no RSA-keys have been generated.

If objects are generated with encryption levels, the application requires RSA keys in order to run correctly. If no RSA keys are available in the application, the application can be operated but with certain restrictions. TACs with encryption levels cannot be called and no connection can be set up to PTERMs or TPOOLs with encryption levels.

Three responses are possible.

– The RSA keys can be transferred from an old KDCFILE to the new KDCFILE using KDCUPD before the application is started, or

– the KDCDEF run can be repeated with the GEN-RSA-KEYS=YES operand in the OPTION statement, or

– the application can be started and the required RSA keys can be created and activated by administration (e.g. using WinAdmin or WebAdmin).

K516 Error &DIAG1 for cluster user-file &STRING64

The insert &DIAG1 describes the cause of the error; the insert &STRING64 contains the file name.

The insert &DIAG1 can have the following values:

Value	Meaning
1	OPEN_ERROR Error opening the file.
2	CLOSE_ERROR Error closing the file.
3	FILE_ERROR The file is inconsistent.
4	LOCK_ERROR Error requesting the file lock.
5	UNLOCK_ERROR Error releasing the file lock.
6	USER_NOT_FOUND KCCUFUE sends an incorrect return code on an update to a user entry; the second insert contains the name of the user
7	UNEXPECTED_RETURNCODE An unexpected return code was returned by a called function; the second insert contains the value of the return code
8	KDCDEF_RUNNING The cluster user file is already locked by a KDCDEF run

Value	Meaning
9	FILE_IS_EMPTY The cluster user file is cataloged but empty
10	Error on getLocalHostName call The second insert contains the error code.
11	NO_XCS_GROUP The computer does not belong to an XCS cluster; the "distributed lock" function is not therefore available.

K517 &PARAM30 not supported for &PARAM30

K518 File &FNAM already exists. The configuration run will be aborted.

K519 LMS library element generated
library: &FNAM
element: &ELEM
version: &EVER
type: &LMSTYPE

Meaning of the inserts:

&FNAM   Name of the PLAM library to which the message refers.

&ELEM   Name of the LMS element to which the message refers.

&EVER   LMS version of the LMS element.

&LMSTYPE   LMS type of the LMS element.

K520 LMS error (PLA&PLARC, DMS&DMSRC) for
library: &FNAM
element: &ELEM
version: &EVER
type: &LMSTYPE

Meaning of the inserts:

&PLARC   PLAM return code. The help text can also be queried using /HELP
             PLA&PLARC.

&DMSRC   DMS return code. The help text can also be queried using /HELP
             DMS&DMSRC.

&FNAM   Name of the PLAM library to which the message refers.

&ELEM   Name of the LMS element to which the message refers.

&EVER   LMS version of the LMS element.

&LMSTYPE   LMS type of the LMS element.

K521   MAX PRIVILEGED-LTERM=&UTMNAME is not defined as Dialog-LTERM.

K522   Note: No privileged LTERM generated (see MAX PRIVILEGED-LTERM=)

5.4 Messages from the UTM tools KDCCSYSL and KDCPSYSL

K600 Program &PRGNMSG &VERS started.

K601 Program KDCCSYSL terminated normally - SYSLOG file converted

K602 Program KDCPSYSL terminated normally - SYSLOG file edited

K603 Program KDCCSYSL terminated abnormally - SYSLOG file not converted

K604 Program KDCPSYSL terminated abnormally - SYSLOG file not edited

K605 &ERROR error

K606 File with link name &LINK is not a SYSLOG file.

K607 DMS error &DMSE on file with link name &LINK

K608 Please enter name of user message module, a blank, or /EOF.

K609 User message module assumed

K610 System message module assumed

K611 Program error in &PRGNMSG; reason: &TRMA

K612 Incorrect number of messages in user message module

K613 Incorrect message &MSGNR in user message module

5.5   Messages of the UTM tools KDCMMOD / KDCMTXT

K650   Program &PRGNMSG terminated abnormally.

K651   Program error (&TRMA).

Action: Write PR.

K652   DMS error &DMSE on file &FNAM.

K653   Line &LINENR: mandatory operand missing.

K654   Line &LINENR: syntax error.

K655   Line &LINENR: insert &INSMSG is not permitted for message &IDMSG.

K656   Line &LINENR: constant &CONMSG not defined.

K657   EOF reached on SYSDTA. END statement generated.

K658   * * * * * Statement ignored. * * * * *

K659   File &FNAM is not a valid message definition file.

K660   Program &PRGNMSG &VERS started.

K661   Program &PRGNMSG terminated normally.

K662   Line &LINENR: function unit &FUMSG not defined.

K663   Line &LINENR: language &LANGMSG not defined for function unit &FUMSG.

K664   Line &LINENR: message &IDMSG is not permitted for function unit &FUMSG.

K665   Line &LINENR: GEN statement already entered.

K666   Line &LINENR: message destination &DESTMSG is not permitted for message &IDMSG.

K667   Line &LINENR: message destination &DESTMSG is required for message &IDMSG.

K668   Line &LINENR: GEN statement missing.

K669   Line &LINENR: function unit &FUMSG may not be modified.

K670   Source file &FNAM for message module created.

K671   Source file for message module not created.

K672   Line &LINENR: constant &CONMSG already defined.

K673   Wrong version &VERS of message definition file &FNAM.

K686 Line &LINENR: text for message &IDMSG longer than &MSGMAXL.

The text of the message including the insert is longer than 512 characters. The utilities KDCMMOD and KDCMTXT cannot process the text. KDCMMOD does not generate a source for a new message module. KDCMTXT does not amend the message definition file.

K687 Line &LINENR: warning - text for message &IDMSG (SYSLINE) longer than &MSGMAXL.

The text of the message (&IDMSG) with the destination SYSLINE including the insert is longer than 80 characters. The utilities KDCMMOD and KDCMTXT accept the message text. On subsequent output of the message in the system line, UTM outputs only the first 80 characters of the message text.

K688 Line &LINENR: message number > 999 not allowed.

A message number greater than 999 was specified. The utilities KDCMMOD and KDCMTXT reject this. KDCMMOD does not generate a source for a new message module. KDCMTXT does not amend the message definition file.

K690 Message text for message &IDMSG in language &LANGMSG and function unit &FUMSG not defined.

K691 Message definition file &FNAM generated.

K692 Message definition file not generated.

K693 Message definition file &FNAM modified.

K694 Message definition file not modified.

K695 Line &LINENR: insert &INSMSG not defined.

K696 Line &LINENR: FU/constant/insert &CONMSG already defined.

K697 Line &LINENR: illegal value for SOURCE length, default value assumed.

K698 Line &LINENR: range already used for another function unit.

5.6 **KDCDUMP messages**

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 Error on calling EDT. EDT call : &EDTINT Return code: &EDTRET

This message is issued when EDT reports an error. The insert EDTINT shows which subprogram interface of EDT reported the error (INF=IEDTINF or CMD = IEDTCMD).
The insert EDTRET contains the EDT return code in the following sequence: sub-return code 1 (1 byte), sub-return code 2 (1 byte), main return code (2 bytes).
Refer to the EDT documentation for a description of the return codes.

K708 EDT version not supported by KDCDUMP. EDT version: &EDTVER

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.

   This message indicates that the UTM dump creator was unable to write the UTM dump completely. There may have been insufficient space available under the ID. The UTM areas are dumped in the following order:

   KAA, SLOT, CACHE, MPGP, XAPTP-GLOBAL, KTA, XAPTP-LOCAL, STACK, ROO,T REP, User File, Journal File 1, Journal File 2, Buffer Segments, GSSB File, Lock File, CFG File, ULS File

   In the case of a PENDER dump, only ROOT and REP are dumped.

   The individual tables for the UTM areas can be output by means of the KDCDUMP statement HELP TABLE-NAMES (page 72); they appear in the order of the dump. The XAPTP sections form a block here.

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.

K738   No REP file is contained in UTM dump.

K739   No REP records (PM-NO: &PMNO) found.

K740   Abbreviation of table name ambiguous with regard to &TABNAM &TABNAM1 &TABNAM2 &TABNAM3 &TABNAM4 &TABNAM5.

K741 Information not available. Table name: &TABNAM

This message indicates that a UTM area or a table could not be dumped by the UTM dump creator because the UTM area or the UTM table was not fully accessible.

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.

Recommendation: Read domain with `FILE=<dumpfile>,DOMAIN=<domain>` and repeat the SFIND command.

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 Messages from the UTM tool KDCUPD

The UTM tool KDCUPD outputs both transaction monitor messages and its own messages. With all KDCDUP messages where the text begins with an "*" (asterisk), it was not possible to transfer the data.

K800 KDCUPD &VERS started

K801 Please enter parameters

K802 Control statement KDCFILE: parameter &UPDCMD is not specified

K803 Base names of old and new &PARAM17 must be different

K804 Transfer from UTM &UPDVERS to UTM &UPDVERS not supported

K805 Consistency check for filebase &FBASUPD okay

K806 All requested data transferred

K807 Requested data partially transferred

K808 No data to transfer from &FNAM!

K809 Control statement KDCFILE/CHECK: base name &FBASUPD is too long

K810 CHECK and KDCFILE commands cannot be combined

K811 Control statement KDCFILE: file base must be specified without a cat ID if CATID control statement is used

K812 KDCUPD abnormal end

K813 KDCUPD normal end

K814 CLUSTER-Transfer of different versions not supported

K851 &UPDTYP data transferred. KCRN = &UKCRN,&PARAM4 = &UKCLA.

K852 &PARAM3 data transferred. KCRN = &UKCRN, &PARAM4 = &UKCLT, KCLA = &UKCLA

K853 Database configuration. number: &DBCOUNT FILEOLD: &DBOLD FILENEW: &DBNEW

K854 Warning: Security level of user &USER increased. Password may be invalid.

Meaning: The complexity level of the password for the USER &USER is higher in the new KDCFILE than in the old KDCFILE, see USER statement in the KDCDEF generation. If the password transferred does not satisfy this condition, the USER can then no longer sign on.

Action: The administrator must then issue a new password.

K855   * &UPDTYP data not transferred. KCRN = &UKCRN, &PARAM4 = &UKCLA,
           KCRCCC = &RCCC, KCRCDC = &RCDC.

K856   * &PARAM3 data not transferred. KCRN = &UKCRN, &PARAM4 = &UKCLT, KCLA
           = &UKCLA, KCRCCC = &RCCC, KCRCDC = &RCDC

K857   * Load module &PROG not found. Current version &PVER not transferred.

K858   Current version &PVER of load module &PROG transferred.

5.8   Standard message definition file, inserts

5.8.1   Standard message definition file constants

Constant name	Constant value	Remark
DAR	12	DARK LAYOUT
DEL	07	DELETE
EM3	13	EMPHASIZED LAYOUT
EPA	08	END PROTECTED AREA
NL	15	NEW LINE
NOR	1E	NORMAL LAYOUT
NP	0C	NEW PAGE
SPA	36	START PROTECTED AREA

5.8.2   Inserts in messages

The "Length" column contains the output length of the individual inserts in bytes, i.e. on output of the message text the insert occupies as many characters as specified in the "Length" column. The length of the inserts is significant particularly for the creation of message texts using KDCMMOD and KDCMTXT.

The abbreviations in the 'data type' column have the following meanings:

Char   =   printable characters
Int   =   numeric field
Hexa   =   hexadecimal information

Insert name	Data type	Length	Meaning
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
ANNO	Hexa	64	ANNO RECEIVED
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	BCAM 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
CAMODE	Hexa	2	CONTXT MACRO: ADDRESS MODE
CBRC	Hexa	8	VTSUCB RETURN CODE
CCC	Hexa	2	CONTXT MACRO: CONDITION CODE IN PCR FORMAT
CCSN	Char	8	CCSNAME
CDSPL	Char	19	CONTEXT MODULE + DISPLACEMENT

Insert name	Data type	Length	Meaning
CHAIN	Char	3	CHAINED MESSAGE INFORMATION
CID	Char	8	PRINTER CONTROL ID
CILC	Hexa	2	CONTXT MACRO: INSTRUCTION LENGTH IN PCR FORMAT
CIW	Hexa	2	CONTEXT INTERRUPT WEIGHT
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
CPC	Hexa	8	CONTEXT PROGRAM COUNTER
CPM	Hexa	2	CONTXT MACRO: PROGRAM MASK IN PCR FORMAT
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
CPUTEXT	Char	8	TEXT: OVERFLOW OR NEGATIV
CPUUSED	Hexa	8	CPU TIME USED OF THIS TAC IN MILLISECONDS
CR0	Hexa	8	CONTXT MACRO: REGISTER R0
CR1	Hexa	8	CONTXT MACRO: REGISTER R1
CR10	Hexa	8	CONTXT MACRO: REGISTER R10
CR11	Hexa	8	CONTXT MACRO: REGISTER R11
CR12	Hexa	8	CONTXT MACRO: REGISTER R12
CR13	Hexa	8	CONTXT MACRO: REGISTER R13
CR14	Hexa	8	CONTXT MACRO: REGISTER R14
CR15	Hexa	8	CONTXT MACRO: REGISTER R15
CR2	Hexa	8	CONTXT MACRO: REGISTER R2
CR3	Hexa	8	CONTXT MACRO: REGISTER R3
CR4	Hexa	8	CONTXT MACRO: REGISTER R4
CR5	Hexa	8	CONTXT MACRO: REGISTER R5

Insert name	Data type	Length	Meaning
CR6	Hexa	8	CONTXT MACRO: REGISTER R6
CR7	Hexa	8	CONTXT MACRO: REGISTER R7
CR8	Hexa	8	CONTXT MACRO: REGISTER R8
CR9	Hexa	8	CONTXT MACRO: REGISTER R9
CTYP	Char	4	TYPE OF PROGRAM EXCHANGE
DBCALL	Char	12	FUNCTION-CALL OF IUTMDB INTERFACE
DBCON	Char	8	DATABASE CONNECTION MODULE
DBTRAC	Hexa	8	DB TRACE INFO
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
DMSRC	Hexa	4	DMS RETURN 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
ELEM	Char	64	ELEMENT NAME
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)

Insert name	Data type	Length	Meaning
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
ESQS	Hexa	8	SAVED SEQUENCE NUMBER
ESRR	Hexa	10	ACTUAL RESPONSE SEQUENCE NUMBER
EVER	Char	24	ELEMENT VERSION
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	BCAM INFOWORD
FIL3	Hexa	4	PTERM STATE
FMH7	Char	80	ERROR RECOVERY PROCEDURE MESSAGE
FMTN	Char	8	FORMAT NAME
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
GNUSER	Char	8	USER NAME OF ASYNCHRON. MESSAGE GENERATION
GTRID	Hexa	128	FIRST 64 BYTE OF GLOBAL TRANSACTION ID
HITR	Char	3	CACHE HIT RATE
HSTACK	Int	2	HEIGHT OF STACK
HST1	Char	8	HOST NAME
HST2	Char	8	HOST NAME
HST3	Char	8	HOST NAME
IDEFRC	Hexa	16	RETURNCODE OF INVERSE KDCDEF

Insert name	Data type	Length	Meaning
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
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
KRACT	Int	1	ACTION CODE OF KERBEROS INTERFACE KRBGSS
KRLH	Int	4	LENGTH OF KERBEROS PRINCIPAL
KRPR	Char	100	KERBEROS CLIENT OR SERVER PRINCIPAL
KRRC	Hexa	8	RETURN CODE OF KERBEROS INTERFACE KRBGSS
LMSTYPE	Char	8	LMS TYPE
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
MXPR	Char	8	MUX PROCESSOR

Insert name	Data type	Length	Meaning
MXPT	Char	8	MUX PTERM
MXP1	Char	4	MUX PROTOCOL VERSION (LOWER BOUNDARY)
MXP2	Char	4	MUX PROTOCOL VERSION (UPPER BOUNDARY)
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
NUMMSGS	Int	11	NUMBER OF WAITING OUTPUT MESSAGES
OBJ1	Char	10	OBJECT NAME
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
OMSG3	Hexa	64	FIRST PART OF OUTPUT 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

Insert name	Data type	Length	Meaning
PID	Int	11	BS2000 TASK ID (ITN)
PLARC	Char	4	PLAM RETURN CODE
PRCN	Char	200	PROCEDURE/SCRIPT/COMMAND-FILE NAME
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
RCF1A	Char	4	KCRCDC
RCF1B	Char	3	RETURN CODE 1
RCF1C	Char	4	RETURN CODE 1
RCF2A	Char	4	SECONDARY FHS/VTSU RET 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

Insert name	Data type	Length	Meaning
RSOANNO	Hexa	64	RSO ANNO
RSOMSG	Char	7	RSO ERROR MESSAGE
RSOOPC	Hexa	2	RSO ACTION
RSORC	Hexa	8	RSO RETURNCODE
RSO2RC	Hexa	8	RSO ASYN RETURNCODE
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
SNLT	Char	8	MUX SESSION LTERM NAME
SNPR	Char	8	MUX SESSION PROCESSOR NAME
SNPT	Char	8	MUX SESSION PTERM NAME
SRFG	Hexa	8	SAVED SESSION STATE
SSINCID	Hexa	8	SSO CALL CID
SSLENNAM	Hexa	4	SSO NAME LENGTH
SSORC	Hexa	2	SSO RETURNCODE
SSOUTCID	Hexa	8	SSO RETURN CID
SSTYPNAM	Hexa	8	SSO NAME TYPE
STATE	Char	1	STATE
STA2	Char	1	STATE
STDHEAD	Hexa	16	BS2000 STANDARDHEADER
STDRC	Hexa	8	BS2000 STANDARDHEADER-RETURNCODE
STMT	Char	11	STATEMENT OF KDCDEF
STRTPAR	Char	8	STARTPAR LTH = 8
STSK	Char	1	SYSTEM TASK (Y/N)
STS1	Hexa	4	STSN-REQ SEQUENCE NUMBER RCV-CNT
STS2	Hexa	4	STSN-REQ SEQUENCE NUMBER SEND-CNT
STS3	Hexa	4	STSN-RSP SEQUENCE NUMBER SLU-PLU
STS4	Hexa	4	STSN-RSP SEQUENCE NUMBER PLU-SLU
SUFF	Char	5	FILE SUFFIX
SWNR	Int	11	NUMBER OF JOURNAL SWITCHES
SYN	Char	50	SYNTAX ERROR

Insert name	Data type	Length	Meaning
SYSD	Hexa	4	SYSTEM SENSE DATA
SYSFILE	Char	6	SYSOUT/SYSLST
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	TSN OF UTM TASK
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)
TTYP	Hexa	2	TERMINAL TYPE
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
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

Insert name	Data type	Length	Meaning
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
VERS	Char	8	UTM VERSION
VER1	Char	6	VERSION NUMBER
VER2	Char	6	VERSION NUMBER
VGCNTR	Int	11	SERVICE COUNTER OF ACTUAL SERVICE
VTRC	Hexa	8	VTSU OR ASECO 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
XCS1	Char	8	NAME OF XCS GROUP
XCS2	Char	8	NAME OF XCS GROUP
XPAPDU	Char	20	OSI-TP APDU TYPE
XPASST	Char	20	ASSOCIATION STATE
XPBCAM	Hexa	8	BCAM INFOWORD
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

Insert name	Data type	Length	Meaning
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
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
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
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

Insert name	Data type	Length	Meaning
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

5.9 Destinations of UTM messages

The table below shows which options are possible for each message. Only the K messages are considered, which can be modified by the user.

For the individual messages the information in the columns of the table means:

R (required) : The message destination is strictly assigned to the message; it cannot be modified.

+ : The message destination is permitted for this message.

D (default) : The message destination is specified for this message. It can be removed from the message.

- : The message destination is not permitted for this message.

in the column COMPRESS:

Y (yes): superfluous blanks removed from the message.
N (no): superfluous blanks left in the message.

in the column EXTEND:

Y (yes): The message can be output in EXTENDED LINE MODE.
N (no): The message cannot be output in EXTENDED LINE MODE.

in the column BELL:

Y (yes): Acoustic signal output with message.
N (no): No acoustic signal can be output with the message.

All K messages from openUTM that occur during the start phase are output to SYSLIST and SYSOUT, regardless of which destination was specified for these messages, so that you can more easily diagnose errors when starting a UTM application or a follow-up process.

The functional unit can be modified by the user.

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	E x t e n d	B e l l
K001[1]	PTRM , PRNM , BCAP , LTRM , APPL	R	-	+	+	+	+	+	+	-	Y	N	N
K002[1]	PTRM , PRNM , BCAP , LTRM , APPL	R	-	-	+	+	+	+	+	+	Y	N	N
K003	PTRM , PRNM , BCAP , LTRM , CMD	R	D	-	+	+	+	+	+	+	N	N	N
K004[1]	PTRM , PRNM , BCAP , LTRM , USER , REA7	R	+	-	+	+	+	+	+	+	N	N	N
K005[1]	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	+	+	+	+	+	+	N	N	N
K006[1]	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	+	+	+	+	+	+	N	N	N
K007[1]	PTRM , PRNM , BCAP , LTRM , USER , REA7	R	+	-	+	+	+	+	+	+	N	N	N
K008[1]	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	-	Y	N	N
K009	PTRM , PRNM , BCAP , LTRM , USER , TAC , RCDC	R	D	D	+	+	+	+	+	+	N	N	N
K010	PTRM , PRNM , BCAP , LTRM , USER , TAC	R	D	D	+	+	+	+	+	+	N	N	N
K011	PTRM , PRNM , BCAP , LTRM , USER , ATAC1	R	+	+	+	+	+	+	+	+	N	N	N
K012	NUMMSGS	R	R	-	-	-	-	-	-	-	N	N	N
K013	PTRM , PRNM , BCAP , LTRM , CMD	R	D	-	+	+	+	+	+	+	N	N	N
K014	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	+	+	+	+	+	+	N	N	N
K015	PTRM , PRNM , BCAP , LTRM , USER , TAC , FORM , RCDC , RCF2A	R	+	-	+	+	+	+	+	+	N	N	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	E x t e n d	B e l l
K016	PTRM , PRNM , BCAP , LTRM , USER	R	D	+	+	+	+	+	+	+	N	N	N
K017	PTRM , PRNM , BCAP , LTRM , USER , TCVG , RCCC , RCDC , RCF2A , TAC	R	-	D	D	+	+	+	+	+	Y	N	N
K018	PTRM , PRNM , BCAP , LTRM , APPL	R	+	-	+	+	+	+	+	+	N	N	N
K019	PTRM , PRNM , BCAP , LTRM , APPL	R	D	-	+	+	+	+	+	+	N	N	N
K020	PTRM , PRNM , BCAP , LTRM , USER	R	D	-	+	+	+	+	+	+	N	N	N
K021	PTRM , PRNM , BCAP , LTRM	R	D	D	+	+	+	+	+	+	N	N	N
K022	PTRM , PRNM , BCAP , LTRM	R	-	+	+	+	+	+	+	+	Y	N	N
K023	OMSG2	R	D	+	-	-	-	-	-	-	N	N	N
K024	PTRM , PRNM , BCAP , LTRM , USER	R	+	D	+	+	+	+	+	+	N	N	N
K025	PTRM , PRNM , BCAP , LTRM	R	-	D	D	+	+	+	+	+	Y	N	N
K026	PTRM , PRNM , BCAP , LTRM , USER	R	+	+	+	+	+	+	+	+	N	N	N
K027	PTRM , PRNM , BCAP , LTRM	R	+	D	+	+	+	+	+	+	N	N	N
K028	PTRM , PRNM , BCAP , LTRM , USER , PAS1	R	-	-	+	+	+	+	+	-	N	Y	N
K029	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	-	N	N	N
K030[1]	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	+	+	+	+	+	+	N	N	N
K031[1]	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	+	+	+	+	+	+	N	N	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	E x t e n d	B e l l
K032	CON , PRNM , BCAP , LPAP , USER , RCF1B , RCF2B	-	-	-	D	+	D	+	+	+	N	N	N
K033	PTRM , PRNM , BCAP , LTRM , USER , REST , GLOBALSG	-	-	-	D	+	+	+	+	+	N	N	N
K034		R	R	+	-	-	-	-	-	-	N	N	N
K035		R	R	+	-	-	-	-	-	-	N	N	N
K036	PTRM , PRNM , BCAP , LTRM , RSLT , REA1	-	-	-	D	+	+	+	+	+	N	N	N
K038	SYN	-	-	-	-	-	R	+	-	-	Y	N	N
K039	STRTPAR	-	-	-	-	-	R	+	-	-	Y	N	N
K040	WLEV , PGPOOL	-	-	-	D	+	R	+	+	+	Y	N	N
K041	WLEV , PGPOOL	-	-	-	D	+	R	+	+	+	Y	N	N
K042		-	-	-	-	-	R	+	-	-	Y	N	N
K043	DMSE , FNAM	-	-	-	D	+	R	D	+	+	Y	N	N
K044		-	-	-	-	-	R	+	-	-	Y	N	N
K045	PTRM , PRNM , BCAP , LTRM , PALTRM , CID	-	-	-	-	+	-	-	-	+	N	N	N
K046	PTRM , PRNM , BCAP , LTRM , PALTRM , CID , DPID , ERPRT , IMSG2	-	-	-	+	+	+	+	+	+	N	N	N
K047	OPCD1 , RTCD	-	-	-	-	-	R	+	-	-	Y	N	N
K048		-	-	-	-	-	R	+	-	-	Y	N	N
K049	RCCC2	-	-	-	-	-	D	D	-	-	Y	N	N
K050	APPL , VERS , AMOD , TERM , OST1 , BMD1 , ATYP , FNOD	-	-	-	D	+	R	+	+	+	Y	N	N
K051	APPL , VERS , AMOD , TERM , OST1 , BMD1 , ATYP , FNOD	-	-	-	D	+	R	+	+	+	Y	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K052	TASK , APPL , PID , STSK	-	-	-	D	+	R	+	+	+	Y	N	N
K053	CNTR	-	-	-	D	+	R	+	+	+	Y	N	N
K054		-	-	-	D	+	R	+	+	+	Y	N	N
K055	ATAC1 , RCCC , RCDC , USER , LTRM	-	-	-	D	+	R	+	+	+	Y	N	N
K056	TASK , PID	-	-	-	D	+	D	+	+	+	Y	N	N
K057		-	-	-	D	-	D	+	+	-	Y	N	N
K058	TASK , PID	-	-	-	+	+	D	+	+	+	Y	N	N
K059		-	-	-	+	-	D	+	+	-	Y	N	N
K060	TRMA	-	-	-	D	-	R	D	+	-	Y	N	N
K061	FNAM	-	-	-	D	-	R	D	+	-	Y	N	N
K062		-	-	-	D	-	R	D	+	-	Y	N	N
K063	PTRM , PRNM , BCAP , LTRM , FMTN , RCF1A , RCF2A	-	-	-	D	+	R	+	+	+	Y	N	N
K064	PTRM , PRNM , BCAP , LTRM , DEVC , FIL1A , FIL2A , FIL3 , VTRC , IMSG2 , REA1 , CBRC	-	-	-	D	+	+	+	+	+	Y	N	N
K065	PTRM , PRNM , BCAP , LTRM , FIL1B , FIL2B	-	-	-	D	+	+	+	+	+	N	N	N
K066		-	-	-	-	-	R	+	-	-	Y	N	N
K067	MOD , ERCD1 , ERCD2 , OPCD2	-	-	-	-	-	R	+	-	-	Y	N	N
K068	DBCON , DBV1 , DBV2	-	-	-	-	-	R	+	-	-	Y	N	N
K069	PTRM , PRNM , BCAP , LTRM , COTM , REA4 , REA6	-	-	-	D	+	+	+	+	+	N	N	N
K070	PTRM , PRNM , BCAP , LTRM , USER , COTM , CPTM , GLOBALSG	-	-	-	D	+	+	+	+	+	N	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K071	OPCD1 , ERCD3 , DBTRAC , DBCON	-	-	-	-	-	R	+	-	-	Y	N	N
K072	STMT	-	-	-	-	-	D	D	-	-	Y	N	N
K073	ATTR , STMT , PROG	-	-	-	-	-	D	D	-	-	Y	N	N
K074	CTYP , PROG , PVER	-	-	-	D	+	R	+	+	+	Y	N	N
K075	CTYP , PROG , PVER , TASK , PID	-	-	-	D	+	D	+	+	+	Y	N	N
K076	RCCC , RCDC , ADTC , USER , LTRM	-	-	-	D	+	R	+	+	-	Y	N	N
K077	PTRM , PRNM , BCAP , LTRM , CLSIGT	-	-	-	D	-	D	D	-	-	Y	N	N
K078	ERRNAME , ERRCODE , REA3	-	-	-	-	-	R	D	-	-	Y	N	N
K079	REA2	-	-	-	D	+	+	+	+	+	Y	N	N
K080		-	-	-	D	+	+	+	+	+	Y	N	N
K081	IMSG1 , OMSG1 , CONU , ATAC2 , LWRT , HITR , WTBF	-	-	-	D	+	+	+	+	+	N	N	N
K082	FNAM	-	-	-	-	-	R	D	-	-	Y	N	N
K083	FNAM	-	-	-	-	-	R	D	-	-	Y	N	N
K084	OBJ1 , VER1 , OBJ2 , VER2 , OST1 , BMD1 , OST2 , BMD2	-	-	-	-	-	R	D	-	-	Y	N	N
K085	FNKT	-	-	-	-	-	R	D	-	-	Y	N	N
K086	PTRM , PRNM , BCAP , LTRM , USER , SYSD , USSD , FMH7 , AGUS	-	-	-	D	+	+	+	+	+	N	N	N
K087	OPCD3 , ERCD4 , ERCD5	-	-	-	-	-	R	+	-	-	Y	N	N
K088	LSES , RSES , LPAP , SRFG , PSQN , ESQS , EBSS , ESQR , ESRR , EBSR	-	-	-	D	+	+	+	+	+	N	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K089	GNDATE , GNTIME , DEST , GNUSER , USER , DLDATE , DLTIME , CHAIN	-	-	-	D	+	+	+	+	+	N	N	N
K090	DEST , USER , DLDATE , DLTIME	-	-	-	D	+	+	+	+	+	N	N	N
K091	PTRM , PRNM , BCAP , LTRM , USER , REA7	R	+	+	+	+	+	+	+	+	N	N	N
K092	PTRM , PRNM , BCAP , LTRM , USER , PAS1 , PAS2 , PAS3	R	-	-	+	+	+	+	+	-	N	Y	N
K093	PTRM , PRNM , BCAP , LTRM , USER , HSTACK , MSTACK	R	D	+	+	+	+	+	+	+	Y	N	N
K094	PTRM , PRNM , BCAP , LTRM , USER , RCF1B , REA4	-	-	-	D	+	+	+	+	+	N	N	N
K095		D	R	-	-	-	-	-	-	-	N	N	N
K096		R	R	-	-	-	-	-	-	-	N	N	N
K097[1]	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	Y	N	N
K098	PTRM , PRNM , BCAP , LTRM , USER , RCF1C , RCF2B	R	D	-	+	+	+	+	+	+	N	N	N
K099	MSG	-	-	-	-	-	R	+	-	-	Y	N	N
K101	PTRM , PRNM , BCAP , LTRM , USER , REA1 , DEST	R	D	+	D	+	D	+	+	+	Y	N	N
K102	CIW , CPC , CILC , CPM , CCC , CR0 , CR1 , CR2 , CR3 , CR4 , CDSPL , CAMODE	-	-	-	-	-	R	+	+	-	N	N	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	E x t e n d	B e l l
K103	CR5 , CR6 , CR7 , CR8 , CR9 , CR10 , CR11 , CR12 , CR13 , CR14 , CR15	-	-	-	-	-	R	+	+	-	N	N	N
K104	UTMDEVT , LSES , LPAP , AGUS , OCVST , OTAST , ACTION , NCVST , NTAST , SESSCNTR, VGCNTR , TACNTR , RCVDANNO, TACINDX , LTACINDX	-	-	-	D	+	+	+	+	+	N	N	N
K105	LSES , LPAP , AGUS , SYST	-	-	-	D	+	+	+	+	+	N	N	N
K106	PTRM , PRNM , BCAP , LTRM , USER , DEVC , FIL1A , FIL2A , FIL3 , VTRC , CBRC , OMSG3 , FMTN , CCSN	R	+	-	D	+	+	+	+	+	Y	N	N
K107	TTYP	-	-	-	D	-	R	+	+	+	Y	N	N
K108	PTRM , PRNM , BCAP , LTRM , USER , KRRC , KRLH , KRPR , KRACT	-	-	-	D	+	+	+	+	+	Y	N	N
K109	PTRM , PRNM , BCAP , LTRM , USER , KRRC , KRLH , KRPR , KRACT	R	-	-	D	+	+	+	+	+	Y	N	N
K110	PTRM , PRNM , BCAP , LTRM , USER , KRRC , KRLH , KRPR	R	-	-	D	+	+	+	+	+	Y	N	N
K111	PTRM , PRNM , BCAP , LTRM , USER	R	-	-	+	+	+	+	+	+	Y	N	N
K115	PTRM , PRNM , BCAP , LTRM , SNPT , SNPR , SNLT , CCC , REA1 , ANNO	-	-	-	R	+	+	+	+	+	N	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K116	PTRM , PRNM , BCAP , LTRM , SNPT , SNPR , SNLT , USER , REA1	-	-	-	+	+	+	+	+	+	N	N	N
K117	PTRM , PRNM , BCAP , LTRM , SNPT , SNPR , SNLT , USER , REA1	-	-	-	+	+	+	+	+	+	N	N	N
K119	OSLPAP , USER , TAC , DIA1 , DIA2 , DIA3 , SESSCNTR, VGCNTR , TACNTR , AAIS , AAID	-	-	-	D	+	+	+	+	+	Y	N	N
K120	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	D	+	+	+	+	+	N	N	N
K121	PTRM , PRNM , BCAP , LTRM , USER , PAS1 , PAS2 , PAS3 , NUMDAYS	R	-	-	+	+	+	+	+	-	N	Y	N
K122	PTRM , PRNM , BCAP , LTRM , USER , NUMDAYS	+	R	-	-	-	-	-	-	-	Y	N	N
K123	LTRM , TAC , USER	R	+	-	D	+	+	+	+	+	N	N	N
K124	RCXAPTP , PHAXAPTP	-	-	-	D	-	D	D	+	-	Y	N	N
K125	PTRM , PRNM , BCAP , LTRM , USER	R	+	-	D	+	+	+	+	+	Y	N	N
K126	SATRC	-	-	-	D	+	D	D	+	+	N	N	N
K127	ERCD6	-	-	-	-	-	R	+	-	-	Y	N	N
K128	CON , PRNM , BCAP , LPAP , LSES , REA1 , RCDC , TAC	-	-	-	D	+	+	+	+	+	N	N	N
K129	CMD	-	-	-	-	-	R	+	-	-	Y	N	N
K130	TPRIO , TASK	-	-	-	D	+	+	+	+	+	Y	N	N
K132	OBJ1 , OBJ2	-	-	-	-	-	R	+	-	-	Y	N	N
K133	EXIT	-	-	-	-	-	R	+	-	-	Y	N	N
K134	DEST , NMSG	-	-	-	D	+	+	+	+	+	Y	N	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	E x t e n d	B e l l
K135	PTRM , PRNM , BCAP , LTRM , UPCREAS , UPCSTAT , UPCPROT , UPVENC1 , UPPENC2	-	-	-	D	+	+	+	+	+	N	N	N
K136	FNAM	-	-	-	-	-	D	+	+	+	Y	N	N
K137	FNAM	-	-	-	+	+	D	+	+	+	Y	N	N
K138	FNAM	-	-	-	-	+	D	+	+	+	Y	N	N
K139	FNAM	-	-	-	+	+	D	+	D	+	Y	N	N
K140	PTRM , PRNM , BCAP , LTRM , MXP1 , MXP2	-	-	-	D	+	+	+	+	+	Y	N	N
K141	PTRM , PRNM , BCAP , LTRM , MXP1	-	-	-	D	+	+	+	+	+	Y	N	N
K142	PTRM , PRNM , BCAP , LTRM , MXPT , MXPR , MXLT	-	-	-	D	+	+	+	+	+	Y	N	N
K143	PTRM , PRNM , BCAP , LTRM , STS1 , STS2 , STS3 , STS4	-	-	-	D	+	+	+	+	+	Y	N	N
K144	PTRM , PRNM , BCAP , LTRM , DEVC , FIL1A , FIL2A , FIL3 , VTRC , CBRC , OMSG3 , FMTN , CCSN	-	-	-	D	+	R	+	+	+	Y	N	N
K145	PTRM , PRNM , BCAP , LTRM , USER	R	+	+	+	+	+	+	+	+	N	N	N
K146	BCMOPCD , BCMRTCD , STDHEAD , TASK , BCAP	-	-	-	D	+	+	+	+	+	Y	N	N
K147	PTRM , PRNM , BCAP , LTRM , USER , REA7 , USRTYPE	-	-	-	D	+	D	+	+	+	Y	N	N
K149	DIA5	-	-	-	-	-	R	+	+	-	Y	N	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	E x t e n d	B e l l
K150	PTRM , PRNM , BCAP , LTRM , RSOANNO , RSOOPC , RSOMSG , RSORC , RSO2RC , STDHEAD	-	-	-	D	+	+	+	+	+	Y	N	N
K151	IDEFRC , DMSE , FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K152	COND , MTYPE , OSLPAP , USER , LTAC , AAIS , AAID , SESSCNTR, VGCNTR , TACNTR	-	-	-	D	+	+	+	+	+	Y	N	N
K153	PTRM , PRNM , BCAP , LTRM , USER	-	-	-	-	-	+	+	-	-	N	N	N
K154	PTRM , PRNM , BCAP , LTRM , TCPCL , TCPRC	-	-	-	D	+	D	+	+	+	N	N	N
K155	PTRM , PRNM , BCAP , LTRM , USER , PAS1 , PAS2 , PAS3	R	-	-	+	+	+	+	+	-	N	Y	N
K156	RSLT	-	-	-	D	+	+	R	+	+	N	N	N
K157	PTRM , PRNM , BCAP , ANNO , STDRC , SSORC , STDHEAD , SSINCID , SSOUTCID, SSTYPNAM, SSLENNAM	-	-	-	D	+	+	+	+	+	Y	N	N
K158	PTRM , PRNM , BCAP , LTRM , USER , CPUTEXT , CPUBEGIN, CPUEND , CPUUSED , CPUCLNT , CPUREAS	-	-	-	D	+	+	+	+	+	Y	N	N
K159	USER , ENCPW	-	-	-	+	+	+	+	+	-	Y	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K160	PTRM , PRNM , BCAP , LTRM , USER , TCVG , TAC , TACNTR , RBCAUSER, RCCC , RCDC , TASK , SESSCNTR, VGCNTR , LTHGTRID, GTRID , PID	-	-	-	D	+	+	+	+	+	Y	N	N
K161	TASK , PID	-	-	-	+	+	+	+	+	-	N	N	N
K162	TASK , IOPG , IOMS , PID	-	-	-	D	-	R	D	+	-	Y	N	N
K163	TASK , IOPG , IOMS , PID	-	-	-	D	+	R	D	+	-	Y	N	N
K164	FNAM	-	-	-	+	+	D	+	+	+	N	N	N
K165	FNAM	-	-	-	+	+	+	D	+	+	N	N	N
K166	FNAM , RCHX , SYSFILE	-	-	-	R	+	D	D	+	+	N	N	N
K167		-	-	-	+	+	D	D	+	+	Y	N	N
K169	ACTION , IDX1 , HST1 , STATE , IDX2 , HST2 , IDX3 , HST3 , NNM1 , NNM2 , NNM3	-	-	-	R	-	R	D	+	-	Y	N	N
K170	DTTM , STATE	-	-	-	R	-	R	D	+	-	Y	N	N
K171	HST1 , XCS1 , XCS2	-	-	-	R	-	R	D	+	-	Y	N	N
K174	DIA1	-	-	-	R	-	R	D	+	-	Y	N	N
K175	FNAM	-	-	-	R	-	R	D	+	-	Y	N	N
K176	PRCN , MSG2 , RCHX	-	-	-	R	+	R	D	+	+	Y	N	N
K178	ACTION , STATE , STA2 , PGS1 , PGS2 , SWNR	-	-	-	R	-	R	D	+	-	Y	N	N
K179		-	-	-	R	+	D	D	+	+	Y	N	N
K180		-	-	-	R	+	D	D	+	+	Y	N	N
K181	FNAM	-	-	-	R	+	D	D	+	+	Y	N	N
K182	FNAM	-	-	-	R	+	D	D	+	+	Y	N	N
K183	FNAM	-	-	-	R	+	D	D	+	+	Y	N	N
K184		-	-	-	R	+	D	D	+	+	Y	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K185		-	-	-	R	+	D	D	+	+	Y	N	N
K186		-	-	-	R	+	D	D	+	+	Y	N	N
K187	FNAM	-	-	-	R	+	D	D	+	+	Y	N	N
K188	OBJ1 , DTTM , OBJ2 , DTM2	-	-	-	R	-	R	D	+	-	Y	N	N
K189	PTRM , PRNM , BCAP , LTRM , USER , HST1	R	+	-	D	+	+	+	+	+	Y	N	N
K190	DIA1 , INF1 , INF2 , SUFF	-	-	-	R	-	R	D	+	-	Y	N	N
K191	SUFF , RQM	-	-	-	D	-	D	D	+	-	Y	N	N
K192	NNM1 , HST1 , RSPTC	-	-	-	-	-	D	D	-	-	Y	N	N
K193	PTCID , USER , LPAP , LSES , USTYPPTC	-	-	-	D	-	D	D	+	-	Y	N	N
K194	GBLNBR , ULLNBR	-	-	-	D	-	D	D	+	-	Y	N	N
K195	MSG2	+	+	+	+	+	+	+	+	+	N	N	N
K196	MSG2	+	+	+	+	+	+	+	+	+	N	N	N
K197	MSG2	+	+	+	+	+	+	+	+	+	N	N	N
K198	MSG2	+	+	+	+	+	+	+	+	+	N	N	N
K199	MSG2	+	+	+	+	+	+	+	+	+	N	N	N
K200	MSG2	-	-	-	+	+	+	+	+	+	N	N	N
K201	TSNPID , XASPEC	-	-	-	-	-	R	D	-	-	Y	N	N
K202	TSNPID , INSTNUM , TEXT32 , RMSTAT	-	-	-	-	-	R	D	-	-	Y	N	N
K203	TSNPID , TEXT32 , INSTNUM , RTAANZ	-	-	-	-	-	R	D	-	-	Y	N	N
K204	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y	N	N
K205	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y	N	N
K206	TSNPID , INTTAID , XATXT	-	-	-	-	-	R	D	-	-	Y	N	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	E x t e n d	B e l l
K207	TSNPID , INTTAID , INSTNUM	-	-	-	-	-	R	D	-	-	Y	N	N
K210	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y	N	N
K211	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y	N	N
K212	TSNPID , XATXT , XAFLAG , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K213	TSNPID , XATXT , XAFLAG , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K214	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K215	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K216	TSNPID , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y	N	N
K217	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K218	TSNPID , XATXT , INTTAID	-	-	-	-	-	R	D	-	-	Y	N	N
K220	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y	N	N
K221	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y	N	N
K222	TSNPID , XASPEC , TEXT32	-	-	-	-	-	R	D	-	-	Y	N	N
K223	TSNPID	-	-	-	-	-	R	D	-	-	Y	N	N
K224	TSNPID , XACALL , XATXT , TEXT32 , INSTNUM	-	-	-	-	-	R	D	-	-	Y	N	N
K225	TSNPID , XADBC1 , XADBC2	-	-	-	-	-	R	D	-	-	Y	N	N
K230	TSNPID , TEXT32	-	-	-	-	-	R	D	-	-	Y	N	N
K231	TSNPID	-	-	-	-	-	R	D	-	-	Y	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
K232	TSNPID	-	-	-	-	-	R	D	-	-	Y	N	N
K233	TSNPID , INSTNUM , XACALL , DBCALL , XATXT , LTHGTRID, GTRID	-	-	-	D	+	+	+	+	+	Y	N	N
K234	FNAM , ELEM , EVER , LMSTYPE , PLARC , DMSRC	-	-	-	D	+	D	D	+	+	Y	N	N
K235	TCPCL , PRNM , TCPRC , TCPMS , IPADDR	-	-	-	D	+	D	+	+	+	Y	N	N
K251	IMPVER , FBASUPD , DEFVER	-	-	-	D	+	D	D	+	+	Y	N	N
K252	UPDERR	-	-	-	D	+	D	D	+	+	Y	N	N
K255	DMSE , FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K256	FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K257		-	-	-	D	+	D	D	+	+	Y	N	N
K258	FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K260	DEFVER , FBASUPD	-	-	-	D	+	D	D	+	+	Y	N	N
K261	FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K262	FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K263	FNAM	-	-	-	D	+	D	D	+	+	Y	N	N
K269	OBJ1 , OST1 , BMD1 , OBJ3 , OST2 , BMD2	-	-	-	D	+	D	D	+	+	Y	N	N
K273	TRMA , UPDMODUL	-	-	-	D	+	D	D	+	+	Y	N	N
K274		-	-	-	D	+	D	D	+	+	Y	N	N
K277		-	-	-	D	+	D	D	+	+	Y	N	N
K278		-	-	-	D	+	D	D	+	+	Y	N	N
K279		-	-	-	D	+	D	D	+	+	Y	N	N
K300	UPDPRO , PGPOOL	-	-	-	D	+	D	D	+	+	Y	N	N
K303	UKCOP , UKCRN , UPDTYP , UKCLM	-	-	-	D	+	D	D	+	+	N	N	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	E x t e n d	B e l l
K304	USER , TACTYPE , UKCHSTA	-	-	-	D	+	D	D	+	+	N	N	N
K305	UPDTYP , PGS1 , PGS2 , UKCRN , PGPOOL	-	-	-	D	+	D	D	+	+	Y	N	N
K306	UPDTYP , PGS1 , PGPOOL	-	-	-	D	+	D	D	+	+	Y	N	N
K310	UPDTYP , UKCRN	-	-	-	D	+	D	D	+	+	N	N	N
K311	UPDTYP , UKCRN	-	-	-	D	+	D	D	+	+	N	N	N
K314		-	-	-	D	+	D	D	+	+	N	N	N
K317	UKCOP , UKCRN , UPDTYP , UKCLM , RCCC , RCDC , LTRM , USER	-	-	-	D	+	D	D	+	+	N	N	N
K318	UPDTYP , UKCRN , LTRM , USER	-	-	-	D	+	D	D	+	+	N	N	N
K320	USER , TACTYPE , UERCODE , UERINFO , RCDC	-	-	-	D	+	D	D	+	+	N	N	N
P001	XPFUNC , ACPNT , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	N
P002	XPFUNC , ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	N
P003	ACPNT , XPRJCT , XPLTH	-	-	-	D	+	+	+	+	+	Y	N	N
P004	ACPNT , OSLPAP , XPRJCT	-	-	-	D	+	+	+	+	+	Y	N	N

MSG-Id	Inserts	Station	SYSLINE	Partner	SYSLOG	MSGTAC	SYSOUT	SYSLST	Console	USERDEST	Compress	Extend	Bell
P005	ACPNT , XPNSEL , XPTSEL , XPLSSEL , XPCSSEL , XPHSSEL , XPLPSEL , XPCPSEL , XPHPSEL	-	-	-	D	+	+	+	+	+	Y	N	N
P006	ACPNT , OSLPAP , XP0OBID , XP1OBID , XP2OBID , XP3OBID , XP4OBID , XP5OBID , XP6OBID , XP7OBID , XP8OBID , XP9OBID	-	-	-	D	+	+	+	+	+	Y	N	N
P007	ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	N
P008	ACPNT , OSLPAP , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P009	ACPNT , OSLPAP , XPRJCT , XPLTH , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P010	ACPNT , OSLPAP , XPNSEL , XPTSEL , XPLSSEL , XPCSSEL , XPHSSEL , XPLPSEL , XPCPSEL , XPHPSEL , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P011	ACPNT , OSLPAP , XP0OBID , XP1OBID , XP2OBID , XP3OBID , XP4OBID , XP5OBID , XP6OBID , XP7OBID , XP8OBID , XP9OBID , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P012	XPCTYPE , XPCCLS , XPCVAL , XPBCAM , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	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	E x t e n d	B e l l
P013	ACPNT , OSLPAP , XPCRES , XPSRC , XPNDIA , XP1BOOL , XP2BOOL , XP3BOOL , XP4BOOL , XP5BOOL , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P014	XPFUNC , ACPNT , OSLPAP , XPRET , XPERR , XP1INFO , XP2INFO , XPOSAS , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	N
P015	XPFUNC , ACPNT , OSLPAP , XPLNK , XPSRC , XPNDIA , XPINI , XP1INFO , XP2INFO , XPOSAS , XPCORR	-	-	-	D	+	+	+	+	+	Y	N	N
P016	ACPNT , OSLPAP , XPLNK , XPNDIA , XPOSAS	-	-	-	D	+	+	+	+	+	Y	N	N
P017	XPPDU , XP1DIA , XP2DIA , XP3DIA	-	-	-	+	+	+	+	+	+	Y	N	N
P018	ACPNT , OSLPAP , XPPTYP , XPFSMN	-	-	-	D	+	+	+	+	+	Y	N	N
P019	ACPNT , OSLPAP , XPAPDU , XP3INFO	-	-	-	D	+	+	+	+	+	Y	N	N
P020	XPTRFAIL	-	-	-	D	+	+	+	+	+	Y	N	N
P021	XPEVT , ACPNT , OSLPAP , XPOSAS , XPASST	-	-	-	D	+	+	+	+	+	Y	N	N

[1] If you are working with a sign-on service in a UTM application, then these messages are not generated, i.e. they are also not output to the message destinations MSGTAC or SYSLOG.

6 UTM return codes

6.1 KDCS return codes in KCRCCC

The following categories must be distinguished here:

Code	Category	Actions
000	No errors	Operation carried out without errors.
01Z - 09Z	Remarks	Operation carried out.
10Z - 19Z	Warning or minor error	Once suitable actions have been taken, the program unit run can be continued. Operation not carried out.
20Z - 39Z	Special functions	A KDCS special function is signalled. Otherwise, as warnings.
40Z - 69Z	Error	Operation not carried out. It is not normally practical to continue the program run. Communication with the terminal is still possible.
70Z - 99Z	Serious error	It is not possible to continue the program run. openUTM resets the transaction and aborts the service. openUTM generates a system-specific error message and issues it during an inter-active service on the terminal.

Please note the following points:

– For a more detailed explanation of the KCRCCC return codes see the openUTM manual „Programming Applications with KDCS" for each individual KDCS call.

– If two or more errors occur simultaneously, the errors in the highest category are displayed. If an operation was not carried out (return code > 09Z), the contents in <parm2> are not changed.

– The evaluation of a UTM dump is described in chapter "The UTM dump" on page 53.

The following return codes are defined (field name specifications: COBOL name/C name):

Code	Meaning
000	Operation carried out successfully
01Z	Length conflict in KCLA/kcla or KCLKBPRG/kclcapa
02Z	Length conflict in KCLPAB/kclspa
03Z	Name in KCRN/kcrn invalid
04Z	Name in KCRN changed
05Z	Input formatting not carried out with the format characters specified in KCMF/kcfn. Line mode: 1st KCMF/kcfn character blank
06Z	Time entry changes within message (DPUT call)
07Z	Length conflict regarding KCLI/kcli
08Z	When reading with waiting in the DGET call: There is currently no message.
09Z	For INFO CD: The Kerberos dialog has returned an error or the Kerberos information was returned in truncated form,
10Z	Message already read in full
11Z	When reading without waiting in the DGET call: There is no message.
12Z	There are no more messages from specified service ID or there is no service stack with the specified number.
14Z	Name in KCRN/kcrn not found
16Z	Operation illegal and was not carried out
19Z	Function key or special function not generated
20Z to 39Z	KDCS special function (short messages)
40Z	System cannot perform operation (generation error, system error, deadlock, long lasting locks)
41Z	Operation illegal at this point
42Z	Operation modification illegal
43Z	Length entry in KCLM/kclm, KCLI/kcli or KCLA/kcla or KCWTIME/kcwtime invalid
44Z	Name in KCRN/kcrn invalid
45Z	Message format ID KCMF/kcfn or creation time (DGET) invalid
46Z	Name in KCLT/kclt or KCPA/kcpa or KCLANGID/kclangid, KCTERRID/kcterrid, KCCSNAME/kccsname or KCQMODE/kcqmode is invalid
47Z	No memory area <parm2>, invalid area address, or area cannot be read/written at the length specified
48Z	Invalid interface version
49Z	Unused parameters are not binary zero.

Code	Meaning
51Z	Sequence not maintained in DPUT call
52Z	Message destination is not permitted in KCRN (DPUT call)
53Z	Value in KCDPID/kcdpid or KCGTM/kcgtm is invalid (DGET call).
54Z	Value in KCNORPLY/kcnoreply is invalid (in the case of CTRL PR or PE: Value not equal Y or binary zero).
55Z	Name in KCPI/kcpi is invalid (APRO call)
56Z	Value of KCMOD or time entry in KCTAG/kcday,...,KCSEK/kccec is invalid (DADM, DPUT call)
57Z	Value in KCPOS/kcpos is invalid (MCOM call)
58Z	Value in KCOF/kcof (APRO call)is invalid or the value in KCNEG/kcneg is invaliid (MCOM call).
70Z	System cannot perform operation (generation error, system error)
71Z	Operation illegal at this point, e.g. still no INIT
72Z	Operation modification illegal
73Z	Length entry in KCLA/kcla, KCLM/kclm or KCLI/kcli invalid
74Z	Name in KCRN/kcrn invalid
75Z	Message format ID KCMF/kcfn invalid
77Z	No memory area <parm2>, invalid area address, or area cannot be read/written at the length specified
78Z	FORMAT exit reports error (MGET call)
79Z	Operation code cannot be interpreted *
80Z	Generation error prior to program start
81Z	PEND destination in conflict with MPUT destination
82Z	PEND variant in conflict with the operands KCOM/kcom or KCRN/kcrn in the MPUT call
83Z	No MPUT specified before PEND KP,RE,FI,ER,FR in an interactive program or no MPUT was specified before PEND KP,RE in an asynchronous program or an MPUT was issued prior to a PGWT PR.
84Z	PEND missing
86Z	A complex message was not completed or no FPUT/DPUT for the addressed job receiving service was given after an APRO AM.
87Z	Operation modification is illegal
88Z	Invalid interface version
89Z	Unused parameters are not equal to binary zero.

* The return code 79Z (operation code cannot be interpreted) is not to be assigned to any special operation.

6.2 Internal return code KCRCDC

The incompatible KDCS return code KCRCDC consists of four printable characters and contains more detailed information on the cause of the error than the compatible return code KCRCCC.

The incompatible return code can be set in the following situations:

– if the KDCS call returns the compatible return code 40Z
– if openUTM terminates a service abnormally with PEND ER (KCRCCC $\geq$ 70Z)
– if a transaction is implicitly reset by openUTM.

The return code KCRCDC has the following structure:

sm ##

The first character (s) specifies the part of openUTM which reported the error:

A   Administration
F   Formatting component
K   UTM system code or ROOT code.
U   UPIC (server part)

For all return codes which start with the letter K, the second letter (m) indicates the UTM module which reported the error (see the list below).

3   KCSPEND (PEND and RSET operations)
6   KCSMPUT (MPUT operation)
7   KCSFPUT(PEND and RSET operations)
8   KCSSSB (SGET, SPUT, SREL, PTDA, GTDA and UNLK operations)
9   KCSLPUT (LPUT operation)
A   KCSWAIT
B   KCSDGET (DGET operation)
C   KCDCTRL (CTRL operation)
D   KCDAPRO (UTM-D operations APRO, MPUT, MGET, ...)
E   KCSPADM (PADM operation)
F   KCSINFO (INFO operation)
G   KCSDADM (DADM operation)
H   KCSSIGN (SIGN operation)
I   KCSISLP (internal operations)
K   General DC code
L   KCSSTRT
M   KCSTRHD (internal operations)
N   KCSDLO
Q   KCSCRO (QCRE operation)
R   KDCRTMM (ROOT code)
S   KCDSEFU (UTM-D service functions)

T KCDOSTM (OSI TP transaction termination)
U KDCRTDB (handling of DB processes)
V KCSTUTM (MGET, MPUT operation for UPIC and socket clients)
W KCDOSSF (special OSI TP functions)
Y KCDSYPM (LU6.1/OSI TP transaction termination)

The following table describes the error causes and, where applicable, the required recovery action.
If TESTMODE=OFF applies, the PENDER dump is suppressed in cases marked with ([1]).
At the same time, loading is suppressed unless PGWT calls are permitted for the current TAC and the KDCS call in which the error occurred was not a PEND call.

Code	Module	Cause of error and action (if any)
A010	KCSSADM	User is not a SAT administrator.
A011	KCSSADM	TAC has no SAT admission.
A012	KCSSADM	SAT subsystem is not available.
A013	KCSSADM	SAT version is not compatible.
A015	KCSSADM	An inverse KDCDEF is running or is to be started.
A016	KCSSADM	The gobal cluster administration is locked.
A100	KCSADMI	When calling the program interface for administration, the address specified for the parameter area is either inaccessible or cannot be accessed for the length of the parameter area or it is not on a word boundary. Response: Checking the address specified for the parameter area in the program unit.
A101	KCSADMI	An invalid return code was issued internally when calling the programming interface for administration. System error. Action: Contact System Support. The PEND ER dump is required for diagnosis.
ABTR	KCSENDE	The code is reported in UTM messages K017 or K055. It occurs whenever a service is forced to terminate by UTM and UTM-STXIT handling is not run (STXIT switched off or STXIT code overwritten). This error is comparable to an ABNORMAL TASK TERMINATION, and should therefore be accurately diagnosed. Generally, this does not adversely affect application execution.
FH01	KDCRTST	Error occurred during formatting. See additional return code from FHSCON or FHS in KCRMF field. [1]
FH02	KDCRTST	Warning: hardware function not available. [1]
FH04	KDCRTST	Serious error when initializing FHS Format Handling System UTM aborts start of application.
FR00	KDCRTST	Error occurred during FORMAT-Exit-Function "CHECK_FOR_DOWNLOAD"

Code	Module	Cause of error and action (if any)
FR01	KDCRTST	– A format ID with "-" was specified although the application was generated without FORMAT exit, or – a "*" or "+" format was specified but the generation does not contain a FORMSYS statement. [1]
FR02	KDCRTST	KDCFOR not allowed for partial format. [1]
FR98	KDCRTFH	'-' Format, i.e. user-own formatting; the length of the input message was illegally set by the FORMAT EXIT (< 0). The return code is set by KDCRTFH.
FR99	KDCRTFH	'-' Format, i.e. user-own formatting; the length of the restart area was illegally changed by FORMAT EXIT. The return code is set by KDCRTFH.
FRxx	KDCRTST	Format in '-' Format, i.e. user-own formatting: xx is the return code entered in the ....MCRF field in the user-own control area MDCBL. The entry comes from the FORMAT exit. The values X'01' , x'02' , X'03' , X'04' , X'08' , X'10', X'98' and X'99' are not permitted for the format exit, because these return codes are reserved for UTM and FHS.
K300	KCSPEND	Incorrect operation modification in the PEND call.
K301	KCSWAIT	(As for KA00) The buffer is too small for the input message. Action: Define larger buffer length with MAX TRMSGLTH=length in KDCDEF generation. No PEND ER dump created.
K302	KCSPEND	The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
K303	KCSPEND	An incorrect operation modification was used in the PGWT call.
K304	KCSPEND	The DB system requests a CLOSE DB call prior to END RE/FI, but the call was not entered.
K305	KCSPEND	A PEND KP or a PEND PA/PR with task switch was called, although the DB system with which the transaction was opened does not permit a PEND KP.
K306	KCSPEND	The transaction and the service had to be reset since the DB system reset the DB transaction when closing down.
K307	KCSPEND	PEND PS is only permitted in the first part of the sign-on process.
K308	KCSPEND	The DB transaction was closed prior to PEND KP or PEND PA/PR with task switch.
K309	KCSPEND	The TAC specified in KCRN is not permitted as a follow-on TAC.
K310	KCSPEND	Error when storing the MPUT message during PEND PA/PR with task switch.
K311	KCSPEND	A temporary end of transaction (PTC) was reached with PEND RE/FI, but the DB system where the transaction is open does not permit the transaction status.

Code	Module	Cause of error and action (if any)
K312	KCSPEND	A formatted message cannot be output because it is too long. Action: create a simpler format.
K313	KCSPEND	A SIGN OF/OB was issued in a program unit run, but the output message is for the job-receiver or the program run was terminated with an PEND variant that is not permitted.
K314	KCSPEND	A program unit run of a sign-on service was not terminated with PEND PS following a successful SIGN ON.
K315	KCSPEND	With service stacking, an MPUT PM was issued with KCLM/kclm > 0 prior to PEND FI. The last output from the preceding service was a LINEMODE message and can only be output in unmodified form.
K316	KCSPEND	PEND RS was called in the program unit or a situation (e.g. loss of connection) has occurred which requires UTM to roll back the transaction. However, PEND RS is not permitted because no service restart is possible. This is the case if: – the service was started by a UPIC client or an OSI TP job submitter which has not selected the Functional Unit Commit and no user with the restart attribute is signed on at the connection/association and no local service restart is possible because the last synchronization point was not set with PEND SP or PEND FC (UPIC only) – or the last synchronization point was set with PGWT CM. No PENDER dump is written.
K317	KCSPEND	The actual size of the service memory (SQL) to be saved is greater than the permitted maximum value limited by VGMSIZE.
K318	KCSPEND	PGWT was called in a program unit for whose TAC – no TAC class is generated or – if TAC classes have been generated: PGWT=YES is not generated for the TAC class – if TAC-PRIORITIES has been generated: the TAC was not generated with PGWT=YES Action: Correct generation with KDCDEF.
K319	KCSPEND	There are not enough tasks to use the PGWT or PGWT was called from a system task (only possible via a privieged LTERM). Action: Increase number of tasks.
K320	KCSPEND	After a RSET in a distributed transaction, the transaction was not reset, although UTMD RSET = GLOBAL was generated.
K321	KCSPEND	The edit profile name of the saved message of the stacked service is not generated.
K322	KCSPEND	The operation modification FC is not permitted in asynchronous services or server services.
K323	KCSPEND	The operation modification FC is not permitted in the sign-on procedure if a service restart is required.
K324	KCSPEND	KCRN does not contain blanks on PEND RS or PEND FR.

Code	Module	Cause of error and action (if any)
K325	KCSPEND	The operation modification RS is not permitted in the sign-on procedure or in the MSGTAC program, or the Resource Manager of the XA connection requires a reset of the transaction in the case of an xa_end of the sign-on procedure or the MSGTAC program,.
K326	KCSPEND	Switch between dialog transaction code and asynchronous transaction code is not permitted, or the follow-up TAC in PEND FC is not a service TAC, or the follow-up TAC in PEND PA/PR, PS, KP, RE or SP is not a follow-up TAC.
K327	KCSPEND	The operation modification FC is not allowed in the sign-on service if the last sign-on attempt failed.
K328	KCSPEND	The sign-on service cannot be terminated normally when the validity period of the password has expired but the password has not been changed.
K329	KCSPEND	The sign-on service cannot be terminated normally when the password passed with KDCUPD does not meet the complexity level requirement or is too short and the password has not been changed with SIGN CP.
K330	KCSPEND	A SIGN OB in the sign-on service for terminals is only allowed in conjunction with an MPUT NT/NE.
K331	KCSPEND	If a sign-on service with a service restart is terminated without the user signing off, then no MPUT NT/NE is allowed.
K332	KCSPEND	If the sign-on service with a service restart PEND FI is terminated without a preceding MPUT, then openUTM terminates the open service.
K333	KCSPEND	There should be a wait for a DGET message, but the follow-up program unit is not in a TAC class. Action: Generate follow-up TAC with TAC class.
K334	KCSPEND	There should be a wait for a DGET message; only PEND PA/PR/RS/ER/FR and PGWT RB are permitted.
K335	KCSPEND	The process is terminated abnormally as during a timeout on the PGWT bourse it was determined that not enough processes are active to continue the procedure. No PENDER dump is written.
K336	KCSPEND	Although required, no MPUT call was made prior to PEND KP/RE/FI/FR/ER or PGWT KP.
K337	KCSPEND	An MPUT call was made prior to PGWT PR.
K338	KCSPEND	The transaction was reset because the application is being terminated.
K339	KCSPEND	The MSGTAC program unit was terminated without reading a message with FGET.
K340	KCSPEND	On transaction rollback, the database reported an error which makes it necessary to dynamically load the application program.
K341	KCSPEND	Only in cluster applications: A user was generated or deleted in the current transaction. The user file could not be locked because it is currently being accessed by a KDCDEF run.

Code	Module	Cause of error and action (if any)
K342	KCSPEND	Only in UTM cluster applications: A user was generated or deleted in the current transaction. The cluster user file could not be locked because another process belonging to the same node application or another node application has locked the cluster user file.
K343	KCSPEND	PET state rejected because the service is locked be administration.
K344	KCSPEND	CR could not be written because a global ADM lock is set.
K345	KCSPEND	The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
K346	KCSPEND	The sign-on service is not permitted to terminate an open service if a transaction in the open service is in the "prepare to commit" (PTC) state. The sign-on service is terminated abnormally.
K347	KCSPEND	In a UTM cluster application, a transaction wants to go to the status "prepare to commit" (PTC) but an error occurred while writing the status to the cluster user file.
K348	KCSPEND	The cluster page pool in the KDCFILE is full. Action: Generation with KDCDEF; increase size of cluster page pool with CLUSTER PGPOOL=(number,...).
K349	KCSPEND	The connection to the partner has been cleared down.
K350	KCSPEND	Invalid operation modification after the abnormal termination with CTRL AB of a dialog in which the commit functional unit was selected.
K351	KCSPEND	The transaction is marked to be rolled back but a PEND/PGWT call was used to set a transaction forward.
K360	KCSPEND	The transaction was rolled back with PGWT RB because KCSPEND was called from KDCROOT with PGWT RB.
K361	KCSPEND	The transaction was reset with PEND RS because KCSPEND was called by KDCROOT with PEND RS.
K362	KCSPEND	The transaction was reset with PEND ER because KCSPEND was called by KDCROOT with PEND ER.
K363	KCSPEND	The transaction was reset with RSET because KCSPEND was called by KDCROOT with RSET.
K601	KCSMPUT	Buffer for dialog messages generated too small. Action: Modify generation, define larger buffer with MAX NB=length. [1]
K602	KCSMPUT	MPUT call demands format output but formatting was not generated. [1]
K603	KCSMPUT	MPUT with screen function KCCARD, but terminal does not have an ID card reader. Action: Change generation or program unit [1]

Code	Module	Cause of error and action (if any)
K604	KCSMPUT	MPUT with screen function KCCARD and user ID has operand CARD=(position,string): the ID card reader cannot be used for KDCSIGN checks and data input at the same time. Action: Change generation or program unit
K605	KCSMPUT	MPUT with screen function KCCARD and output in formatted mode. Action: Change program
K606	KCSMPUT	Caused by a MPUT call with KCDF not equal to binary zero and one of the following conditions: - follow-up partial message in formatted mode - KCMF = name of an EUA format - KCMF = name of an edit profile. Action: Change program unit
K607	KCSMPUT	The MPUT message specified by address and length partly covers an internal ROOT buffer area (the MPUT buffer) Action: Change program (length of MPUT too large)
K608	KCSPEND	(See K345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
K609	KCSMPUT	With a follow-up message segment, the edit profile name is different to that entered for the first message segment. Action: Change program unit.
K610	KCSMPUT	The ES operation modification of the MPUT call is only permitted for UPIC and SOCKET partners.
K611	KCSMPUT	In the case of the MPUT ES call, the KCRN field is not filled with blanks.
K612	KCSMPUT	MPUT was called in the MSGTAC program unit.
K613	KCSMPUT	Operation modification illegal.
K614	KCSMPUT	MPUT PM in the asynchronous service.
K615	KCSMPUT	MPUT PM in the first part of the sign-on service.
K616	KCSMPUT	MPUT PM with KCLM/kclm not equal to 0 in the sign-on service.
K617	KCSMPUT	MPUT PM, but the service is neither inserted nor a sign-on service.
K618	KCSMPUT	MPUT PM in the sign-on service for a UPIC connection user.
K701	KCSFPUT	UTM refuses an asynchronous message because level 2 has already been reached in the page pool. Action: See K345
K702	KCSFPUT	The name of a dialog TAC was specified in KCRN. Response: new generation or change the program.
K703	KCSMPUT	(See K602) The MPUT call demands format output but formatting was not generated.

Code	Module	Cause of error and action (if any)
K704	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
K705	KCSFPUT	An asynchronous message is to be sent with FPUT to an LTERM or (OSI-)LPAP partner, for which LTERM ...,QAMSG=N is specified in the KDCDEF generation. UTM therefore rejects the message. Action: Set up connection
K706	KCSFPUT	An asynchronous message is to be sent to an dialog terminal with FPUT/DPUT in line mode. The message is longer than the buffer area defined in the MAX NB= parameter in the generation. UTM does not accept the message. Action: Specify a higher value for MAX NB= or short message segments.
K707	KCSFPUT	The target time specified in a DPUT call exceeds the limits set by the generation (MAX DPUTLIMIT1=, DPUTLIMIT2=). Action: Change time entry in DPUT call or in generation.
K708	KCSFPUT	An asynchronous message is to be sent to an LTERM or (OSI-)LPAP partner, an asynchronous TAC, a TAC queue, a user queue, a temporary queue or an (OSI) LPAP with FPUT/DPUT. The number of asynchronous messages for this LTERM partner or TAC has already reached the maximum value defined at generation (KDCDEF: LTERM QLEV=, TAC QLEV=, USER QLEV=, QUEUE QLEV=, LPAP QLEV=, OSI-LPAP QLEV=).
K709	KCSFPUT	With a follow-up message segment, the edit profile name is different to that entered for the first message segment. Action: Change program unit.
K710	KCSFPUT	Caused by FPUT/DPUT call with KCDF not equal to 0 and one of the following conditions: - KCMF/kcfn = name of an EUA format - KCMF/kcfn = name of an edit profile. Action: Change program unit.
K711	KCSFPUT	A DPUT was issued with a new destination in KCRN although the preceding DPUT to another destination is not yet completed.
K712	KCSFPUT	The name of a UTM-D partner, a MUX connection or a UPIC partner was specified in KCRN. Response: Correct generation or program.
K713	KCSFPUT	Caused by an FPUT/DPUT call to an ID card reader Action: Change program unit.
K714	KCSFPUT	The name of a message complex was specified in KCRN in an FPUT call. Action: Correct program.

Code	Module	Cause of error and action (if any)
K715	KCSFPUT	The name of a message complex was specified in KCRN, but this name does not match the message complex currently open. Action: Correct program.
K716	KCSFPUT	The name of a message complex was specified in KCRN, but no message complex is currently open. Action: Correct program.
K717	KCSFPUT	The name of a TAC was specified in KCRN, but the specification is incompatible with the specification in KCOM. Action: Correct program.
K718	KCSFPUT	An asynchronous service running under a deleted LTERM partner attempted to issue an FPUT or DPUT.
K719	KCSFPUT	An asynchronous service running under a deleted user attempted to issue an FPUT or DPUT.
K720	KCSFPUT	In the case of a DPUT call to a USER queue: There is no USER with the name specified in KCRN, or the USER was deleted.
K721	KCSFPUT	In the case of a DPUT call to a USER queue: The KSET of the user and the write ACL of the USER queue do not have a common key.
K722	KCSFPUT	Time-controlled DPUT calls for USER queues are not possible (KCMOD not equal to blank).
K723	KCSFPUT	In the case of a DPUT call to a QUEUE object: There is no QUEUE object with the name specified in KCRN.
K724	KCSFPUT	In the case of a DPUT call to a USER queue: The KSET of the LTERM and the write ACL of the USER queue do not have a common key.
K725	KCSFPUT	Time-controlled DPUT calls are not possible for QUEUE objects. (KCMOD not equal to blank)
K726	KCSFPUT	Invalid value in KCQTYP
K727	KCSFPUT	The current RSO version does not support this function.
K728	KCSFPUT	The message target is not a RSO printer.
K729	KCSFPUT	The message target is not a LTERM.
K730	KCSFPUT	A message for which a message-part has already been written may not be extended with an FPUT RP or DPUT RP.
K731	KCSFPUT	No asynchronous messages can be sent to a TAC generated with CALL=NEXT.
K732	KCSFPUT	No asynchronous messages may be sent to KDCMSGLT.
K733	KCSFPUT	No asynchronous message may be sent to an LTERM that has been implicitly generated for the internal cluster communication.

Code	Module	Cause of error and action (if any)
K801	KCSPEND	(See 345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
K802	KCSPEND	(See 302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
K804	KCSSSB	No GSSBs were generated or more GSSBs were generated than permitted in the generation.
K805	KCSSSB	SPUT created more LSSBs than specified at generation.
K810	KCSSSB	When accessing a GSSB, TLS or ULS: The area cannot be occupied by the transaction following a certain wait period. Action: Increase wait period by increasing RESWAIT=(time1,...) parameter in KDCDEF generation.
K811	KCSSSB	When accessing a GSSB, TLS or ULS: The area is currently locked by another transaction for an "indeterminate" period, i.e. it has occupied the area and then issued PEND KP or PGWT KP call.
K812	KCSSSB	When accessing a ULS: The user ID for the user whose ULS is to be accessed will be deleted.
K813	KCSSSB	The application is terminated.
K820	KCSSSB	Waiting for a global scondary storage area would lead to a deadlock.
K822	KCSPEND	(See K348) The cluster page pool in the KDCFILE is full. Action: Generation with KDCDEF; increase size of cluster page pool with CLUSTER PGPOOL=(number,...).
K823	KCSSSB	Only in cluster applications: A timeout occurred on a request for a file lock for the administration file for GSSB or ULS.
K824	KCSSSB	When accessing a GSSB, TLS or ULS: The area is currently locked and the task cannot wait for the lock to be released because there are already too many tasks waiting. Action: – Start more tasks. – Limit the number of tasks for the TACs that access GSSB or ULS and then start more tasks than the specified limit value. For information on how to distribute UTM services that use GSSB or ULS memory areas to the tasks in a UTM cluster application, see openUTM manual "Using openUTM Applications under BS2000 Systems".

Code	Module	Cause of error and action (if any)
K825	KCSSSB	Only in cluster applications when accessing a GSSB or ULS: The queue at the requested area has already reached the maximum length; the service cannot currently wait for this area.
K826	KCSSSB	Only in cluster applications when accessing a GSSB or ULS: A transient error occurred on a request for an internal resource; access to the requested area is not currently possible.
K827	KCSSSB	Only in cluster applications when accessing a GSSB or ULS: The area is currently locked by a failed node; there is at present no point in waiting for this area.
K901	KCSLPUT	UTM rejects an LPUT call because level 2 has already been reached in the page pool . Action: Change program or increase page pool in generation.
K902	KCSPEND	(See 302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
K903	KCSLPUT	UTM rejects an LPUT call because a DMS error occurred during the last write procedure to the user log file(s). The DMS error was output with message K043. The records buffered in the page pool remain. Action: Analyze the K043 message and the DMS error code, restore or recreate the user log file(s) and then continue with the KDCLOG administration command (or via the corresponding program interface). The LPUT records in the page pool are then output. The lock for the LPUT call is removed.
KA00	KCSWAIT	The buffer is too small for the input message. Action: Define larger buffer length with MAX TRMSGLTH=length in KDCDEF generation. No PEND ER dump created.
KA01	KCSWAIT	A serious error ocuured during communication between job submitter and job receiver (UTM-D). Communication cannot be continued; UTM aborts the job-receiving service with PEND ER. Possible cause: - PEND ER by the job submitter - timeout on the connection to the job submitter. Diagnostic aids: – analyze VTV error message K086 – for PEND ER in job-submitting service, look for cause in dump.
KA02	KCSWAIT	The page pool in the KDCFILE is full, consequently the messages could not be passed to the task waiting in the PGWT. Action: Generation in KDCDEF: increase page pool using MAX PGPOOL=(number,...)

Code	Module	Cause of error and action (if any)
KA03	KCSWAIT	UTM has received an unencrypted message although an encrypted message was expected.
KA04	KCSWAIT	The connection to the partner was shut down.
KA05	KCSWAIT	A transaction in PET state was reset following an administration request.
KB01	KCSDGET	DGET calls are not permitted in the first part of the sign-on service.
KB02	KCSDGET	A DGET message must be waited for; no further DGET calls are therefore permitted.
KB03	KCSDGET	Invalid value of KCQTYP
KB04	KCSDGET	There is no object for the name of the type KCQTYP specified in KCRN, or this object has been deleted.
KB05	KCSDGET	Messages to dialog or asynchronous TACs cannot be read by means of a DGET call.
KB06	KCSDGET	The read ACL or the TAC or USER queue and the KSET of the user's LTERM do not have a common key.
KB07	KCSDGET	The read ACL or the TAC or USER queue and the KSET of the user do not have a common key.
KB08	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KB10	KCSDGET	Negative wait time in the case of a DGET FT /BF call, or wait time is not equal to zero in the case of a DGET NT/BN/PF/PN call.
KB11	KCSDGET	DGET calls with waiting are not permitted for the MSGTAC program.
KB12	KCSDGET	In the case of the call DGET NT/BN/PN, the name or type of the specified queue does not suit the previous DGET call of the current program unit run.
KB13	KCSDGET	An attempt was made to read from a USER queue although the application was generated without any users.
KB14	KCSDGET	An attempt was made to read from a TAC queue with STATUS = HALT or KEEP.
KB15	KCSDGET	An attempt was made to read from a TAC queue with ADMIN=Y, but the user is not an administrator.
KB16	KCSDGET	Operation modifier in KCOM is invalid.
KB17	KCSDGET	For DGET BF/PF: Value in KCDPID does not contain a valid DPUT ID.
KB18	KCSDGET	For DGET FT/NT: KCMF/kcfn does not contain blanks.
KB19	KCSDGET	For DGET NT/BN/PN (next): KCOM does not match the preceding DGET call, or no DGET FT/BF/PF (first) has been issued yet in this program run, or a PGWT call was made in the intervening period.

Code	Module	Cause of error and action (if any)
KB20	KCSDGET	For DGET NT/BN/PN: Since the last DGET call the DGET queue deleted and regenerated.
KB21	KCSDGET	For DGET BN/PN (next): Since the last DGET call the USER specified in the KCRN and KCQTYP or the temporary queue was deleted and regenerated.
KB22	KCSDGET	For DGET BN: There is no message with the creation time specified in KCGTM, or processing has taken place in the meantime.
KB23	KCSDGET	For DGET PF: There is no message with the creation time specified in KCGTM, or processing has taken place in the meantime.
KB24	KCSDGET	For DGET BF/PF: KCDPID does not match the specifications in KCRN and KCQTYP.
KB25	KCSDGET	DGET calls with waiting are not permitted in the sign-on service.
KB26	KCSDGET	Attempts were made with DGET FT/NT/PF/PN to read from the dead letter queue KDCDLETQ.
KC01	KCDCTRL	VTV has not been generated.
KC02	KCDCTRL	The first character in the service ID specified in KCRN is not '>'.
KC03	KCDCTRL	The call was issued for an asynchronous service, i.e. the service ID specified in KCRN was defined with an APRO AM call.
KC04	KCDCTRL	The VGID specified in KCRN is invalid.
KC06	KCDCTRL	The CTRL call was addressed to a partner which is not communicated with via the OSI-TP protocol
KC07	KCDCTRL	CTRL PR was intended for a partner for which the commit functional unit has not been selected.
KC08	KCDCTRL	CTRL PE was intended for a partner for which the commit functional unit has not been selected.
KC09	KCDCTRL	CTRL AB was intended for a partner to whom a message has already been sent with MPUT.
KC10	KCDCTRL	The operation modification OM is nether PR nor PE nor AB.
KC11	KCDCTRL	KCLA/kcla is not zero.
KC12	KCDCTRL	KCLM/kclm is not zero.
KC13	KCDCTRL	KCMF/kcfn is not filled with spaced (blanks).
KC14	KCDCTRL	KCDF/kcdf does not contain binary zeros.
KC15	KCDCTRL	The extended parameter area (EXTENT) does not contain binary zeros.
KC16	KCDCTRL	The call was intended for a partner for whom an MPUT HM has already been issued.
KC17	KCDCTRL	The call CTRL PR was intended for a partner in a middle node, but the local service had not already received a Prepare from its job submitter.

Code	Module	Cause of error and action (if any)
KC18	KCDCTRL	The call CTRL PE was intended for a partner in a middle node, but the local service had not already received a Prepare from its job submitter.
KC19	KCDCTRL	The CTRL AB call was intended for a partner for which the commit functional unit has not been selected and the service status is not O.
KC20	KCDCTRL	The CTRL AB call was intended for a partner for which the commit functional unit has not been selected and the service status is neither O nor C.
KC21	KCDCTRL	The CTRL PR call is not permitted because the service status is not O.
KC22	KCDCTRL	The CTRL PE call is not permitted because the service status is not O.
KC23	KCDCTRL	The CTRL PR call is not permitted because the transaction status is P.
KC24	KCDCTRL	The CTRL PE call is not permitted because the transaction status is P.
KC25	KCDCTRL	Binary zeros have not been entered for the unused part of the extended parameter area (EXTENT).
KC26	KCDCTRL	The KCNORPLY field does not have either the value Y or binary zero.
KD00	KCDAPRO	KCRN does not contain a valid service ID (MGET, MPUT, FPUT, DPUT).
KD01	KCDAPRO	LTAC is not defined.
KD02	KCDAPRO	LTAC cannot be used; LTAC is locked.
KD03	KCDAPRO	User does not have a key in his/her key set for the lock on the LTAC.
KD04	KCDAPRO	APRO DM was called, but the LTAC is generated as an asynchronous LTAC, or APRO AM was called and LTAC is generated as a dialog LTAC.
KD05	KCDAPRO	APRO created more job-receiving services than specified at generation time (KDCDEF generation: UTMD MAXJR=)
KD06	KCDAPRO	MPUT to job-receiving service and KCDF not binary 0
KD08	KCSPEND	(See K345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
KD09	KCDAPRO	APRO DM was called, but the virtual connections to the remote application were cleared down with the administration command KDCLPAP ACT=QUIET.
KD10	KCDAPRO	APRO DM was called, but no virtual connection was set up to the remote application.
KD11	KCDAPRO	APRO DM was called, but the remote application was generated as "contention winner" and the wait period for session occupancy is 0 (only for LU6.1).
KD12	KCDAPRO	Following an MPUT NE/HM to a job-receiving service, another MPUT was issued to the job-receiving service.
KD13	KCDAPRO	No connection exists to the remote application with FPUT (or DPUT with KCMOD='_') to a job-receiving service and the wait period for session or association occupancy is 0.

Code	Module	Cause of error and action (if any)
KD14	KCDAPRO	An asynchronous service in another application is to be addressed with APRO. The number of asynchronous jobs for this application has already reached the maximum value defined at generation (LPAP QLEV= or OSI-LPAP ALEV=).
KD15	KCDAPRO	With an MPUT EM or MPUT HM the destination in KCRN/kcrn is not an OSI TP communication partner.
KD16	KCDAPRO	No MPUT NT was given prior to an MPUT HM.
KD17	KCDAPRO	The RTAC was generated for an OSI TP partner.
KD18	KCDAPRO	A selected functional unit is not supported by this UTM version.
KD22	KCDAPRO	The commit functional unit was selected, but the abstract syntax CCR was not generated for the partner.
KD23	KCDAPRO	Mixed operation of LU6.1 and OSI TP within a distributed transaction.
KD24	KCDAPRO	Mixed operation of LU6.1 and OSI TP within a distributed transaction.
KD25	KCDAPRO	More than one ACCESS-POINT was used in a distributed transaction with OSI TP.
KD26	KCDAPRO	An abstract syntax which has not been generated was specified for an OSI/TP partner for MPUT/FPUT/DPUT in KCMF/kcfn. The syntax names "CCR" and "OSITP" are not permitted.
KD27	KCDAPRO	KCLM/kclm must be 32.
KD28	KCDAPRO	KCLM/kclm must be zero
KD29	KCDAPRO	KCLM/kclm must be equal to the length of the data structure in COPY KCAPROC or include kcapro.h.
KD30	KCDAPRO	Illegal values in the APRO data area.
KD31	KCDAPRO	KCFUCHN must contain blanks for KCFUCOM = 'N'.
KD32	KCDAPRO	Security type "Same" or "Program" was selected, but the abstract syntax UTMSEC has not been generated for the partner.
KD33	KCDAPRO	An invalid length was specified for the user ID or the password with the security type "Program".
KD34	KCDAPRO	Error on encoding the security data.
KD35	KCDAPRO	KCFUHSH = 'Y' is only permitted in the case of dialog partners.
KD36	KCDAPRO	After a CTRL PR or PE, an MPUT HM was issued to the same partner.
KD37	KCDAPRO	After a CTRL AB, an MPUT was issued to the same partner.
KD38	KCDAPRO	MPUT to job submitter, but KCSEND = NO.
KD39	KCDAPRO	The unused fields for the security function when KCSEC TYP is not equal to 'P' are not deleted. (CHAR fields to blank, numeric fields to 0).
KD40	KCDAPRO	There is no active connection assigned to the OSI-LPAP in the APRO call.
KD41	KCDAPRO	The master LPAP is locked.

Code	Module	Cause of error and action (if any)
KD42	KCSPEND	(See K348) The cluster page pool in the KDCFILE is full. Action: Generation with KDCDEF; increase size of cluster page pool with CLUSTER PGPOOL=(number,...).
KE01	KCSPADM	No authorization for the call: The user is not an administrator and the terminal is not a printer control station or is different from the terminal specified in the KDCS parameter area (KCLT/kclt).
KE02	KCSPADM	No printer (PTERM) is assigned to the printer control terminal. Check generation.
KE03	KCSPADM	A PADM call for printer acknowledgment or print repeat (KCOM = OK/PR) was issued even though there was no printout to be acknowledged for the specified printer.
KE04	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KE05	KCSPADM	A printer is to be assigned to another LTERM (KCOM = CA), but is linked with the application.
KE06	KCSPADM	Connection to a printer is to be set up, but the printer is locked.
KE07	KCSPADM	An inverse KDCDEF is running or is to be started.
KE08	KCSPADM	Administration applying globally to the cluster is not currently possible because node applications with different generations are running.
KF01	KCSINFO	The user has been generated without an ID card and no Kerberos dialog was executed for the LTERM client (INFO CD).
KF02	KCSINFO	The information is no longer available, e.g. because of a loss of connection (INFO CD).
KF03	KCSINFO	The Kerberos dialog returned an error, but Kerberos information is possibly transferred anyway (INFO CD).
KF04	KCSINFO	The Kerberos information is not complete because it exceeds the length generated in MAX PRINCIPAL-LTH (INFO CD).
KF05	KCSINFO	The service is not running under a "genuine" user ID.
KG01	KCSDADM	No authorization for the call: The user is not an administrator and the terminal is not a printer control terminal or is different from the terminal specified in the KDCS parameter area (KCLT/kclt)

Code	Module	Cause of error and action (if any)
KG02	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KG03	KCSDADM	A DADM call to delete all asynchronous messages was made (KCOM = DA) even though messages were being processed for the specified destination.
KG04	KCSDADM	An attempt was made to administer a message currently being processed.
KG05	KCSDADM	A call to delete messages (KCOM = DL/DA) was followed by another delete job or a job to modify concatenation of a message (KCOM = DL/DA/CS).
KG06	KCSDADM	An attempt was made to modify the concatenation of a time-driven message (KCOM = CS), although its start time had not yet been reached.
KG07	KCSDADM	A call to read information about messages was issued (KCOM = RQ), although there were no messages for the specified destination (or only messages currently being processed).
KG08	KCSDADM	KCQTYP invalid
KG09	KCSDADM	In the case of DADM MV with blanks in KCLT/kclt, the original destination of the message in the dead letter queue no longer exists. This message must be assigned a new destination.
KG10	KCSDADM	In the case of DADM MV, the destination specified in KCLT/kclt no longer exists.
KH01	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KH02	KCSSIGN	With SIGN CP: The new password does not satisfy the requested complexity level.
KH03	KCSSIGN	With SIGN CP: The new password is too short.
KH04	KCSSIGN	With SIGN CP: The new password is identical to the old password or to a password in the password history.
KH05	KCSSIGN	With SIGN CL: The specified language identifier is invalid.
KH06	KCSSIGN	With SIGN CL: The specified territorial identifier is invalid
KH07	KCSSIGN	With SIGN CL: The specified character set name is invalid
KH08	KCSSIGN	An inverse KDCDEF is running or is to be started.
KH09	KCSSIGN	With SIGN CP: The password may not be changed because the minimum validity period has not been exceeded yet.
KH10	KCSSIGN	With SIGN CP: A password cannot be assigned or changed for a user generated with a certificate.

Code	Module	Cause of error and action (if any)
KH11	KCSSIGN	The service was terminated abnormally because the page list for the cluster service data could not be written when the user signed off due to the fact that the cluster page pool was full.
KH12	KCSSIGN	An open cluster service cannot be continued because the sequence of TAC statements has changed following a regeneration of the node KDCFILE, or the properties of teh follow-up TAC have been changed, or the follow-up TAC or the service TAC no longer exist.
KI01	KCSISLP	VTSU recognizes that editing of dialog output message with format exit contains an error..
KI02	KCSISLP	VTSU reports invalid edit options for the dialog output message.
KI03	KCSISLP	VTSU recognizes that a dialog output message is to be sent with an invalid character set or one which is not supported by the terminal.
KJ01	KCCUSF	The service was terminated abnormally because it was marked for abnormal termination by another node application. This may have been performed via the administration functions or could be due to the fact that, even though there is a service bound to this node application, the user has signed on at another node application.
KK01	KCSSVCS	The second parameter in the KDCS call is missing.
KK02	KCSSVCS	Address validation of the second parameter in the KDCS call failed.
KK03	KCSSVCS	The KDCS call was issued from a program unit in an asynchronous service.
KK04	KCSSVCS	The KDCS operation code is unknown.
KK07	KCSSVCS	The KDCS operation code APRO is not permitted.
KK08	KCSSVCS	The call is not permitted in UTM cluster applications.
KL00	KCSSTRT	During warm start of the application no job-receiver session in the PET state was found.
KL01	KCSSTRT	During warm start of the application no valid NODE table entry was found for the OSI TP service.
KL02	KCSSTRT	The user who started this service has been deleted.
KL03	KCSSTRT	During warm start it was determined that the service cannot be continued. OSI TP job revceiving services are possibly terminated not before the association to OSI TP partner has been established and the subsequent recovery has been processed.
KL04	KCSSTRT	PEND ER triggered by KCSSTRT
KL05	KCSSTRT	PEND ER triggered by KCSSTRT
KL06	KCSSTRT	PEND ER triggered by KCSSTRT
KL07	KCSSTRT	PEND ER triggered by KCSSTRT
KL08	KCSSTRT	PEND ER triggered by KCSSTRT
KM01	KCSTRHD	TAC is not defined. [1]

Code	Module	Cause of error and action (if any)
KM02	KCSTRHD	TAC cannot be used, TAC locked. [1]
KM03	KCSTRHD	There is no key in the key set of the LTERM or (OSI-)LPAP partner for the lock on the TAC. [1]
KM04	KCSTRHD	The user does not have a key in his/her key set for the lock on the TAC. [1]
KM05	KCSTRHD	The TAC is generated as an administration TAC, but the user is not an administrator. [1]
KM07	KCSTRHD	The program unit associated with the transaction code is not linked or could not be loaded. [1]
KM08	KCSTRHD	The transaction code is completely locked. It cannot be specified as a follow-on TAC in a PEND call, nor can new jobs be specified for this TAC. [1]
KM09	KCSTRHD	The transaction code is protected by an encryption level and the caller does not support this encryption level or the input message was not encrypted with the appropriate level.
KM99	KCSTRHD	A TAC which is generated with XOPEN-API may only be used by a job submitter with whom communication takes place via the OSI TP, LU6.1 or UPIC protocol.
KN01	KCSDLO	The user who started the service has been deleted.
KQ01	KCSCRO	In the case of the QCRE WN call, the name in KCRN begins with a digit.
KQ02	KCSCRO	In the case of the QCRE NN call, KCRN was not supplied with blanks.
KQ03	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KQ04	KCSCRO	The tables for temporary queues reserved at generation by means of the QUEUE statement have been used up. Action: Delete or regenerate QUEUE objects that are no longer required.
KQ05	KCSCRO	In the case of the QCRE WN call, KCRN contains an invalid character or is not filled with blanks.
KR01	KDCRTMM	A dialog program unit with which a service was meant to continue is missing (not linked). [1]
KR02	KDCRTMM	An asynchronous program unit for which a message is awaiting processing is missing (not linked). [1]
KR04	KDCRTMM	A program unit run prior to the last KDCS call continued writing beyond the end of the KB. Action: Change program unit or generation.
KR05	KDCRTMM	A program unit run prior to the last KDCS call continued writing beyond the end of the SPAB. Action: Change program unit or generation.

Code	Module	Cause of error and action (if any)
KR06	KDCRTMM	A KDCS call was made by the VORGANG exit program. The service aborts with PEND ER.
KR07	KDCRTMM	A KDCS call was issued by the FORMAT exit program unit. The service aborts with PEND ER.
KR08	KDCRTMM	Illegal DB USER call in sign-on service. The service aborts with PEND ER.
KR09	KDCRTMM	Error when loading a program unit. The service aborts with PEND ER.
KR10	KDCRTMM	Program for VORGANG exit is missing. The service aborts with PEND ER.
KS00	KCDSEFU	In connection with a transaction end request, MPUTs were made for two or more partners.
KS01	KCDSEFU	In connection with a transaction end request, at least one partner has an illegal transaction or service status.
KS02	KCDSEFU	In connection with a transaction end request, at least one session was not used (too many APRO calls).
KS03	KCDSEFU	MPUT was issued to a service ID by the job submitterand teh transaction was with subsequently terminated with PEND RE even though all job submitters had not yet reported PEND RE/FI.
KS04	KCDSEFU	A message was sent with PEND KP to an LU6.1 partner which has already initiated transaction end.
KS05	KCDSEFU	If the session is both job receiver and job submitter, then it may not send PEND RE to its job receiver until after receiving a PEND RE message from its job submitter.
KS06	KCDSEFU	No asynchronous message was issued for the addressed service following an APRO AM.
KS07	KCDSEFU	PEND SP was issued even though there is at least one partner with an open transaction.
KS08	KCDSEFU	The OSI TP client sent a message to the OSI TP server without terminating the transaction, or has requested the OSI TP server to terminate the transaction although the server has already initiated transaction termination.
KS09	KCDSEFU	The OSI TP server has not terminated the transaction even though it was requested to do so by the OSI TP client.
KS10	KCDSEFU	Contrary to the request of the OSI TP client, the OSI TP server terminated the transaction with PEND SP.
KS11	KCDSEFU	The OSI TP client sent a message to the OSI TP server and terminated the transaction, although it does not have the send authorization for transaction termination, since it is an OSI TP server in a different dialog and its OSI TP client has not relinquished the send authorization for transaction termination.
KS12	KCDSEFU	Contrary to the request of the OSI TP client, the OSI TP server terminated the transaction with PEND RE.

Code	Module	Cause of error and action (if any)
KS13	KCDSEFU	The OSI TP client terminated the service with PEND FI although it still has an open server conversation without COMMIT functionality.
KS14	KCDSEFU	Contrary to the request of the OSI TP client, the OSI TP server terminated the transaction with PEND FI.
KS15	KCDSEFU	The OSI TP client terminated the service with PEND FC although it still has an open server conversation without COMMIT functionality.
KS16	KCDSEFU	The OSI TP server sent a message to the OSI TP client although it does not possess send authorization on this dialog.
KS17	KCDSEFU	The OSI TP server terminated the dialog step, but did not send a message to the OSI TP client although it possesses send authorization on this dialog.
KS18	KCDSEFU	The OSI TP client relinquished send authorization for transaction termination to more than one OSI TP server.
KS19	KCDSEFU	The OSI TP client requested the OSI TP server to terminate the transaction or service with CTRL PR or PE, sent a message to this partner and then initiated transaction termination itself.
KS20	KCDSEFU	The local service has called PGWT CM or PGWT RB although a partner is involved in the distributed transaction and the LU6.1 protocol is used to communicate with the partner.
KS21	KCDSEFU	The OSI TP server has terminated the transaction with PGWT CM contrary to the request of the OSI TP client.
KS22	KCDSEFU	The OSI TP client terminated the service with PEND FI although it only requested a server conversation to terminate the transaction with CTRL PR.
KS23	KCDSEFU	The OSI TP client terminated the service with PEND FC although it only requested a server conversation to terminate the transaction with CTRL PR.
KS24	KCDSEFU	The OSI TP client requested a server conversation to terminate the trans-action or service with CTRL PR/PE even though no message has been sent to the server conversation.
KS25	KCDSEFU	Session restart has failed.
KT01	KCSPEND	(See K302) The task-specific buffer area for restart information is too small (see UTM manual Generating appllications, restart area). Action: Define a larger buffer area with MAX RECBUF=(...,length) in KDCDEF generation. [1]
KT02	KCSPEND	(See K345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
KT03	KCSPEND	(See K306) The transaction and the service had to be reset since the DB system reset the DB transaction when closing down.

Code	Module	Cause of error and action (if any)
KT04	KCDOSTM	A log record is too large for the input buffer. Response: address fewer OSI TP job-receivers in this service [1]
KT05	KCDOSTM	The transaction must be rolled back because the database transaction was rolled back.
KT06	KCDOSTM	The database reported that the transaction was rolled back when the transaction was rolled terminated after a Prepare statement.
KT07	KCDOSTM	An OSI TP server reported a heuristic decision (heuristic mixed).
KT08	KCDOSTM	An OSI TP server reported a heuristic decision (heuristic hazard).
KU04	KDCRTDB	The DB transaction had to be reset. UTM also resets the UTM transaction: Control is restored to the program unit run, as with the UTM call RSET.
KU08	KDCRTDB	UTM has to terminate the service with PEND ER. The terminal user is sent a message announcing the termination of the service. The first 4 bytes of the DB error message are included in the display.
KU0C	KDCRTDB	The DB system (the DBH) is not or no longer available. UTM terminates the application abnormally or aborts application startup.
KU10	KDCRTDB	The DB system is no longer available due to DB administrator action. UTM terminates application abnormally.
KU14[1]	KDCRTDB	The DB system is momentarily not available. A new attempt to set up the connection is made after 5 seconds.
KU18	KDCRTDB	A (possibly recoverable) DB system error occurred.
KU1C	KDCRTDB	An unrecoverable DB system error occurred. UTM terminates the application abnormally.
KU20	KDCRTDB	The DB system detected a user error. The error occurs at application startup when the DB-specific start parameters are checked. The DB error message is output to SYSOUT.
KU24	KDCRTDB	UTM behavior unexpected by the DB system. The application is terminated with PEND06.
KU99	KDCRTDB	With multi-DB systems: After a PETA call, the DB system cannot execute the FITA call.
KV01	KCSTUTM	Socket partner sends with USP version 1.0. Buffer for dialog messages generated too small or MPUT length has exceeded the maximum value. Action: Change generation; select larger buffer with MAX NB=length.
KV02	KCSPEND	(See K345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]

Code	Module	Cause of error and action (if any)
KV03	KCSPEND	(See K348) The cluster page pool in the KDCFILE is full. Action: Generation with KDCDEF; increase size of cluster page pool with CLUSTER PGPOOL=(number,...).
KW01	KCDOSSF	TP-HANDSHAKE-IND received.
KW02	KCDOSSF	TP-BEGIN-DIALOGUE-CNF received (negative).
KW03	KCDOSSF	TP-U-ERROR-IND received from a partner which does not have send authorization.
KW04	KCDOSSF	Timer expired.
KW05	KCDOSSF	TP-U-ABORT-IND or TP-P-ABORT-IND received.
KW06	KCDOSSF	TP-ROLLBACK-IND received.
KW07	KCDOSSF	openUTM does not support reception of TP-END-DIALOGUE-IND from a client.
KW08	KCDOSSF	The dialog with the client was not terminated with the first transaction in the unchained functions functional unit.
KY00	KCDSYPM	A negative acknowledgment was received from the partner.
KY01	KCDSYPM	Timeout when reserving a session.
KY02	KCDSYPM	The job-receiving service was terminated with PEND ER.
KY03	KCDSYPM	The job-submitting service was terminated with PEND ER.
KY04	KCDSYPM	Mismatch with the partner, or the local service is in the PET status.
KY05	KCDSYPM	Mismatch with the partner.
KY06	KCDSYPM	Mismatch with the database.
KY07	KCDSYPM	Loss of connection to the partner.
NOTA	KCDRTDB	Transaction is unknown at the XA Resource Manager. (May occur, for example, after timeouts with the Oracle database).
UPCP	KCSPEND	(See K348) The cluster page pool in the KDCFILE is full. Action: Generation with KDCDEF; increase size of cluster page pool with CLUSTER PGPOOL=(number,...).
UPDE	KCSUPIC	The UPIC partner has required a service abort.
UPPG	KCSPEND	(See K345) The page pool in KDCFILE is full. Action: Change generation with KDCDEF, enlarge page pool with MAX PGPOOL=(number,...). [1]
UPRS	KCSUPIC	The UPIC partner has suppressed a service restart. This occurs when a service restart would have been possible, but was not required by the UPIC partner. UTM then resets the service.

Code	Module	Cause of error and action (if any)
UPXC	KCSUPIC	In a program unit with X/OPEN-API, the message to the UPIC job submitter is to be concluded even though a message to a job receiver service has already been concluded.
UPXM	KCSUPIC	In a program unit with X/OPEN-API, the message to the UPIC job submitter is to be concluded even though a message to the UPIC job submitter has already been concluded.
XTnn	KDCROOT	The STXIT routine was called in KDCROOT, nn = event code of the STXIT macro. XT80[1] see footnote.

[1] When TESTMODE=OFF, these PENDER dumps are suppressed. Dynamic loading is suppressed at the same time unless for the current TAC, PGWT calls are permitted and the KDCS call at which the error occurred was not a PEND call.

6.3 STXIT events

The table below shows the assignment of event codes to possible STXIT events. They have been taken from the "Executive Macros" manual.

STXIT event class	STXIT operand	Interrupt error	Event code in R3	Type of queue	Max. nesting depth
Program error	PROCHK	Illegal SVC	X' 04'	LIFO	127
		Illegal operation code	X' 58'		
		Data error	X' 60'		
		Exponent overflow	X' 64'		
		Division error	X' 68'		
		Mantissa = 0	X' 6C'		
		Exponent underflow	X' 70'		
		Decimal overflow	X' 74'		
		Fixed point overflow	X' 78'		
Interval timer CPU time	TIMER	"SETIC interval" expired for CPU time	X' 20'	FIFO	127
Interval timer real time	RTIME	"SETIC interval" expired for real time	X' A0'	FIFO	127
End of program run time	RUNOUT	CPU time limit for task or program exceeded	X' 80'	FIFO	0
Unrecoverable program error	ERROR	Privileged SVC	X' 08'	LIFO	127
		Access to non-existent	X' 48'		
		memory page	X' 54'		
		Privileged operation	X' 5C'		
		Address error	X' 9C'		
		XA error (incorrect addressing mode	X' A4'		
		Real timer (condition error)	X' AC'		
		Alignment error	X' B0'		
		Validation error	X' C4'		
		Invalid UNIT number in	X' 08'		
		standard header	X' 48'		

STXIT event class	STXIT operand	Interrupt error	Event code in R3	Type of queue	Max. nesting depth
Notification to program	INTR	INFORM-PROGRAM-command	X' 44'	LIFO	127
ESCPBRK	ESCPBRK	BREAK/ESCAPE (using keys)	X' 84'	LIFO	127
Program terminated by asynchronous events	ABEND	System error, disconnection	X' 88'	LIFO	0
		START-EXECUTABLE-PROGRAM, LOAD-EXECUTABLE-PROGRAM, ABEND, EXIT-JOB, CANCEL JOB	X' 8C'		
		Address translation error due to hardware error	X' 94'		
		Hardware error (CPU)	X' A8'		
		Forced unloading of a subsystem (system admin.)	X' B8'		
		Unrecoverable DMS error	X'BC'		
Program terminated by synchronous events	TERM	TERM	X' 90'	LIFO	0
		Program terminated by CMD / LOGOFF macro	X' 98'		
SVC interrupt	SVC	Call of a specified SVC	X' 50'	LIFO	127
Hardware error	HWERROR	I/O error in data in virtual technique	X' 28'	LIFO	0

6.4 Codes from openUTM CDUMP calls

Some errors, e.g. errors in return codes from the language connection modules, FHS, the DB connection module, or openUTM errors, result in the generation of a CDUMP. A prerequisite for this is that the generation of dumps has not been deactivated by the user with /MODIFY-JOB-OPTIONS. A message with the insert 'CODE' is output to SYSOUT. This insert helps to identify the cause of the error which initiated the CDUMP.

CDUMPs are stored under the user ID under which the UTM application was started.

This return code, together with the openUTM version and the ROOTDATA address, is contained in the edited dump obtained using a program for dump editing.

If the UTM application was started with TESTMODE=OFF, then the CDUMP is suppressed with reason KDCCK02, depending on the return code KCRRCDC (see).

If the dump occurs following a STXIT event, then the address at which the CONTXT area is located is also output.

The CONTXT area has the following structure:

contents of registers R0 through R15	
program counter	1 byte
instruction length count	1 byte
condition code	1 byte
program mask	1 byte
address mode	1 byte
interrupt weight	4 bytes

KDCBF..

CODE	Error cause	openUTM response
KDCBF01	The SVC UTM returns a return code with errors.	Terminates the task with TERM UNIT=STEP

KDCCK..

CODE	Error cause	openUTM response
KDCCK01	The UTM SVC supplied an incorrect return code.	Terminates the task with TERM UNIT=STEP
KDCCK02[1]	Upon returning from the UTM system code, the UTM linkage program ROOT detected that KCRCCC >= 70Z	The task concerned is stopped by PEND ER. If it is a batch task, the program is subsequently unloaded; with a dialog task it remains loaded and continues
KDCCK03	The language connection module supplied an incorrect return code at the start of the program unit ' INPUT-EXIT'	The task concerned is stopped by PEND ER. If it is a batch task, the program subsequently unloaded; with a dialog task it remains loaded and continues
KDCCK04	CALL „KDCS" called from START, SHUT or INPUT exit.	The relevant task is termi-nated with TERM UNIT=STEP.
KDCCK05	openUTM detects an error in the INPUT exit: output parameters are inconsistent.	
KDCCK07	Invalid call of program unit at the format system.	Terminates the task with TERM UNIT=STEP
KDCCK08	Possible causes: – start error – database is called but is not connected; – formatting system is called but is not generated (for diagnostics see CODEOFP2 field in table module).	Terminates the task with TERM UNIT=STEP
KDCCK09	This CDUMP occurs after a PEND ER (provided a CDUMP did not occur already).	A batch task is terminated with TERM UNIT=PROG. A dialog task continues.

CODE	Error cause	openUTM response
KDCCK10	The language connection module supplied an incorrect return code at the start of a program unit.	The task concerned is stopped by PEND ER. If it is a batch task, the program is subsequently unloaded; with a dialog task it remains loaded and continues.
KDCCK11	The language connection module supplied an incorrect return code at the end of a program unit.	Terminates the task with TERM UNIT=STEP.
KDCCK13	This CDUMP occurs if an incorrect return code is issued in the START EXIT by the corresponding language module.	Terminates the task with TERM UNIT=STEP
KDCCK14	An invalid load module index is specified when exchanging an individual load module.	Terminates the task with TERM UNIT=STEP
KDCCK16	Error in ILCS reinitialization; occurs after loading STARTUP or ONCALL programs or after program exchange	Terminates the task with TERM UNIT=STEP
KDCCK99	Internal error in UTM root: structural error in KDCRTMM	Terminates the task with TERM UNIT=STEP

[1] With TESTMODE=OFF, the CDUMP is suppressed, depending on the return code KCRRCDC and dynamic loading is not carried out.

KDCRT..

CODE	Error cause	openUTM response
KDCRT04	Internal error in UTM root: incorrect ACT_INDEX in KDCRTMM	Terminates the task with TERM UNIT=STEP

KDCLD..

CODE	Error cause	openUTM response
KDCLD01	Internal error in UTM root. Invalid load mode for a load module on starting the application.	After the dump, the task concerned is terminated with TERM UNIT=STEP.
KDCLD02	Internal error in UTM root. Invalid opcode on calling KDCRTLD.	After the dump, the task concerned is terminated with TERM UNIT=STEP.
KDCLD03	Internal error in UTM root. Invalid load status for a load module.	The task is terminated with TERM UNIT=STEP
KDCLD04	The end mark of the linkage section (AREAs) has been overwritten (DEF - ROOT possibly inconsistent).	The task is terminated with TERM UNIT=STEP
KDCLD05	The load module in which AREAs should be loaded has not been loaded. See message K078.	The task is terminated with TERM UNIT=STEP
KDCLD99	Faulty return code for BIND / ASHARE call and DEQAR (with Term). See message K078.	The task is terminated with TERM UNIT=STEP

KDCAF.. , KDCST..

CODE	Error cause	openUTM response
KDCAF01	Stack management error with ILCS in openUTM. See message K078.	The task is terminated with TERM UNIT=STEP
KDCST01	Error on initializing ILCS or SPL or timeout waiting for connection to a memory pool. See message K078.	The task is terminated with TERM UNIT=STEP

KDCDI..

CODE	Error cause	openUTM response
KDCDI01	The STXIT exit of the language connection module supplied an incorrect return code during STXIT processing.	The task concerned is stopped by PEND ER and is subsequently terminated with TERM UNIT=STEP.
KDCDI02	The language connection module supplied an incorrect return code in conjunction with the SHUT-EXIT call during STXIT processing.	The language connection modules are deactivated and the task is then terminated with TERM UNIT=PROG.
KDCDI03	The language connection module supplied an incorrect return code in conjunction with ENDE handling in the STXIT routine.	SHUTEXIT is called; the language connection modules are deactivated and the task is then terminated with TERM UNIT=STEP.
KDCDI04	Error when initializing the language connection modules	Terminates the task with TERM UNIT=STEP
KDCDI05	Error when deactivating the language connection modules	No further error handling
KDCDI07	A STXIT event has arisen in the start phase (the KTA address is incorrect).	Terminates the task with TERM UNIT=STEP
KDCDI08	A STXIT event has arisen in the start phase (the root data address incorrect).	Terminates the task with TERM UNIT=STEP
KDCDI09	A STXIT event has arisen in the start phase (DB-INIT has not been run)	Terminates the task with TERM UNIT=STEP

KDCDB..

CODE	Error cause	openUTM response
KDCDB01	Error of DB connection module (=dBCON) when transferring the start parameters	Terminates the task with TERM UNIT=STEP
KDCDB02	ADMINISTRATION ERROR or DBMS-DOWN when setting up database connection	Terminates the task with TERM UNIT=STEP
KDCDB03	ECOVERABLE or UNRECOVERABLE ERROR when disconnecting the database	The task continues
KDCDB04	DBMS-DOWN but KTA still available after DB transaction 'DELETE DB STATUS'	Application termination with PEND03
KDCDB05	DB return code = TA-CHAIN-RSET	The task continues with PEND ER
KDCDB06	DB return code = DBMS-DOWN	Application termination with PEND03
KDCDB07	The DB return code supplied is not permitted with this DB function (see interface IUTMDB).	Terminates the task with TERM UNIT=STEP; or terminates the application with PEND04 With OPCODE=STAT (see K071 message), start aborted with error code 74 or 82
KDCDB08	UNRECOVERABLE ERROR in DB call	Terminates application with PEND05
KDCDB09	USER CALL in sign-on service, although not allowed by generation	The task continues to run after PEND ER
KDCDB10	USER CALL in INPUT exit, although not allowed	Task terminates with TERM UNIT=STEP
KDCDB11	Illegal USER CALL in START or SHUT exit	The task continues to run
KDCDB12	Error reported by the DB system (diagnostic dump) during task	Task continues termination
KDCDB99	Error when communicating with the DB system (with multi-DB system): The DB system cannot execute the FITA call after a PETA call (= temporary transaction end).	Task continues running after PEND ER

KDCFH..

CODE	Error cause	openUTM response
KDCFH01	The FORM return code supplied is not permitted with this FORM funct.	Terminates with TERM UNIT=STEP
KDCFH02	Initialization of the formatting system could not be executed properly	Terminates with TERM UNIT=STEP
KDCFH03	Negative return code from format system, not FORM_ADMINISTRATION_ERR or FORM_ERR	No further error handling
KDCFH04	The language connection module supplied an incorrect return code at the start of the 'FORMAT EXIT' program unit	The task concerned is stopped by PEND ER. If it is a batch task, the program is subsequently unloaded; with a dialog task it remains loaded and continues.
KDCFHxx	Error when initializing the format system xx = error code supplied by the format system	Terminates with TERM UNIT=STEP

KDCMOVE, KCSMOVE

CODE	Error cause	openUTM response
KDCMOVE or KCSMOVE	This CDUMP corresponds to the TRMA reasons MOVExx. It is produced if a TRMA call cannot be issued (e.g. in the TU section of the administration).	Terminates the task with TERM UNIT=STEP

KDCIL60

CODE	Error cause	openUTM response
KDCIL60	This CDUMP is created in the ILCS error handling routine. Possible cause: insufficient memory	Terminates the task with UNIT=PROGRAM

Tabelle 2:

KDCIL70

CODE	Error cause	openUTM response
KDCIL70	This CDUMP is created in the STXIT handling routine when the ILCS stacks are corrupt. Possible cause: impermissible mixture of ILCS and non-ILCS programs or incorrect generation (also ILCS/non-ILCS mix).	Terminates the task with UNIT=PROGRAM

Tabelle 3:

KDCIW.., KDCSTXT, KDCSTRT

CODE	Error cause	openUTM response
KDCIW..	This CDUMP is produced by the ASSEMBLER, COBOL and SPL language connection modules and the KDCRTDI module (with RUNOUT, NORMAL PROGR. TERMINATION and ABEND) whenever an STXIT occurs. Insert '..' = interrupt weight	A batch task is terminated with PEND ER, a dialog task with TERM UNIT=STEP
KDCIW88	Possible cause: The task was terminated due to UTM system coding. The PC then points to the command after the SVC 134	
KDCSTRT	A STXIT event before or during the initialization of ILCS. Possible cause: installation error	Terminates the task with TERM UNIT=STEP
KDCSTXT	A STXIT event occurred in a loop (number > 3) under ISRKEY E.	Terminates the task with TERM UNIT=STEP

KCSDTTM

CODE	Error cause	openUTM response
KCSDTTM	This CDUMP is produced if the condition code is not equal to 0 following the STCK command for system time checks. The time received is in this case not suited for openUTM to continue processing (only occurs with hardware errors).	Terminates the task with TERM UNIT=STEP

KCSEN..,

CODE	Error cause	Action
KCSEN01 KCSEN02 KCSEN03 KCSEN04 KCSEN05	UTM internal error	System error: write PM

KCSST..

CODE	Error cause	Action
KCSST01	UTM internal system error	System error: write PM

Tabelle 4:

KCSSW..

CODE	Error cause	Action
KCSSW01 KCSSW02 KCSSW03	UTM internal error in F2 mode	System error: write PM

KCSWT..

CODE	Error cause	Action
KCSWT02	Invalid timer announcement	System error: write PM
KCSWT03	Invalid market announcement	System error: write PM

Tabelle 5:

Glossary

A term in *italic* font means that it is explained somewhere else in the glossary.

abnormal termination of a UTM application

Termination of a *UTM application*, where the *KDCFILE* is not updated. Abnormal termination is caused by a serious error, such as a crashed computer or an error in the system software. If you then restart the application, openUTM carries out a *warm start*.

abstract syntax (OSI)

Abstract syntax is defined as the set of formally described data types which can be exchanged between applications via *OSI TP*. Abstract syntax is independent of the hardware and programming language used.

acceptor (CPI-C)

The communication partners in a *conversation* are referred to as the *initiator* and the acceptor. The acceptor accepts the conversation initiated by the initiator with Accept_Conversation.

access list

An access list defines the authorization for access to a particular *service*, *TAC queue* or *USER queue*. An access list is defined as a *key set* and contains one or more *key codes*, each of which represent a role in the application. Users or LTERMs or (OSI) LPAPs can only access the service or *TAC queue*/*USER queue* when the corresponding roles have been assigned to them (i.e. when their *key set* and the access list contain at least one common *key code)*.

access point (OSI)

See *service access point*.

ACID properties

Acronym for the fundamental properties of *transactions*: atomicity, consistency, isolation and durability.

administration

Administration and control of a *UTM application* by an *administrator* or an *administration program*.

administration command

Commands used by the *administrator* of a *UTM application* to carry out administration functions for this application. The administration commands are implemented in the form of *transaction codes*.

administration journal

See *cluster administration journal*.

administration program

Program unit containing calls to the *program interface for administration*. This can be either the standard administration program *KDCADM* that is supplied with openUTM or a program written by the user.

administrator

User who possesses administration authorization.

AES

AES (Advanced Encryption Standard) is the current symmetric encryption standard defined by the National Institute of Standards and Technology (NIST) and based on the Rijndael algorithm developed at the University of Leuven (Belgium). If the AES method is used, the UPIC client generates an AES key for each session.

Apache Axis

Apache Axis (Apache eXtensible Interaction System) is a SOAP engine for the design of Web services and client applications. There are implementations in C++ and Java.

Apache Tomcat

Apache Tomcat provides an environment for the execution of Java code on Web servers. It was developed as part of the Apache Software Foundation's Jakarta project. It consists of a servlet container written in Java which can use the JSP Jasper compiler to convert JavaServer pages into servlets and run them. It also provides a fully featured HTTP server.

application cold start

See *cold start*.

application context (OSI)

The application context is the set of rules designed to govern communication between two applications. This includes, for instance, abstract syntaxes and any assigned transfer syntaxes.

application entity (OSI)

An application entity (AE) represents all the aspects of a real application which are relevant to communications. An application entity is identified by a globally unique name ("globally" is used here in its literal sense, i.e. worldwide), the *application entity title* (AET). Every application entity represents precisely one *application process*. One application process can encompass several application entities.

application entity qualifier (OSI)

Component of the *application entity title*. The application entity qualifier identifies a *service access point* within an application. The structure of an application entity qualifier can vary. openUTM supports the type "number".

application entity title (OSI)

An application entity title is a globally unique name for an *application entity* ("globally" is used here in its literal sense, i.e. worldwide). It is made up of the *application process title* of the relevant *application process* and the *application entity qualifier*.

application information

This is the entire set of data used by the *UTM application*. The information comprises memory areas and messages of the UTM application including the data currently shown on the screen. If operation of the UTM application is coordinated with a database system, the data stored in the database also forms part of the application information.

application process (OSI)

The application process represents an application in the *OSI reference model*. It is uniquely identified globally by the *application process title*.

application process title (OSI)

According to the OSI standard, the application process title (APT) is used for the unique identification of applications on a global (i.e. worldwide) basis. The structure of an application process title can vary. openUTM supports the type *Object Identifier*.

application program

An application program is the core component of a *UTM application*. It comprises the main routine *KDCROOT* and any *program units* and processes all jobs sent to a *UTM application*.

application restart

see *warm start*

---

application service element (OSI)
An application service element (ASE) represents a functional group of the application layer (layer 7) of the *OSI reference model*.

application warm start
see *warm start*.

association (OSI)
An association is a communication relationship between two application entities. The term "association" corresponds to the term *session* in *LU6.1*.

asynchronous conversation
CPI-C conversation where only the *initiator* is permitted to send. An asynchronous transaction code for the *acceptor* must have been generated in the *UTM application*.

asynchronous job
Job carried out by the job submitter at a later time. openUTM includes *message queuing* functions for processing asynchronous jobs (see *UTM-controlled queue* and *service-controlled queue)*. An asynchronous job is described by the *asynchronous message*, the recipient and, where applicable, the required execution time. If the recipient is a terminal, a printer or a transport system application, the asynchronous job is a *queued output job*. If the recipient is an *asynchronous service* of the same application or a remote application, the job is a *background job*. Asynchronous jobs can be *time-driven jobs* or can be integrated in a *job complex*.

asynchronous message
Asynchronous messages are messages directed to a *message queue*. They are stored temporarily by the local *UTM application* and then further processed regardless of the job submitter. Distinctions are drawn between the following types of asynchronous messages, depending on the recipient:
– In the case of asynchronous messages to a *UTM-controlled queue,* all further processing is controlled by openUTM. This type includes messages that start a local or remote *asynchronous service* (see also *background job*) and messages sent for output on a terminal, a printer or a transport system application (see also *queued output job*).
– In the case of asynchronous messages to a *service-controlled queue*, further processing is controlled by a *service* of the application. This type includes messages to a *TAC queue*, messages to a *USER queue* and messages to a *temporary queue*. The USER queue and the temporary queue must belong to the local application, whereas the TAC queue can be in both the local application and the remote application.

asynchronous program

Program unit started by a *background job*.

asynchronous service (KDCS)

Service which processes a *background job*. Processing is carried out independently of the job submitter. An asynchronous service can comprise one or more program units/transactions. It is started via an asynchronous *transaction code*.

audit (BS2000 systems)

During execution of a *UTM application,* UTM events which are of relevance in terms of security can be logged by *SAT* for auditing purposes.

authentication

See *system access control*.

authorization

See *data access control*.

Axis

See *Apache Axis*.

background job

Background jobs are *asynchronous jobs* destined for an *asynchronous service* of the current application or of a remote application. Background jobs are particularly suitable for time-intensive processing or processing which is not time-critical and where the results do not directly influence the current dialog.

basic format

Format in which terminal users can make all entries required to start a service.

basic job

Asynchronous job in a *job complex*.

browsing asynchronous messages

A *service* sequentially reads the *asynchronous messages* in a *service-controlled queue*. The messages are not locked while they are being read and they remain in the queue after they have been read. This means that they can be read simultaneously by different services.

bypass mode (BS2000 systems)

Operating mode of a printer connected locally to a terminal. In bypass mode, any *asynchronous message* sent to the printer is sent to the terminal and then redirected to the printer by the terminal without being displayed on screen.

---

cache

Used for buffering application data for all the processes of a *UTM application*. The cache is used to optimize access to the *page pool* and, in the case of UTM cluster applications, the *cluster page pool*.

CCS name (BS2000 systems)

See *coded character set name*.

client

Clients of a *UTM application* can be:
– terminals
– UPIC client programs
– transport system applications (e.g. DCAM, PDN, CMX, socket applications or UTM applications which have been generated as *transport system applications*).

Clients are connected to the UTM application via LTERM partners.
openUTM clients which use the OpenCPIC carrier system are treated just like *OSI TP partners*.

client side of a conversation

This term has been superseded by *initiator*.

cluster

A number of computers connected over a fast network and which in many cases can be seen as a single computer externally. The objective of clustering is generally to increase the computing capacity or availability in comparison with a single computer.

cluster administration journal

The cluster administration journal consists of:
– two log files with the extensions JRN1 and JRN2 for global administration actions,
– the JKAA file which contains a copy of the KDCS Application Area (KAA). Administrative changes that are no longer present in the two log files are taken over from this copy.

The administration journal files serve to pass on to the other node applications those administrative actions that are to apply throughout the cluster to all node applications in a UTM cluster application.

cluster configuration file

File containing the central configuration data of a *UTM cluster application*. The cluster configuration file is created using the UTM generation tool *KDCDEF*.

cluster filebase

Filename prefix or directory name for the *UTM cluster files*.

cluster GSSB file

File used to administer GSSBs in a *UTM cluster application*. The cluster GSSB file is created using the UTM generation tool *KDCDEF*.

cluster lock file

File in a *UTM cluster application* used to manage cross-node locks of user data areas.

cluster page pool

The cluster page pool consists of an administration file and up to 10 files containing a *UTM cluster application's* user data that is available globally in the cluster (service data including LSSB, GSSB and ULS). The cluster page pool is created using the UTM generation tool *KDCDEF*.

cluster start serialization file

Lock file used to serialize the start-up of individual node applications (only in Unix systems and Windows systems).

cluster ULS file

File used to administer the ULS areas of a *UTM cluster application*. The cluster ULS file is created using the UTM generation tool *KDCDEF*.

cluster user file

File containing the user management data of a *UTM cluster application*. The cluster user file is created using the UTM generation tool *KDCDEF*.

coded character set name (BS2000 systems)

If the product *XHCS* (e**X**tended **H**ost **C**ode **S**upport) is used, each character set used is uniquely identified by a coded character set name (abbreviation: "CCS name" or "CCSN").

cold start

Start of a *UTM application* after the application terminates normally (*normal termination*) or after a new generation (see also *warm start*).

communication area (KDCS)

KDCS *primary storage area*, secured by transaction logging and which contains service-specific data. The communication area comprises 3 parts:
– the KB header with general service data
– the KB return area for returning values to KDCS calls

–   the KB program area for exchanging data between UTM program units
within a single *service*.

communication resource manager

In distributed systems, communication resource managers (CRMs) control
communication between the application programs. openUTM provides CRMs
for the international OSI TP standard, for the LU6.1 industry standard and for
the proprietary openUTM protocol UPIC.

configuration

Sum of all the properties of a *UTM application*. The configuration describes:
–   application parameters and operating parameters
–   the objects of an application and the properties of these objects. Objects
can be *program units* and *transaction codes*, communication partners,
printers, *user IDs*, etc.
–   defined measures for controlling data and system access.
The configuration of a UTM application is defined at generation time (*static con-
figuration*) and can be changed dynamically by the administrator (while the
application is running, *dynamic configuration*). The configuration is stored in the
KDCFILE.

confirmation job

Component of a *job complex* where the confirmation job is assigned to the *basic
job*. There are positive and negative confirmation jobs. If the *basic job* returns a
positive result, the positive confirmation job is activated, otherwise, the negative
confirmation job is activated.

connection bundle

see *LTERM bundle*.

connection user ID

User ID under which a *TS application* or a *UPIC client* is signed on at the *UTM
application* directly after the connection has been established. The following
applies, depending on the client (= LTERM partner) generation:
–   The connection user ID is the same as the USER in the LTERM statement
(explicit connection user ID). An explicit connection user ID must be
generated with a USER statement and cannot be used as a "genuine" *user
ID*.

– The connection user ID is the same as the LTERM partner (implicit connection user ID) if no USER was specified in the LTERM statement or if an LTERM pool has been generated.
In a *UTM cluster application*, the service belonging to a connection user ID (RESTART=YES in LTERM or USER) is bound to the connection and is therefore local to the node.
A connection user ID generated with RESTART=YES can have a separate service in each *node application*.

contention loser

Every connection between two partners is managed by one of the partners. The partner that manages the connection is known as the *contention winner*. The other partner is the contention loser.

contention winner

A connection's contention winner is responsible for managing the connection. Jobs can be started by the contention winner or by the *contention loser*. If a conflict occurs, i.e. if both partners in the communication want to start a job at the same time, then the job stemming from the contention winner uses the connection.

conversation

In CPI-C, communication between two CPI-C application programs is referred to as a conversation. The communication partners in a conversation are referred to as the *initiator* and the *acceptor*.

conversation ID

CPI-C assigns a local conversation ID to each *conversation*, i.e. the *initiator* and *acceptor* each have their own conversation ID. The conversation ID uniquely assigns each CPI-C call in a program to a conversation.

CPI-C

CPI-C (Common Programming Interface for Communication) is a program interface for program-to-program communication in open networks standardized by X/Open and CIW (**C**PI-C **I**mplementor's **W**orkshop).
The CPI-C implemented in openUTM complies with X/Open's CPI-C V2.0 CAE Specification. The interface is available in COBOL and C. In openUTM, CPI-C can communicate via the OSI TP, *LU6.1* and UPIC protocols and with openUTM-LU62.

Cross Coupled System / XCS

Cluster of BS2000 computers with the *Highly Integrated System Complex* Multiple System Control Facility (HIPLEX**®** MSCF).

data access control

In data access control openUTM checks whether the communication partner is authorized to access a particular object belonging to the application. The access rights are defined as part of the configuration.

dead letter queue

The dead letter queue is a TAC queue which has the fixed name KDCDLETQ. It is always available to save queued messages sent to transaction codes or TAC queues but which could not be processed. The saving of queued messages in the dead letter queue can be activated or deactivated for each message destination individually using the TAC statement's DEAD-LETTER-Q parameter.

DES

DES (Data Encryption Standard) is an international standard for encrypting data. One key is used in this method for encoding and decoding. If the DES method is used, the UPIC client generates a DES key for each session.

dialog conversation

CPI-C conversation in which both the *initiator* and the *acceptor* are permitted to send. A dialog transaction code for the *acceptor* must have been generated in the *UTM application*.

dialog job, interactive job

Job which starts a *dialog service*. The job can be issued by a *client* or, when two servers communicate with each other (*server-server communication*), by a different application.

dialog message

A message which requires a response or which is itself a response to a request. The request and the response both take place within a single service. The request and reply together form a dialog step.

dialog program

Program unit which partially or completely processes a *dialog step*.

dialog service

Service which processes a *job* interactively (synchronously) in conjunction with the job submitter (*client* or another server application) . A dialog service processes *dialog messages* received from the job submitter and generates dialog messages to be sent to the job submitter. A dialog service comprises at least one *transaction.* In general, a dialog service encompasses at least one dialog step. Exception: in the event of *service chaining,* it is possible for more than one service to comprise a dialog step.

dialog step

A dialog step starts when a *dialog message* is received by the *UTM application*. It ends when the UTM application responds.

dialog terminal process (Unix systems/Windows systems)

A dialog terminal process connects a terminal of a Unix system or a Windows system with the work processes of the *UTM application*. Dialog terminal processes are started either when the user enters utmdtp or via the LOGIN shell. A separate dialog terminal process is required for each terminal to be connected to a UTM application.

Distributed Lock Manager / DLM (BS2000 systems)

Concurrent, cross-computer file accesses can be synchronized using the Distributed Lock Manager.
DLM is a basic function of HIPLEX**®** MSCF.

distributed processing

Processing of *dialog jobs* by several different applications or the transfer of *background jobs* to another application. The higher-level protocols *LU6.1* and *OSI TP* are used for distributed processing. openUTM-LU62 also permits distributed processing with LU6.2 partners. A distinction is made between distributed processing with *distributed transactions* (transaction logging across different applications) and distributed processing without distributed transactions (local transaction logging only). Distributed processing is also known as server-server communication.

distributed transaction

Transaction which encompasses more than one application and is executed in several different (sub)-transactions in distributed systems.

distributed transaction processing

Distributed processing with *distributed transactions.*

dynamic configuration

Changes to the *configuration* made by the administrator. UTM objects such as *program units*, *transaction codes*, *clients*, *LU6.1 connections,* printers or *user IDs* can be added, modified or in some cases deleted from the configuration while the application is running. To do this, it is necessary to create separate *administration programs* which use the functions of the *program interface for administration*. The WinAdmin administration program or the WebAdmin administration program can be used to do this, or separate *administration programs* must be created that utilize the functions of the *administration program interface.*

encryption level

        The encryption level specifies if and to what extent a client message and password are to be encrypted.

event-driven service

        This term has been superseded by *event service*.

event exit

        Routine in an application program which is started automatically whenever certain events occur (e.g. when a process is started, when a service is terminated). Unlike *event services*, an event exit must not contain any KDCS, CPI-C or XATMI calls.

event function

        Collective term for *event exits* and *event services*.

event service

        Service started when certain events occur, e.g. when certain UTM messages are issued. The *program units* for event-driven services must contain KDCS calls.

filebase

        UTM application filebase
        In BS2000 systems, filebase is the prefix for the *KDCFILE*, the *user log file* USLOG and the *system log file* SYSLOG.
        In Unix and Windows systems, filebase is the name of the directory under which the KDCFILE, the user log file USLOG, the system log file SYSLOG and other files relating to to the UTM application are stored.

generation

        Static configuration of a *UTM application* using the UTM tool KDCDEF and creation of an application program.

global secondary storage area

        See *secondary storage area.*

hardcopy mode

        Operating mode of a printer connected locally to a terminal. Any message which is displayed on screen will also be sent to the printer.

heterogeneous link

        In the case of *server-server communication:* a link between a *UTM application* and a non-UTM application, e.g. a CICS or TUXEDO application.

Highly Integrated System Complex / HIPLEX®

Product family for implementing an operating, load sharing and availability cluster made up of a number of BS2000 servers.

HIPLEX® MSCF

(MSCF = **M**ultiple **S**ystem **C**ontrol **F**acility)
Provides the infrastructure and basic functions for distributed applications with HIPLEX**®**.

homogeneous link

In the case of *server-server communication*: a link between two *UTM applications*. It is of no significance whether the applications are running on the same operating system platforms or on different platforms.

inbound conversation (CPI-C)

See *incoming conversation*.

incoming conversation (CPI-C)

A conversation in which the local CPI-C program is the *acceptor* is referred to as an incoming conversation. In the X/Open specification, the term "inbound conversation" is used synonymously with "incoming conversation".

initial KDCFILE

In a *UTM cluster application*, this is the *KDCFILE* generated by *KDCDEF* and which must be copied for each node application before the node applications are started.

initiator (CPI-C)

The communication partners in a *conversation* are referred to as the initiator and the *acceptor*. The initiator sets up the conversation with the CPI-C calls Initialize_Conversation and Allocate.

insert

Field in a message text in which openUTM enters current values.

inverse KDCDEF

A function which uses the dynamically adapted configuration data in the *KDCFILE* to generate control statements for a *KDCDEF* run. An inverse KDCDEF can be started "offline" under *KDCDEF* or "online" via the *program interface for administration*.

JDK

Java Development Kit
Standard development environment from Sun Microsystems for the development of Java applications.

job

Request for a *service* provided by a *UTM application*. The request is issued by specifying a transaction code. See also: *queued output job*, *dialog job*, *background job*, *job complex*.

job complex

Job complexes are used to assign *confirmation jobs* to *asynchronous jobs*. An asynchronous job within a job complex is referred to as a *basic job*.

job-receiving service (KDCS)

A job-receiving service is a *service* started by a *job-submitting service* of another server application.

job-submitting service (KDCS)

A job-submitting service is a *service* which requests another service from a different server application (*job-receiving service*) in order to process a job.

KDCADM

Standard administration program supplied with openUTM. KDCADM provides administration functions which are called with transaction codes (*administration commands*).

KDCDEF

UTM tool for the *generation* of *UTM applications*. KDCDEF uses the configuration information in the KDCDEF control statements to create the UTM objects *KDC-FILE* and the ROOT table sources for the main routine *KDCROOT*.
In UTM cluster applications, KDCDEF also creates the *cluster configuration file,* the *cluster user file*, the *cluster page pool*, the *cluster GSSB file* and the *cluster ULS file*.

KDCFILE

One or more files containing data required for a *UTM application* to run. The KDCFILE is created with the UTM generation tool *KDCDEF*. Among other things, it contains the *configuration* of the application.

KDCROOT

Main routine of an *application program* which forms the link between the *program units* and the UTM system code. KDCROOT is linked with the *program units* to form the *application program*.

KDCS message area

For KDCS calls: buffer area in which messages or data for openUTM or for the *program unit* are made available.

KDCS parameter area

See *parameter area.*

KDCS program interface

Universal UTM program interface compliant with the national DIN 66 265 standard and which includes some extensions. KDCS (compatible data communications interface) allows dialog services to be created, for instance, and permits the use of *message queuing* functions. In addition, KDCS provides calls for *distributed processing*.

Kerberos

Kerberos is a standardized network authentication protocol (RFC1510) based on encryption procedures in which no passwords are sent to the network in clear text.

Kerberos principal

Owner of a key.
Kerberos uses symmetrical encryption, i.e. all the keys are present at two locations, namely with the key owner (principal) and the KDC (Key Distribution Center).

key code

Code that represents specific access authorization or a specific role. Several key codes are grouped into a *key set*.

key set

Group of one or more *key codes* under a particular a name. A key set defines authorization within the framework of the authorization concept used (lock/key code concept or *access list* concept). A key set can be assigned to a *user ID*, an *LTERM partner* an (OSI) *LPAP partner*, a *service* or a *TAC queue*.

linkage program

See *KDCROOT.*

local secondary storage area

See *secondary storage area.*

Log4j

Log4j is part of the Apache Jakarta project. Log4j provides information for logging information (runtime information, trace records, etc.) and configuring the log output. *WS4UTM* uses the software product Log4j for trace and logging functionality.

lock code

Code protecting an LTERM partner or transaction code against unauthorized access. Access is only possible if the *key set* of the accesser contains the appropriate *key code* (lock/key code concept).

logging process

Process in Unix and Windows systems that controls the logging of account records or monitoring data.

LPAP bundle

LPAP bundles allow messages to be distributed to LPAP partners across several partner applications. If a UTM application has to exchange a very large number of messages with a partner application then load distribution may be improved by starting multiple instances of the partner application and distributing the messages across the individual instances. In an LPAP bundle, *open*UTM is responsible for distributing the messages to the partner application instances. An LPAP bundle consists of a master LPAP and multiple slave LPAPs. The slave LPAPs are assigned to the master LPAP on generation. LPAP bundles exist for both the OSI TP protocol and the LU6.1 protocol.

LPAP partner

In the case of *distributed processing* via the *LU6.1* protocol, an LPAP partner for each partner application must be configured in the local application. The LPAP partner represents the partner application in the local application. During communication, the partner application is addressed by the name of the assigned LPAP partner and not by the application name or address.

LTERM bundle

An LTERM bundle (connection bundle) consists of a master LTERM and multiple slave LTERMs. An LTERM bundle (connection bundle) allows you to distribute queued messages to a logical partner application evenly across multiple parallel connections.

LTERM group

An LTERM group consists of one or more alias LTERMs, the group LTERMs and a primary LTERM. In an LTERM group, you assign multiple LTERMs to a connection.

LTERM partner

LTERM partners must be configured in the application if you want to connect clients or printers to a *UTM application*. A client or printer can only be connected if an LTERM partner with the appropriate properties is assigned to it. This assignment is generally made in the *configuration*, but can also be made dynamically using terminal pools.

LTERM pool

The TPOOL statement allows you to define a pool of LTERM partners instead of issuing one LTERM and one PTERM statement for each *client*. If a client establishes a connection via an LTERM pool, an LTERM partner is assigned to it dynamically from the pool.

LU6.1

Device-independent data exchange protocol (industrial standard) for transaction-oriented *server-server communication*.

LU6.1-LPAP bundle

LPAP bundle for *LU6.1* partner applications.

LU6.1 partner

Partner of the *UTM application* that communicates with the UTM application via the *LU6.1* protocol.
Examples of this type of partner are:
– a UTM application that communicates via LU6.1
– an application in the IBM environment (e.g. CICS, IMS or TXSeries) that communicates via LU6.1

main process (Unix systems / Windows systems)

Process which starts the *UTM application*. It starts the *work processes*, the *UTM system processes*, *printer processes, network processes, logging process* and the *timer process* and monitors the *UTM application*.

main routine KDCROOT

See *KDCROOT*.

management unit

SE Servers component; in combination with the *SE Manager*, permits centralized, web-based management of all the units of an SE server.

mapped host name

Mapping of the partner application's UTM host name to a real host name or vice versa.

message definition file

The message definition file is supplied with openUTM and, by default, contains the UTM message texts in German and English together with the definitions of the message properties. Users can take this file as a basis for their own message modules.

message destination

Output medium for a *message*. Possible message destinations for a message from the openUTM transaction monitor include, for instance, terminals, *TS applications*, the *event service* MSGTAC, the *system log file* SYSLOG or *TAC queues, asynchronous TACs, USER queues,* SYSOUT/SYSLST or stderr/stdout.
The message destinations for the messages of the UTM tools are SYSOUT/SYSLST and stderr/stdout.

message queue

Queue in which specific messages are kept with transaction management until further processed. A distinction is drawn between *service-controlled queues* and *UTM-controlled queues*, depending on who monitors further processing.

message queuing

Message queuing (MQ) is a form of communication in which the messages are exchanged via intermediate queues rather than directly. The sender and recipient can be separated in space or time. The transfer of the message is independent of whether a network connection is available at the time or not. In openUTM there are *UTM-controlled queues* and *service-controlled queues*.

message router (BS2000 systems)

Device in a central host or a communication computer which distributes queued input messages to different *UTM applications* which can be located on different computers. The message router also allows you to work with *multiplex connections*.

MSGTAC

Special event service that processes messages with the message destination MSGTAC by means of a program. MSGTAC is an asynchronous service and is created by the operator of the application.

multiplex connection (BS2000 systems)

Special method of connecting terminals to a *UTM application*. A multiplex connection enables several terminals to share a single transport connection.

multi-step service (KDCS)

Service carried out in a number of *dialog steps*.

multi-step transaction

Transaction which comprises more than one *processing step*.

Network File System/Service / NFS

Allows Unix systems to access file systems across the network.

network process (Unix systems / Windows systems)

A process in a *UTM application* for connection to the network.

network selector

The network selector identifies a service access point to the network layer of the *OSI reference model* in the local system.

node

Individual computer of a *cluster*.

node application

UTM application that is executed on an individual *node* as part of a *UTM cluster application*.

node bound service

A node bound service belonging to a user can only be continued at the node application at which the user was last signed on. The following services are always node bound:
– Services that have started communications with a job receiver via LU6.1 or OSI TP and for which the job-receiving service has not yet been terminated
– Inserted services in a service stack
– Services that have completed a SESAM transaction
In addition, a user's service is node bound as long as the user is signed-on at a node application.

node filebase

Filename prefix or directory name for the *node application's KDCFILE*, *user log file* and *system log file*.

node recovery

If a node application terminates abnormally and no rapid warm start of the application is possible on its associated *node computer* then it is possible to perform a node recovery for this node on another node in the UTM cluster. In this way, it is possible to release locks resulting from the failed node application in order to prevent unnecessary impairments to the running *UTM cluster application*.

normal termination of a UTM application
Controlled termination of a *UTM application*. Among other things, this means that the administration data in the *KDCFILE* are updated. The *administrator* initiates normal termination (e.g. with KDCSHUT N). After a normal termination, openUTM carries out any subsequent start as a *cold start*.

object identifier
An object identifier is an identifier for objects in an OSI environment which is unique throughout the world. An object identifier comprises a sequence of integers which represent a path in a tree structure.

open terminal pool
Terminal pool which is not restricted to clients of a single computer or particular type. Any client for which no computer- or type-specific terminal pool has been generated can connect to this terminal pool.

online import
In a *UTM cluster application*, online import refers to the import of application data from a normally terminated node application into a running node application.

online update
In a *UTM cluster application,* online update refers to a change to the application configuration or the application program or the use of a new UTM revision level while a *UTM cluster application* is running.

OpenCPIC
Carrier system for UTM clients that use the *OSI TP* protocol.

OpenCPIC client
OSI TP partner application with the *OpenCPIC* carrier system.

openSM2
The openSM2 product line offers a consistent solution for the enterprise-wide performance management of server and storage systems. openSM2 offers the acquisition of monitoring data, online monitoring and offline evaluation.

openUTM application
See *UTM application*.

openUTM cluster
From the perspective of UPIC clients, **not** from the perspective of the server: Combination of several node applications of a UTM cluster application to form one logical application that is addressed via a common symbolic destination name.

openUTM-D

openUTM-D (openUTM distributed) is a component of openUTM which allows *distributed processing*. openUTM-D is an integral component of openUTM.

OSI-LPAP bundle

LPAP bundle for *OSI TP* partner applications.

OSI-LPAP partner

OSI-LPAP partners are the addresses of the *OSI TP partners* generated in openUTM. In the case of *distributed processing* via the *OSI TP* protocol, an OSI-LPAP partner for each partner application must be configured in the local application. The OSI-LPAP partner represents the partner application in the local application. During communication, the partner application is addressed by the name of the assigned OSI-LPAP partner and not by the application name or address.

OSI reference model

The OSI reference model provides a framework for standardizing communications in open systems. ISO, the International Organization for Standardization, described this model in the ISO IS7498 standard. The OSI reference model divides the necessary functions for system communication into seven logical layers. These layers have clearly defined interfaces to the neighboring layers.

OSI TP

Communication protocol for distributed transaction processing defined by ISO. OSI TP stands for Open System Interconnection Transaction Processing.

OSI TP partner

Partner of the UTM application that communicates with the UTM application via the OSI TP protocol.
Examples of such partners are:
– a UTM application that communicates via OSI TP
– an application in the IBM environment (e.g. CICS) that is connected via openUTM-LU62
– an application of the OpenCPIC carrier system of the openUTM client
– applications from other TP monitors that support OSI TP

outbound conversation (CPI-C)

See *outgoing conversation*.

outgoing conversation (CPI-C)

A conversation in which the local CPI-C program is the *initiator* is referred to as an outgoing conversation. In the X/Open specification, the term "outbound conversation" is used synonymously with "outgoing conversation".

page pool

Part of the *KDCFILE* in which user data is stored.

In a *standalone application* this data consists, for example, of *dialog messages*, messages sent to *message queues*, *secondary memory areas*.

In a UTM cluster application, it consists, for example, of messages to *message queues, TLS*.

parameter area

Data structure in which a program unit passes the operands required for a UTM call to openUTM.

partner application

Partner of a UTM application during *distributed processing*. Higher communication protocols are used for distributed processing (*LU6.1*, *OSI TP* or LU6.2 via the openUTM-LU62 gateway).

postselection (BS2000 systems)

Selection of logged UTM events from the SAT logging file which are to be evaluated. Selection is carried out using the SATUT tool.

prepare to commit (PTC)

Specific state of a distributed transaction

Although the end of the distributed transaction has been initiated, the system waits for the partner to confirm the end of the transaction.

preselection (BS2000 systems)

Definition of the UTM events which are to be logged for the *SAT audit*. Preselection is carried out with the UTM-SAT administration functions. A distinction is made between event-specific, user-specific and job-specific (TAC-specific) preselection.

presentation selector

The presentation selector identifies a service access point to the presentation layer of the *OSI reference model* in the local system.

primary storage area

Area in main memory to which the *KDCS program unit* has direct access, e.g. *standard primary working area*, *communication area*.

print administration

Functions for *print control* and the administration of *queued output jobs*, sent to a printer.

print control

openUTM functions for controlling print output.

printer control LTERM

A printer control LTERM allows a client or terminal user to connect to a UTM application. The printers assigned to the printer control LTERM can then be administered from the client program or the terminal. No administration rights are required for these functions.

printer control terminal

This term has been superseded by *printer control LTERM*.

printer group (Unix systems)

For each printer, a Unix system sets up one printer group by default that contains this one printer only. It is also possible to assign several printers to one printer group or to assign one printer to several different printer groups.

printer pool

Several printers assigned to the same *LTERM partner*.

printer process (Unix systems)

Process set up by the *main process* for outputting *asynchronous messages* to a *printer group*. The process exists as long as the printer group is connected to the *UTM application*. One printer process exists for each connected printer group.

process

The openUTM manuals use the term "process" as a collective term for processes (Unix systems / Windows systems) and tasks (BS2000 systems).

processing step

A processing step starts with the receipt of a *dialog message* sent to the *UTM application* by a *client* or another server application. The processing step ends either when a response is sent, thus also terminating the *dialog step*, or when a dialog message is sent to a third party.

program interface for administration

UTM program interface which helps users to create their own *administration programs*. Among other things, the program interface for administration provides functions for *dynamic configuration*, for modifying properties and application parameters and for querying information on the configuration and the current workload of the application.

program unit

UTM *services* are implemented in the form of one or more program units. The program units are components of the *application program*. Depending on the employed API, they may have to contain KDCS, XATMI or CPIC calls. They can be addressed using *transaction codes*. Several different transaction codes can be assigned to a single program unit.

queue

See *message queue*.

queued output job

Queued output jobs are *asynchronous jobs* which output a message, such as a document, to a printer, a terminal or a transport system application. Queued output jobs are processed by UTM system functions exclusively, i.e. it is not necessary to create program units to process them.

Quick Start Kit

A sample application supplied with openUTM (Windows systems).

redelivery

Repeated delivery of an *asynchronous message* that could not be processed correctly because, for example, the *transaction* was rolled back or the *asynchronous service* was terminated abnormally. The message is returned to the message queue and can then be read and/or processed again.

reentrant program

Program whose code is not altered when it runs. In BS2000 systems this constitutes a prerequisite for using *shared code*.

request

Request from a *client* or another server for a *service function*.

requestor

In XATMI, the term requestor refers to an application which calls a service.

resource manager

Resource managers (RMs) manage data resources. Database systems are examples of resource managers. openUTM, however, also provides its own resource managers for accessing message queues, local memory areas and logging files, for instance. Applications access RMs via special resource manager interfaces. In the case of database systems, this will generally be SQL and in the case of openUTM RMs, it is the KDCS interface.

restart

> See *screen restart*,
> see *service restart*.

RFC1006

> A protocol defined by the IETF (Internet Engineering Task Force) belonging to the TCP/IP family that implements the ISO transport services (transport class 0) based on TCP/IP.

RSA

> Abbreviation for the inventors of the RSA encryption method (Rivest, Shamir and Adleman). This method uses a pair of keys that consists of a public key and a private key. A message is encrypted using the public key, and this message can only be decrypted using the private key. The pair of RSA keys is created by the UTM application.

SAT audit (BS2000 systems)

> *Audit* carried out by the SAT (Security Audit Trail) component of the BS2000 software product SECOS.

screen restart

> If a *dialog service* is interrupted, openUTM again displays the *dialog message* of the last completed *transaction* on screen when the service restarts provided that the last transaction output a message on the screen.

SE manager

> Web-based graphical user interface (GUI) for the SE series of Business Servers. SE Manager runs on the *management unit* and permits the central operation and administration of server units (with /390 architecture and/or x86 architecture), application units (x86 architecture), net unit and peripherals.

SE server

> A Business Server from Fujitsu's SE series.

secondary storage area

> Memory area secured by transaction logging and which can be accessed by the KDCS *program unit* with special calls. Local secondary storage areas (LSSBs) are assigned to one *service*. Global secondary storage areas (GSSBs) can be accessed by all services in a *UTM application*. Other secondary storage areas include the *terminal-specific long-term storage (TLS)* and the *user-specific long-term storage (ULS)*.

selector

A selector identifies a service access point to services of one of the layers of the *OSI reference model* in the local system. Each selector is part of the address of the access point.

semaphore (Unix systems / Windows systems)

Unix systems and Windows systems resource used to control and synchronize processes.

server

A server is an *application* which provides *services*. The computer on which the applications are running is often also referred to as the server.

server-server communication

See *distributed processing*.

server side of a conversation (CPI-C)

This term has been superseded by *acceptor*.

service

Services process the *jobs* that are sent to a server application. A service of a UTM application comprises one or more transactions. The service is called with the *service TAC*. Services can be requested by *clients* or by other servers.

service access point

In the OSI reference model, a layer has access to the services of the layer below at the service access point. In the local system, the service access point is identified by a *selector*. During communication, the *UTM application* links up to a service access point. A connection is established between two service access points.

service chaining (KDCS)

When service chaining is used, a follow-on service is started without a *dialog message* specification after a *dialog service* has completed .

service-controlled queue

Message queue in which the calling and further processing of messages is controlled by *services*. A service must explicitly issue a KDCS call (DGET) to read the message. There are service-controlled queues in openUTM in the variants *USER queue*, *TAC queue* and *temporary queue*.

service restart (KDCS)
> If a service is interrupted, e.g. as a result of a terminal user signing off or a *UTM application* being terminated, openUTM carries out a *service restart*. An *asynchronous service* is restarted or execution is continued at the most recent *synchronization point*, and a *dialog service* continues execution at the most recent *synchronization point*. As far as the terminal user is concerned, the service restart for a dialog service appears as a *screen restart* provided that a dialog message was sent to the terminal user at the last synchronization point.

service routine
> See *program unit*.

service stacking (KDCS)
> A terminal user can interrupt a running *dialog service* and insert a new dialog service. When the inserted *service* has completed, the interrupted service continues.

service TAC (KDCS)
> Transaction code used to start a *service*.

session
> Communication relationship between two addressable units in the network via the SNA protocol *LU6.1*.

session selector
> The session selector identifies an *access point* in the local system to the services of the session layer of the *OSI reference model*.

shared code (BS2000 systems)
> Code which can be shared by several different processes.

shared memory
> Virtual memory area which can be accessed by several different processes simultaneously.

shared objects (Unix systems / Windows systems)
> Parts of the *application program* can be created as shared objects. These objects are linked to the application dynamically and can be replaced during live operation. Shared objects are defined with the KDCDEF statement SHARED-OBJECT.

sign-on check
> See *system access control*.

sign-on service (KDCS)
Special *dialog service* for a user in which *program units* control how a user signs on to a UTM application.

single-step service
Dialog service which encompasses precisely one *dialog step*.

single-step transaction
Transaction which encompasses precisely one *dialog step*.

SOA
(Service-Oriented Architecture)
SOA is a system architecture concept in which functions are implemented in the form of re-usable, technically independent, loosely coupled *services*. Services can be called independently of the underlying implementations via interfaces which may possess public and, consequently, trusted specifications. Service interaction is performed via a communication infrastructure made available for this purpose.

SOAP
SOAP (Simple Object Access Protocol) is a protocol used to exchange data between systems and run remote procedure calls. SOAP also makes use of the services provided by other standards, XML for the representation of the data and Internet transport and application layer protocols for message transfer.

socket connection
Transport system connection that uses the socket interface. The socket interface is a standard program interface for communication via TCP/IP.

standalone application
See *standalone UTM application*.

standalone UTM application
Traditional *UTM application* that is not part of a *UTM cluster application*.

standard primary working area (KDCS)
Area in main memory available to all KDCS *program units*. The contents of the area are either undefined or occupied with a fill character when the program unit starts execution.

start format
Format output to a terminal by openUTM when a user has successfully signed on to a *UTM application* (except after a *service restart* and during sign-on via the *sign-on service*).

static configuration

Definition of the *configuration* during generation using the UTM tool *KDCDEF*.

SYSLOG file

See *system log file*.

synchronization point, consistency point

The end of a *transaction*. At this time, all the changes made to the *application information* during the transaction are saved to prevent loss in the event of a crash and are made visible to others. Any locks set during the transaction are released.

system access control

A check carried out by openUTM to determine whether a certain *user ID* is authorized to work with the *UTM application*. The authorization check is not carried out if the UTM application was generated without user IDs.

system log file

File or file generation to which openUTM logs all UTM messages for which SYSLOG has been defined as the *message destination* during execution of a *UTM application.*

TAC

See *transaction code*.

TAC queue

Message queue generated explicitly by means of a KDCDEF statement. A TAC queue is a *service-controlled queue* that can be addressed from any service using the generated name.

temporary queue

Message queue created dynamically by means of a program that can be deleted again by means of a program (see *service-controlled queue*).

terminal-specific long-term storage (KDCS)

Secondary storage area assigned to an *LTERM, LPAP* or *OSI-PAP partner* and which is retained after the application has terminated.

time-driven job

Job which is buffered by openUTM in a *message queue* up to a specific time until it is sent to the recipient. The recipient can be an *asynchronous service* of the same application, a *TAC queue*, a partner application, a terminal or a printer. Time-driven jobs can only be issued by KDCS *program units*.

timer process (Unix systems / Windows systems)

Process which accepts jobs for controlling the time at which *work processes* are executed. It does this by entering them in a job list and releasing them for processing after a time period defined in the job list has elapsed.

TNS (Unix systems / Windows systems)

Abbreviation for the Transport Name Service. TNS assigns a transport selector and a transport system to an application name. The application can be reached through the transport system.

Tomcat

see *Apache Tomcat*

transaction

Processing section within a *service* for which adherence to the *ACID properties* is guaranteed. If, during the course of a transaction, changes are made to the *application information*, they are either made consistently and in their entirety or not at all (all-or-nothing rule). The end of the transaction forms a *synchronization point*.

transaction code/TAC

Name which can be used to identify a *program unit*. The transaction code is assigned to the program unit during *static* or *dynamic configuration*. It is also possible to assign more than one transaction code to a program unit.

transaction rate

Number of *transactions* successfully executed per unit of time.

transfer syntax

With *OSI TP,* the data to be transferred between two computer systems is converted from the local format into transfer syntax. Transfer syntax describes the data in a neutral format which can be interpreted by all the partners involved. An *Object Identifier* must be assigned to each transfer syntax.

transport selector

The transport selector identifies a service access point to the transport layer of the *OSI reference model* in the local system.

transport system application

Application which is based directly on a transport system interface (e.g. CMX, DCAM or socket). When transport system applications are connected, the partner type APPLI or SOCKET must be specified during *configuration*. A transport system application cannot be integrated in a *distributed transaction*.

TS application

See *transport system application.*

typed buffer (XATMI)

Buffer for exchanging typed and structured data between communication partners. Typed buffers ensure that the structure of the exchanged data is known to both partners implicitly.

UPIC

Carrier system for openUTM clients. UPIC stands for Universal Programming Interface for Communication.

UPIC Analyzer

Component used to analyze the UPIC communication recorded with *UPIC Capture*. This step is used to prepare the recording for playback using *UPIC Replay.*

UPIC Capture

Used to record communication between UPIC clients and UTM applications so that this can be replayed subsequently (*UPIC Replay*).

UPIC client

The designation for openUTM clients with the UPIC carrier system.

UPIC Replay

Component used to replay the UPIC communication recorded with *UPIC Capture* and prepared with *UPIC Analyzer*.

user exit

This term has been superseded by *event exit.*

user ID

Identifier for a user defined in the *configuration* for the *UTM application* (with an optional password for *system access control*) and to whom special data access rights (*system access control*) have been assigned. A terminal user must specify this ID (and any password which has been assigned) when signing on to the UTM application. In BS2000 systems, system access control is also possible via *Kerberos*.
For other clients, the specification of a user ID is optional, see also *connection user ID*.
UTM applications can also be generated without user IDs.

user log file

File or file generation to which users write variable-length records with the KDCS LPUT call. The data from the KB header of the *KDCS communication area* is prefixed to every record. The user log file is subject to transaction management by openUTM.

USER queue

Message queue made available to every user ID by openUTM. A USER queue is a *service-controlled queue* and is always assigned to the relevant user ID. You can restrict the access of other UTM users to your own USER queue.

user-specific long-term storage

Secondary storage area assigned to a *user ID*, a *session* or an *association* and which is retained after the application has terminated.

USLOG file

See *user log file.*

UTM application

A UTM application provides *services* which process jobs from *clients* or other applications. openUTM is responsible for transaction logging and for managing the communication and system resources. From a technical point of view, a UTM application is a process group which forms a logical server unit at runtime.

UTM cluster application

UTM application that has been generated for use on a cluster and that can be viewed logically as a **single** application.
In physical terms, a UTM cluster application is made up of several identically generated UTM applications running on the individual cluster *nodes*.

UTM cluster files

Blanket term for all the files that are required for the execution of a UTM cluster application. This includes the following files:
– *Cluster configuration file*
– *Cluster user file*
– Files belonging to the *cluster page pool*
– *Cluster GSSB file*
– *Cluster ULS file*
– Files belonging to the *cluster administration journal**
– *Cluster lock file**
– Lock file for start serialization* (only in Unix systems and Windows systems)
The files indicated by * are created when the first node application is started. All the other files are created on generation using KDCDEF.

UTM-controlled queue

Message queues in which the calling and further processing of messages is entirely under the control of openUTM. See also *asynchronous job, background job* and *asynchronous message*.

UTM-D

See *openUTM-D.*

UTM-F

UTM applications can be generated as UTM-F applications (UTM fast). In the case of UTM-F applications, input from and output to hard disk is avoided in order to increase performance. This affects input and output which *UTM-S* uses to save user data and transaction data. Only changes to the administration data are saved.

In UTM cluster applications that are generated as UTM-F applications (APPLI-MODE=FAST), application data that is valid throughout the cluster is also saved. In this case, GSSB and ULS data is treated in exactly the same way as in UTM cluster applications generated with UTM-S. However, service data relating to users with RESTART=YES is written only when the relevant user signs off and not at the end of each transaction.

UTM message

Messages are issued to *UTM message destinations* by the openUTM transaction monitor or by UTM tools (such as *KDCDEF*). A message comprises a message number and a message text, which can contain *inserts* with current values. Depending on the message destination, either the entire message is output or only certain parts of the message, such as the inserts).

UTM page

A UTM page is a unit of storage with a size of either 2K, 4K or 8 K. In *standalone UTM applications,* the size of a UTM page on generation of the UTM application can be set to 2K, 4K or 8 K. The size of a UTM page in a *UTM cluster application* is always 4K or 8 K. The *page pool* and the restart area for the KDCFILE and *UTM cluster files* are divided into units of the size of a UTM page.

utmpath (Unix systems / Windows systems)

The directory under which the openUTM components are installed is referred to as *utmpath* in this manual.

To ensure that openUTM runs correctly, the environment variable UTMPATH must be set to the value of *utmpath*. On Unix systems, you must set UTMPATH before a UTM application is started. On Windows systems, UTMPATH is set on installation.

UTM-S

In the case of UTM-S applications, openUTM saves all user data as well as the administration data beyond the end of an application and any system crash which may occur. In addition, UTM-S guarantees the security and consistency of the application data in the event of any malfunction. UTM applications are usually generated as UTM-S applications (UTM secure).

UTM SAT administration (BS2000 systems)

UTM-SAT administration functions control which UTM events relevant to security which occur during operation of a *UTM application* are to be logged by *SAT*. Special authorization is required for UTM-SAT administration.

UTM system process

UTM process that is started in addition to the processes specified via the start parameters and which only handles selected jobs. UTM system processes ensure that UTM applications continue to be reactive even under very high loads.

UTM terminal

This term has been superseded by *LTERM partner*.

virtual connection

Assignment of two communication partners.

warm start

Start of a *UTM-S* application after it has terminated abnormally. The *application information* is reset to the most recent consistent state. Interrupted *dialog services* are rolled back to the most recent *synchronization point*, allowing processing to be resumed in a consistent state from this point (*service restart*). Interrupted *asynchronous services* are rolled back and restarted or restarted at the most recent *synchronization point*.
For *UTM-F* applications, only configuration data which has been dynamically changed is rolled back to the most recent consistent state after a restart due to a preceding abnormal termination.
In UTM cluster applications, the global locks applied to GSSB and ULS on abnormal termination of this node application are released. In addition, users who were signed on at this node application when the abnormal termination occurred are signed off.

WebAdmin

Web-based tool for the administration of openUTM applications via a Web browser. WebAdmin includes not only the full function scope of the *administration program interface* but also additional functions.

Web service

Application which runs on a Web server and is (publicly) available via a standardized, programmable interface. Web services technology makes it possible to make UTM program units available for modern Web client applications independently of the programming language in which they were developed.

WinAdmin

Java-based tool for the administration of openUTM applications via a graphical user interface. WinAdmin includes not only the full function scope of the *administration program interface* but also additional functions.

work process (Unix systems / Windows systems)

A process within which the *services* of a *UTM application* run.

workload capture & replay

Family of programs used to simulate load situations; consisting of the main components *UPIC Capture*, *UPIC Analyzer* and *Upic Replay* (on Unix and Windows systems) the utility program *kdcsort*. Workload Capture & Replay can be used to record UPIC sessions with UTM applications, analyze these and then play them back with modified load parameters.

WS4UTM

WS4UTM (**W**eb**S**ervices for open**UTM**) provides you with a convenient way of making a service of a UTM application available as a Web service.

XATMI

XATMI (X/Open Application Transaction Manager Interface) is a program interface standardized by X/Open for program-program communication in open networks.
The XATMI interface implemented in openUTM complies with X/Open's XATMI CAE Specification. The interface is available in COBOL and C. In openUTM, XATMI can communicate via the OSI TP, *LU6.1* and UPIC protocols.

XHCS (BS2000 systems)

XHCS (Extended Host Code Support) is a BS2000 software product providing support for international character sets.

XML

XML (eXtensible Markup Language) is a metalanguage standardized by the W3C (WWW Consortium) in which the interchange formats for data and the associated information can be defined.

Abbreviations

Please note: Some of the abbreviations used here derive from the German acronyms used in the original German product(s).

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 - Loader - Starter (BS2000)
CCP	Communication Control Program
CCR	Commitment, Concurrency and Recovery
CCS	Coded Character Set
CCSN	Coded Character Set Name
CICS	Customer Information Control System
CID	Control Identification
CMX	Communication Manager in Unix Systems
COM	Component Object Model
CPI-C	Common Programming Interface for Communication
CRM	Communication Resource Manager
CRTE	Common Runtime Environment (BS2000)
DB	Database
DC	Data Communication
DCAM	Data Communication Access Method

DES	Data Encryption Standard
DLM	Distributed Lock Manager (BS2000)
DMS	Data Management System
DNS	Domain Name Service
DP	Distribted Processing
DSS	Terminal (Datensichtstation)
DTD	Document Type Definition
DTP	Distributed Transaction Processing
EBCDIC	Extended Binary-Coded Decimal Interchange Code
EJB	Enterprise JavaBeans^{TM}
FGG	File Generation Group
FHS	Format Handling System
FT	File Transfer
GSSB	Global Secondary Storage Area
HIPLEX$^{®}$	Highly Integrated System Complex (BS2000)
HLL	High-Level Language
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IFG	Interactive Format Generator
ILCS	Inter-Language Communication Services (BS2000)
IMS	Information Management System (IBM)
IPC	Inter-Process Communication
IRV	International Reference Version
ISO	International Organization for Standardization
Java EE	Java Platform, Enterprise Edition
JCA	Java EE Connector Architecture
JDK	Java Development Kit
KAA	KDCS Application Area
KB	Communication Area
KBPRG	KB Program Area
KDCADMI	KDC Administration Interface
KDCS	Compatible Data Communication Interface

KTA	KDCS Task Area
LAN	Local Area Network
LCF	Local Configuration File
LLM	Link and Load Module (BS2000)
LSSB	Local Secondary Storage Area
LU	Logical Unit
MQ	Message Queuing
MSCF	Multiple System Control Facility (BS2000)
NB	Message Area
NEA	Network Architecture for BS2000 Systems
NFS	Network File System/Service
NLS	Native Language Support
OLTP	Online Transaction Processing
OML	Object Module Library
OSI	Open System Interconnection
OSI TP	Open System Interconnection Transaction Processing
OSS	OSI Session Service
PCMX	Portable Communication Manager
PID	Process Identification
PIN	Personal Identification Number
PLU	Primary Logical Unit
PTC	Prepare to commit
RAV	Computer Center Accounting Procedure
RDF	Resource Definition File
RM	Resource Manager
RSA	Encryption algorithm according to Rivest, Shamir, Adleman
RSO	Remote SPOOL Output (BS2000)
RTS	Runtime System
SAT	Security Audit Trail (BS2000)
SECOS	Security Control System
SEM	SE Manager
SGML	Standard Generalized Markup Language
SLU	Secondary Logical Unit

SM2	Software Monitor 2
SNA	Systems Network Architecture
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SPAB	Standard Primary Working Area
SQL	Structured Query Language
SSB	Secondary Storage Area
SSO	Single Sign-On
TAC	Transaction Code
TCEP	Transport Connection End Point
TCP/IP	Transport Control Protocol / Internet Protocol
TIAM	Terminal Interactive Access Method
TLS	Terminal-Specific Long-Term Storage
TM	Transaction Manager
TNS	Transport Name Service
TP	Transaction Processing (Transaction Mode)
TPR	Privileged Function State in BS2000 (Task Privileged)
TPSU	Transaction Protocol Service User
TSAP	Transport Service Access Point
TSN	Task Sequence Number
TU	Non-Privileged Function State in BS2000 (Task User)
TX	Transaction Demarcation (X/Open)
UDDI	Universal Description, Discovery and Integration
UDS	Universal Database System
UDT	Unstructured Data Transfer
ULS	User-Specific Long-Term Storage
UPIC	Universal Programming Interface for Communication
USP	UTM Socket Protocol
UTM	Universal Transaction Monitor
UTM-D	UTM Variant for Distributed Processing in BS2000
UTM-F	UTM Fast Variant
UTM-S	UTM Secure Variant
UTM-XML	openUTM XML Interface

VGID	Service ID
VTSU	Virtual Terminal Support
WAN	Wide Area Network
WS4UTM	Web-Services for openUTM
WSDD	Web Service Deployment Descriptor
WSDL	Web Services Description Language
XA	X/Open Access Interface (X/Open interface for acess to the 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

Related publications

You will find the manuals on the internet at *http://manuals.ts.fujitsu.com*. You can order printed copies of those manuals which are displayed with an order number.

i PDF files of all openUTM manuals are included on the openUTM Enterprise DVD with open platforms and on the openUTM WinAdmin DVD (for BS2000 systems).

openUTM documentation

**openUTM
Concepts and Functions**
User Guide

**openUTM
Programming Applications with KDCS for COBOL, C and C++**
Core Manual

**openUTM
Generating Applications**
User Guide

**openUTM
Using openUTM Applications under BS2000 Systems**
User Guide

**openUTM
Using openUTM Applications under Unix Systems and Windows Systems**
User Guide

**openUTM
Administering Applications**
User Guide

**openUTM
Messages, Debugging and Diagnostics in BS2000 Systems**
User Guide

openUTM
Messages, Debugging and Diagnostics in Unix Systems and Windows Systems
User Guide

openUTM
Creating Applications with X/Open Interfaces
User Guide

openUTM
XML for openUTM

openUTM Client (Unix systems)
for the OpenCPIC Carrier System
Client-Server Communication with openUTM
User Guide

openUTM Client
for the UPIC Carrier System
Client-Server Communication with openUTM
User Guide

openUTM WinAdmin
Graphical Administration Workstation for openUTM
Description and online help system

openUTM WebAdmin
Web Interface for Administering openUTM
Description and online help system

openUTM, openUTM-LU62
Distributed Transaction Processing
between openUTM and CICS, IMS and LU6.2 Applications
User Guide

openUTM (BS2000)
Programming Applications with KDCS for Assembler
Supplement to Core Manual

openUTM (BS2000)
Programming Applications with KDCS for Fortran
Supplement to Core Manual

openUTM (BS2000)
Programming Applications with KDCS for Pascal-XT
Supplement to Core Manual

openUTM (BS2000)
Programming Applications with KDCS for PL/I
Supplement to Core Manual

WS4UTM (Unix systems and Windows systems**)**
WebServices for openUTM

openUTM
Master Index

Documentation for the openSEAS product environment

BeanConnect
User Guide

JConnect
Connecting Java Clients to openUTM
User documentation and Java docs

WebTransactions
Concepts and Functions

WebTransactions
Template Language

WebTransactions
Web Access to openUTM Applications via UPIC

WebTransactions
Web Access to MVS Applications

WebTransactions
Web Access to OSD Applications

Documentation for the BS2000 environment

AID
Advanced Interactive Debugger
Core Manual
User Guide

BCAM
BCAM Volume 1/2
User Guide

BINDER
User Guide

BS2000 OSD/BC
Executive Macros
User Guide

BS2000
BLSSERV
Dynamic Binder Loader / Starter
User Guide

DCAM
COBOL Calls
User Guide

DCAM
Macros
User Guide

DCAM
Program Interfaces
Description

FHS
Format Handling System for openUTM, TIAM, DCAM
User Guide

IFG for FHS
User Guide

HIPLEX AF
High-Availability of Applications in BS2000/OSD
Product Manual

HIPLEX MSCF
BS2000 Processor Networks
User Guide

IMON
Installation Monitor
User Guide

MT9750 (MS Windows)
9750 Emulation under Windows
Product Manual

OMNIS/OMNIS-MENU (BS2000)
Functions and Commands
User Guide

OMNIS/OMNIS-MENU (BS2000)
Administration and Programming
User Guide

OSS (BS2000)
OSI Session Service
User Guide

RSO
Remote SPOOL Output
User Guide

SECOS
Security Control System
User Guide

SECOS
Security Control System
Ready Reference

SESAM/SQL
Database Operation
User Guide

openSM2
Software Monitor
Volume 1: Administration and Operation

TIAM
User Guide

UDS/SQL
Database Operation
User Guide

Unicode in BS2000/OSD
Introduction

VTSU
Virtual Terminal Support
User Guide

XHCS
8-Bit Code and Unicode Support in BS2000/OSD
User Guide

Documentation for the Unix system environment

CMX V6.0 (Unix systems)
Betrieb und Administration (only available in German)
User Guide

CMX V6.0
Programming CMX Applications
Programming Guide

OSS (UNIX)
OSI Session Service
User Guide

PRIMECLUSTER*TM*
Concepts Guide (Solaris, Linux)

openSM2
The documentation of openSM2 is provided in the form of detailed online help systems, which are delivered with the product.

Other publications

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

XTAMI (X/Open)
Distributed Transaction Processing
X/Open CAE Specification
ISBN 1 85912 130 6

XML
W3C specification (www consortium)
Web page: *http://www.w3.org/XML*

Index