z/OS

IBM

SAP on DB2 UDB for OS/390 and z/OS: Planning Guide
2nd Edition
SAP Web Application Server 6.20

z/OS

IBM

SAP on DB2 UDB for OS/390 and z/OS: Planning Guide
2nd Edition
SAP Web Application Server 6.20

> **Note:**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page 247.

Second Edition, September 2004

This edition applies to

- SAP on DB2 UDB for OS/390 and z/OS
- SAP Web Application Server 6.20, including the 6.40 downward-compatible kernel in ICLI mode
- z/OS Version 1 Release 2 (5694-A01) and higher supported versions
- AIX Release 5.1 (5765-E61), and higher supported 5.x versions
- Linux for zSeries (for distribution details, see SAP Note 81737)
- IBM DB2 Universal Database for OS/390 (5645-DB2) Version 6 and DB2 Universal Database for OS/390 and z/OS (5675-DB2) Version 7
- Sun Solaris 8 (64–bit) and higher supported versions
- Windows 2000 and higher supported 32-bit versions

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for your comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany

FAX (Germany): 07031-16-3456
FAX (Other Countries): (+49) 7031-16-3456

Internet e-mail: `s390id@de.ibm.com`
World Wide Web:
`http://www.ibm.com/servers/eserver/zseries/software/sap`
`http://www.ibm.com/servers/eserver/zseries/zos`

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

About this document

This book describes all necessary planning and preparation steps that are specific to deploying *SAP systems* on DB2 UDB for OS/390 and z/OS. Additionally, it contains administration and diagnosis information for some of the components used. This document is therefore still needed after the installation of your SAP system is complete.

> **Caution:**
> This document applies specifically to the mySAP.com Technology component *SAP Web Application Server*. This means that, if not explicitly stated otherwise, all statements, values, etc., given in this document are related to the SAP Web Application Server kernel release and not to the SAP database release.
>
> Due to the projected replacement of ICLI with DB2 Connect as of SAP Web Application Server 6.40 in the framework of SAP NetWeaver '04, this is the last edition of the *Planning Guide* in its current form. For 6.40, an updated version will be available as a new SAP document entitled *Planning Guide: z/OS Configuration for SAP on DB2 UDB for z/OS*.

Who should read this document

This document is intended for administrators who need to plan and prepare the hardware and software environment for the SAP Web Application Server on DB2 UDB for OS/390 and z/OS.

Important remarks

SAP Web Application Server 6.20 requires z/OS Version 1 Release 2 or higher, and the information in this edition applies to use of either the 6.20 kernel or the downward-compatible 6.40 kernel in ICLI mode with this operating system. The document contains essential information on preparing a z/OS system to implement the SAP Web Application Server on DB2 UDB for OS/390 and z/OS.

Note: ICLI component names of the form "fome620..." have the designations "fome640..." when using the SAP 6.40 kernel. Unless otherwise noted, a reference to a 6.20 component is valid for 6.40 by replacing '620' with '640'.

Critical topics covered in this book are:
- configuration structure
- system security
- setup of z/OS
- installation, customizing, and maintenance of the ICLI client and server components
- performance topics
- service information
- printing

Chapter 10, "Planning the transition to DB2 Connect," on page 147 has been added to provide information on transitioning to SAP Web Application Server 6.40, which employs DB2 Connect instead of ICLI for basic DB2 connectivity.

Information included in earlier releases of this document concerning the setup of DB2 can now be found in the SAP documentation *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* and *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*.

References in this book to the *Connectivity Guide* implicitly apply as well to the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8026, which contains the latest connectivity information. Please consult both publications.

Information on hardware and software requirements, which was included in earlier releases of this document as Appendix A, has been deleted. Refer instead to the current SAP installation documentation *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* and *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*.

As of release 6.20, the ICLI Alert Router is no longer used for routing exception events. The functionality previously provided by the ICLI Alert Router is now incorporated in the SAP RFCOSCOL executable. RFCOSCOL also provides DB2 IFI performance data. For details on RFCOSCOL, refer to the *SAP Database Administration Guide* and the SAP installation guides, which are available at

http://service.sap.com/instguides

Before starting the SAP installation, read the respective sections and perform the required tasks described in Part 2, "Preparation," on page 21. This document is still needed after the installation is complete.

Conventions and terminology used in this document

In this document, the following naming conventions apply:
- IBM DB2 Universal Database for OS/390 and z/OS is referred to as DB2.
- The SAP system on DB2 UDB for OS/390 and z/OS is referred to as SAP on DB2.
- The term "UNIX" stands for AIX, Solaris, and z/OS UNIX System Services. "UNIX(-like)" or "UNIX(-style)" refers to UNIX and Linux.
- The term AIX encompasses the supported 64-bit versions of AIX 5.x.
- Linux for zSeries (64-bit) is referred to as Linux.
- The term "Windows" is used to encompass Windows 2000 and its supported 32-bit successors.
- The term "currently" refers to this document's edition date.
- The term *SAP installation tool* refers to the current SAP installation utility (see *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*).
- The IBM documentation *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition*, is referred to as the *Connectivity Guide*.
- The SAP documentation *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server* is referred to as the *SAP Database Administration Guide*. This is not to be confused with the IBM DB2 *Administration Guide* publication.

- The term *SAP installation guides* refers to the following SAP documentation:
  - *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*
  - *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*
- The term *SAP Planning Guide* refers to the SAP publication *Planning Guide: z/OS Configuration for SAP on IBM DB2 Universal Database for z/OS*, which applies to SAP releases beginning with SAP NetWeaver '04 and SAP Web Application Server 6.40.

Highlighting conventions

Italics are used for:
- document titles
- emphasis
- options, variables and parameters

Boldface is used for:
- check box labels
- choices in menus
- column headings
- entry fields
- field names in windows
- menu-bar choices
- menu names
- radio button names
- spin button names

`Monospace` is used for:
- coding examples
- commands and subcommands
- entered data
- file names
- group and user IDs
- message text
- path names

Underlined settings are:
- default values

Bold italics are used for:
- recommended values

Syntax diagrams

This document uses railroad syntax diagrams to illustrate how to use commands. This is how you read a syntax diagram:

A command or keyword that you must enter (a required command) is displayed like this:

```
►►──Command─────────────────────────────────────────────────────────►◄
```

An optional keyword is shown below the line, like this:

```
        ┌─Option─┐
►►──┴────────┴───────────────────────────────────────────►◄
```
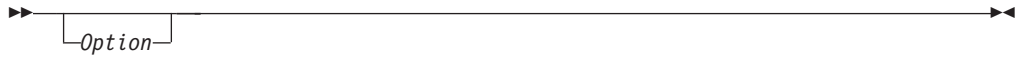
A default is shown over the line, like this:

```
        ┌─Default─┐
►►──┴─────────┴──────────────────────────────────────────►◄
```

An item that can be repeated (meaning that more than one optional keyword can be called) is shown like this:

```
        ┌─────────────┐
►►──▼──┴─Repeat─┴──────────────────────────────────────────►◄
```

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM® messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS® elements and features, z/VM®, VSE/ESA™, and Clusters for AIX® and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at

  http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/

- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX® System Services).

- Your Microsoft® Windows® workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).

- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Prerequisite and related information

SAP on DB2 uses a variety of different hardware and software systems. This document concentrates on information that goes beyond the standard knowledge needed for DB2 and SAP system administration. Therefore, it is assumed that you are familiar with:

- The z/OS environment (TSO, z/OS, UNIX System Services, RACF, JCL, RMF, WLM)
- DB2 administration (for example, SQL, SPUFI, and the utilities REORG and RUNSTATS)
- AIX, Linux for z/Series, Solaris, or Windows (or all)

Refer to "Bibliography" on page 241 for a list of related documentation.

How to send in your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document or any other z/OS documentation:

- Visit our home page at
  http://www.ibm.com/servers/eserver/zseries/software/sap
  Use the "Contact" field at the bottom of the page to send your comments.
- Send your comments by e-mail to s390id@de.ibm.com. Be sure to include the document's name and part number, the version of z/OS, and, if applicable, the specific location of the passage you are commenting on (for example, a page number or table number).
- Fill out one of the forms at the back of this document and return it by mail, by fax, or by giving it to an IBM representative.

Content of this document

This document describes the activities that need to be completed before the actual SAP installation via the SAP system installation tool can be started, and administrative tasks that may have to be performed repeatedly during the lifetime of the system. Chapter descriptions follow below:

"**Introduction**"

Provides general information and identifies related manuals.

Part 1, "Planning"

Provides you with input for your current and future planning activities for SAP systems on DB2.

Chapter 1, "Configuration structure"
> Focuses on configuration-related topics, for example, supported application server platforms and supported and recommended communication protocols and hardware. A number of sample configurations will give you an impression of the spectrum of possible system configurations.

Part 2, "Preparation"

Describes the tasks you have to complete before starting the actual SAP installation.

Chapter 2, "z/OS security, group IDs and user IDs"
Discusses security considerations and describes the group IDs and user IDs required on z/OS, AIX, Linux, Windows, Solaris, and DB2.

Chapter 3, "Configuring z/OS"
Describes z/OS configuration aspects. In particular, it concentrates on UNIX System Services parameters, TCP/IP, NFS setup, and DB2. A large section of the chapter is dedicated to the setup and configuration of the Workload Manager (WLM) for the ICLI server and the SAP application server on z/OS.

Chapter 4, "Setting up instances of the ICLI server"
Provides information on the Integrated Call Level Interface (ICLI), which enables application and database hosts (servers) to communicate remotely across a network. It describes the tasks you need to set up an ICLI server.

Chapter 5, "Performance tuning considerations before installing an SAP system"
Describes performance tuning steps you should complete before starting the actual SAP installation. Many of the tasks described here also apply during any stage of an SAP system's life cycle.

Part 3, "Administration"

Describes tasks you may have to perform repeatedly during the lifetime of your system.

Chapter 6, "Performance tuning considerations"
Covers a wide range of performance monitoring and tuning aspects. A large part of the chapter relates to the Workload Manager (WLM).

Chapter 7, "ICLI client and server"
Gives you a short overview of ICLI client-server architecture and provides the information needed to administer the ICLI client and server.

Chapter 8, "Applying service to ICLI and the SAP application server on z/OS"
Describes administrative tasks that need to be performed, for example, when a plan name other than the default is used or when a new SAP patch level or IBM ICLI PTF is installed on z/OS.

Chapter 9, "Using z/OS printing facilities to print SAP documents"
Describes the z/OS printing support.

Chapter 10, "Planning the transition to DB2 Connect"
Contains introductory information related to transitioning to DB2 Connect and DB2 Version 8. For more information, see the *SAP Planning Guide*, the SAP document *SAP on IBM DB2 UDB for OS/390 and z/OS: Best Practice for Installing or Migrating to DB2 V8*, and IBM documentation for DB2 Version 8 and DB2 Connect.

Appendix A, "Checklists guiding through this document"
Contains checklists that will guide you through the tasks you need to complete before you install the SAP system. It is recommended that you make a copy of the checklist for each SAP system you are going to set up, and that you fill it out while working through the document.

Appendix B, "Where to find further information on important tasks"
Lists important tasks in the SAP on DB2 for OS/390 and z/OS environment and refers to documentation describing these tasks.

Appendix C, "ICLI environment variables"
Describes ICLI environment variables.

Appendix D, "ICLI messages and codes"
Describes ICLI messages and codes.

List of abbreviations
Contains a list of important abbreviations appearing in this document.

Glossary
Explains the meaning of the most important technical terms employed in this document.

Bibliography
Contains lists of the IBM and SAP documentation referred to elsewhere in this document, as well as a list of selected SAP Notes.

Summary of changes

Summary of changes
for SC33–7959–02
SAP on DB2 UDB for OS/390 and z/OS
Planning Guide, 2nd Edition,
SAP Web Application Server 6.20

A vertical bar (|) in the left margin indicates major modifications and additions to the text with respect to the previous edition (SC33–7959–01).

New information

- Workload Management (WLM) ("Workload Management (WLM)" on page 36 and "Workload Management (WLM) — Classification of SAP workload" on page 82)
- Use of the SAP J2EE Engine ("File system setup" on page 49)
- Enhanced ASCII ("Enhanced ASCII feature overview" on page 50)
- Language Environment setup ("Language Environment setup" on page 40)
- Differences in ICLI APF authorization when using the 6.40 kernel ("ICLI client/server internals" on page 95)
- Transitioning to DB2 Connect and SAP Web Application Server 6.40 (Chapter 10, "Planning the transition to DB2 Connect," on page 147)
- Maintaining multiple ICLI versions at different maintenance levels ("Multiple ICLI versions on one system" on page 139)
- Using the LookAt message documentation facility ("Using LookAt to look up message explanations" on page xiv). ICLI messages are enabled for LookAt.
- List of applicable SAP Notes ("Bibliography" on page 241)

Information that has been removed or moved to other documentation:

- The checklists for hardware and software requirements, formerly included as Appendix A, have been deleted. Refer to current installation documentation for this information.
- The previously included information on ASCII/EBCDIC conversion has been replaced by a description of the Enhanced ASCII function required for J2EE (see "Enhanced ASCII feature overview" on page 50).
- The ″ICLI Alert Router″ chapter has been deleted.
- Information on Samba has been removed because it does not support file tagging for Enhanced ASCII.
- Information on high availability is now included in a new IBM publication entitled *High Availability for SAP on zSeries Using Autonomic Computing Technologies,* SC33–8206, available via the SAP on DB2 Web site:

  `http://www.ibm.com/servers/eserver/zseries/software/sap`

- Information on WLM compatibility mode has been deleted since it is no longer supported as of z/OS V1R3. Goal mode is recommended in any case and is implied in all WLM discussions.
- Information on ESCON has been deleted because it is no longer considered a state-of-the-art connectivity option.

- The description of remote command execution by **R3up** has been deleted because it applies only to SAP releases 4.5 and earlier.

Changed information

Major changes include:
- SMB setup ("SMB setup" on page 58)
- FTP transfer ("Transferring files with FTP" on page 62)

Deletions or relocations of text passages and changes to figures are not indicated.

Introduction

Before you can use the SAP installation package or the SAP migration kit to install, migrate, or copy SAP systems on DB2, you need to perform a number of planning and preparation steps. Some of these steps are described in this book, but you will need to consult some additional IBM and SAP documentation to complete all the required tasks.

The following figure and the list in Table 1 give you an overview of the books you will need to successfully install your SAP system.
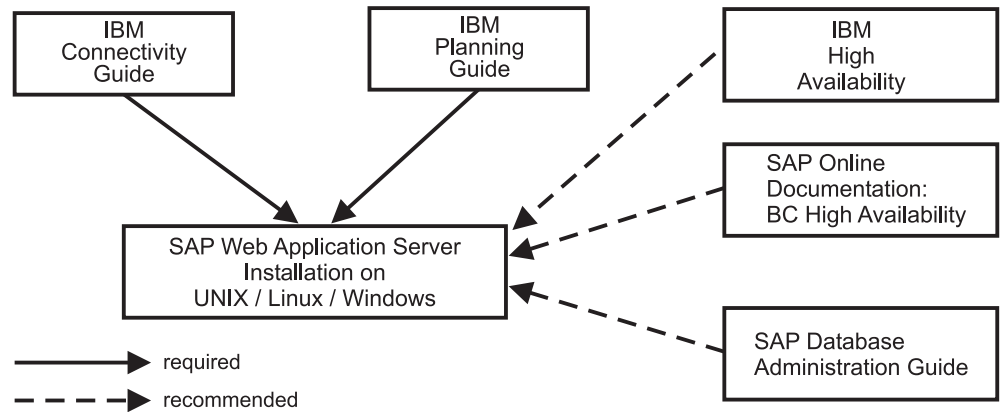
Figure 1. Steps and corresponding books in setting up an SAP system

Table 1. Required documentation

Documentation (for full titles, see "Bibliography" on page 241)	Purpose	Needed for
IBM documentation: *Connectivity Guide*	Describes supported connectivity options and SAP-specific required and recommended settings for communication setups, and parameters for SAP systems on DB2 for OS/390 and z/OS. **Note:** In each case, refer also the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, which contains the latest connectivity information.	Planning and preparation

Table 1. Required documentation (continued)

Documentation (for full titles, see "Bibliography" on page 241)	Purpose	Needed for
IBM documentation: *Planning Guide* (this book)	Describes most of the preparation and planning tasks that are not related to the communication setup and that you need to complete before an SAP system can be installed. The described procedures can be performed by the customers themselves and do not necessarily require the presence of an SAP certified consultant on site. The guide also contains information on the administration of some IBM software components and is still needed after the SAP installation is complete. The detailed checklists in Appendix A, "Checklists guiding through this document," on page 153 will guide you through the planning and preparation phases and the required documentation.	Planning, preparation, administration
SAP installation guides: *SAP Web Application Server Installation* on UNIX and *SAP Web Application Server Installation on Windows*	Describe some additional, required preparatory steps and the actual installation of the SAP Web Application Server. The books come on CD with the SAP installation package. The most up-to-date version of these guides is available in the SAP Service Marketplace at `http://service.sap.com/instguides`.	Preparation, installation
SAP documentation: *SAP Database Administration Guide*	Provides database-related information (for example, database tools, database performance considerations). It also provides performance-relevant information that you may want to consider in the planning phase. The most up-to-date version of this guide is available in the SAP Service Marketplace at `http://service.sap.com/instguides`.	Administration
IBM documentation: *High Availability for SAP on zSeries Using Autonomic Computing Technologies*	Describes the IBM SAP high availability solution, which provides for fully automating the management of all SAP components and related products running on z/OS, AIX, Windows, or Linux. The automation software monitors all resources and controls the restart and/or takeover of failing components, thereby ensuring near continuous availability of the SAP system. This book contains the latest connectivity information and should always be consulted in conjunction with the *Connectivity Guide*.	Planning, preparation
SAP Online Documentation: *BC SAP High Availability*	Helps you assess your current configuration and procedures and their implications on systems availability and offers recommendations in formulating a high systems availability strategy for your SAP systems. It is primarily intended to illustrate the available possibilities. You can find up-to-date information on high availability at: `http://service.sap.com/ha`	Planning, preparation

Table 1. Required documentation (continued)

Documentation (for full titles, see "Bibliography" on page 241)	Purpose	Needed for
SAP documentation: *SAP Web Application Server Heterogeneous System Copy*	Describes the migration from another platform to SAP Systems on DB2. The guide comes on CD with the SAP migration kit. The most up-to-date version of this guide is available in the SAP Service Marketplace at `http://service.sap.com/instguides`.	Planning, migration
SAP documentation: *SAP Web Application Server Homogeneous System Copy*	Describes copying an existing SAP on DB2. The guide comes on CD with the SAP migration kit. The most up-to-date version of this guide is available in the SAP Service Marketplace at `http://service.sap.com/instguides`.	Planning, copying

Part 1. Planning

Chapter 1. Configuration structure

This chapter describes the possible configuration structures of SAP on DB2.

The focus is on the application server and database server and the network located in between them.

Components

This section describes the components used on the application server and on the database server, and the components for their connection. Figure 2 shows these components for the simplest configuration.
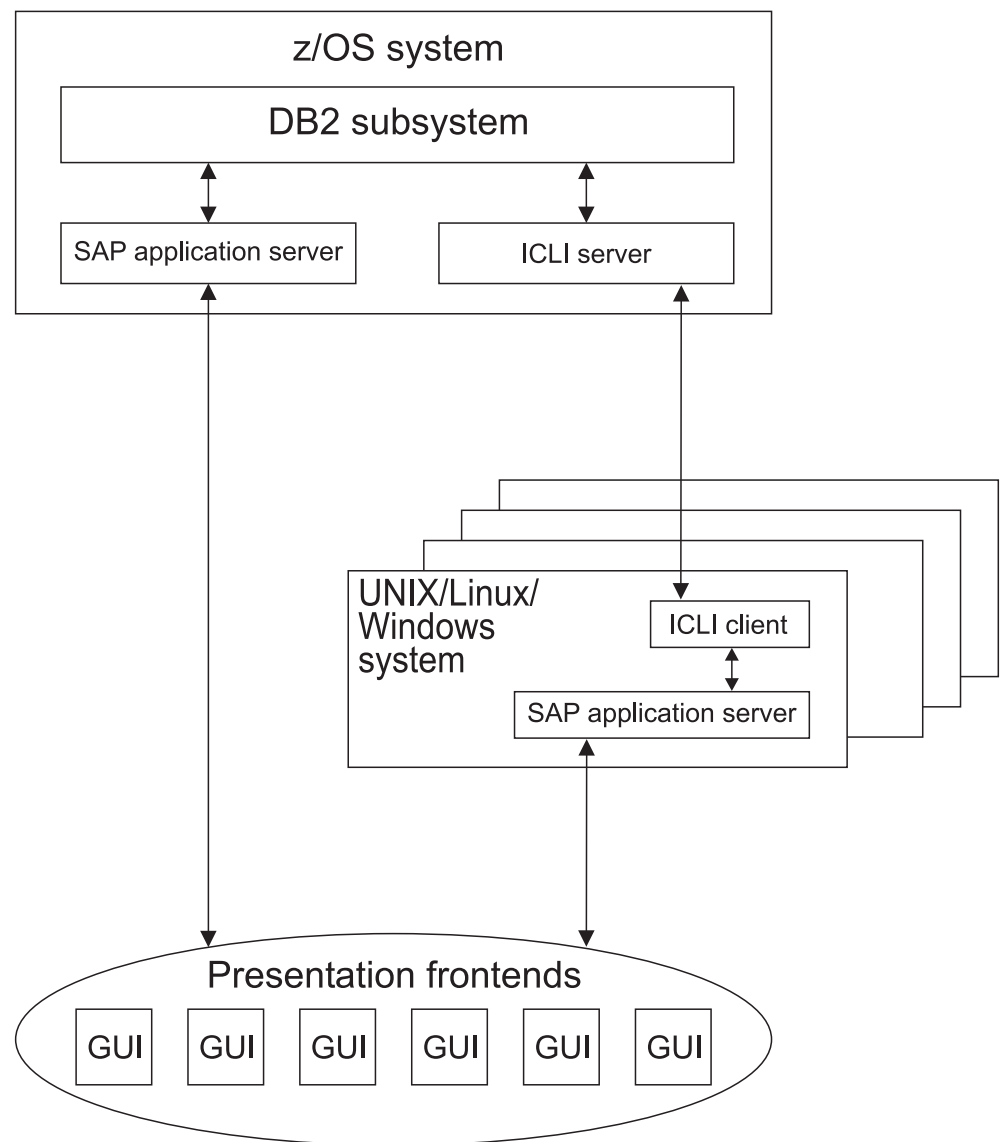
Figure 2. Components in the SAP configuration

DB2 subsystem and z/OS system

A DB2 subsystem is a formal z/OS subsystem that manages data structures (such as tablespaces, tables, and indexes) and system structures (such as buffer pools, directories, and logs). In order to increase availability and scalability, DB2 can also be run as a parallel database system. This mode is called *DB2 data sharing*, each subsystem is referred to as a *data sharing member*, and their set of subsystems is referred to as a *data sharing group*. SAP on DB2 supports DB2 data sharing without any restrictions. In the past, SAP installation only allowed one SAP System per DB2 subsystem or data sharing group. Today, SAP systems can be mapped to DB2 subsystems in a number of ways.

The following topologies are possible:
- a single DB2 subsystem (not involved in a data sharing group) manages only one SAP system.
- a data sharing group manages only one SAP system.
- a single DB2 subsystem (not involved in a data sharing group) manages multiple SAP systems.
- a DB2 data sharing group manages multiple SAP systems.

In SAP terminology, the last two topologies are referred to as MCOD (Multiple Components in One Database). (For detailed technical information on MCOD, refer to the SAP documentation available at `http://service.sap.com/mcod`.) Since each of the above configurations has its advantages and disadvantages, it is important to consider them carefully and assess their adequacy for a given scenario thoroughly.

An MCOD solution allows consistent backup and recovery across multiple SAP systems. For a more detailed discussion of MCOD and the possibility of point-in-time recovery of one system in an MCOD landscape, see *SAP on DB2 UDB for OS/390 and z/OS: Multiple Components in One Database (MCOD)*, SG24-6914.

One criterion that might speak for using MCOD is that it can reduce database administration costs. However, database administration efforts are mostly apparent during planning and installation and these are one-time exercises only. While the number of administrative tasks in everyday operations actually will grow as more DB2 subsystems are created, these tasks will not be more varied but rather very similar for all of the subsystems and can be easily automated and monitored.

MCOD in combination with *a single DB2 subsystem* imposes some limitations:
- Only a single set of DB2 installation parameters is available. For production systems this means that SAP systems that primarily show OLTP (online transaction processing) behavior (for example, SAP CRM) cannot reside in the same DB2 subsystem with SAP systems that primarily show OLAP (online analytical processing) behavior (for example, SAP BW).

  Note: This limitation is lifted with DB2 V8, which allows using a single set of parameters that ensures optimal performance for both OLTP and OLAP workload.
- Outages of the subsystem affect all SAP systems of an MCOD landscape.
- Large SAP systems with a high number of work processes may exceed the maximum number of active DB2 threads. In addition, shortages of virtual storage may be encountered as the virtual storage needs of the SAP systems add up.

For these reasons, MCOD with a single DB2 subsystem is a good choice for the consolidation of test or development systems with light load and modest performance requirements.

For *MCOD with DB2 data sharing*, these restrictions do not apply if an affinity between SAP systems and data sharing members is established (see also Figure 8 on page 16). In particular, such a topology particularly bridges the gap between OLTP-like SAP systems and OLAP-like SAP Systems. Note that this topology can be extended by assigning a dedicated subset of data sharing members to each SAP system, which enhances availability.

The storage and CPU overhead of additional DB2 subsystems should not be the crucial factor in the decision process, because it is not as high as customers usually expect. In fact, it is almost negligible compared to the resources used by the data belonging to an SAP system.

The same reasons that justify limiting one SAP system per DB2 subsystem or data sharing group should be applied to the decision of whether non-SAP data should be managed using the same subsystem or data sharing group that contains an SAP system. While there are no restrictions in that respect, it is recommended not to do so, especially in the case of critical applications.

Note that even without MCOD, one z/OS system can host multiple SAP systems. This is possible because many DB2 subsystems can run in the same z/OS system and each of these DB2 instances can manage different SAP systems. The same is valid for z/OS LPARs, which can also host multiple SAP systems. Each of these DB2 subsystems can use a different level or even a different release, if needed.

Application servers

In this book, the following terminology is used:

SAP application server
> The SAP infrastructure to run the SAP application, which includes
> - SAP dispatcher(s)
> - SAP work processes
> - SAP utilities and other SAP programs needed to run an SAP system

SAP application server instance
> A single SAP dispatcher process and its associated work processes, whose configuration is determined by a single SAP instance profile, and the SAP utilities and other SAP programs needed by this instance.

Application server host
> A hardware and operating system configuration on which an SAP application server can run.
>
> SAP on DB2 supports application servers on the following operating systems:
> - AIX 5.x (64–bit)
> - Linux for zSeries (64–bit)
> - z/OS UNIX System Services
> - Solaris (64-bit)
> - Windows (32-bit)

Application server hosts with all of these operating systems can be run in the same SAP system at the same time.

For the latest information about supported release combinations of these operating systems and possible restrictions, refer to the respective SAP Notes. These Notes are available at

http://service.sap.com/notes

SAP supports the operation of certified SAP systems only. For all operating systems listed above except Windows, the certification process is performed by the operating system vendor. For Windows application server models, AddOn Systemhaus GmbH is exclusively responsible for the certification. Check the AddOn Systemhaus GmbH Web site at

http://www.saponwin.com

and its link to certification for the latest information. In general, contact your IBM ERP Competence Center or IBM/SAP support group to obtain information on supported application server hardware.

If DB2 data sharing is used, each application server connects to only one database server at any one time. There is a mechanism called *sysplex failover* that automatically redirects an application server to another database server in case of a failure. For details, refer to the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*.

One or more SAP instances can run on each application server. Within each SAP instance, there are usually multiple SAP work processes.

There is no hard limit on the number of SAP application servers in an SAP System. Rather, it is implicitly limited by the number of work processes that could be serviced by a DB2 subsystem (up to 2000) or by a DB2 data sharing group (up to 32 * 2000).

ICLI client and server

ICLI stands for *Integrated Call Level Interface* and is part of z/OS's scope of delivery. It consists of a client component and a server component, which establish a remote SQL connection between SAP application servers on *non-z/OS operating systems* and z/OS database servers. Consequently, ICLI server and client components must be installed if non–z/OS SAP application servers are to communicate with DB2 on z/OS. ICLI servers reside on z/OS; ICLI client components are available for each operating system supported for SAP application servers. SAP application servers on z/OS do not use ICLI client and server components to access the database.

> **Note**
>
> ICLI will be replaced by DB2 Connect for database connectivity when the SAP Web Application Server starts using and exploiting DB2 V8 (see Chapter 10, "Planning the transition to DB2 Connect," on page 147).

Whether you need to set up one or several ICLI server instances depends on the following:
- Multiple SAP work processes on multiple SAP applications servers can use a single ICLI server. Each SAP work process uses one dedicated work thread within the ICLI server and one dedicated DB2 thread. The maximum number of

SAP work processes supported by a single ICLI server instance depends mainly on the maximum allowed storage for the address space of the ICLI server instances and on the utilization of z/OS shared storage areas. Consequently, more than one ICLI server is needed if the number of SAP work processes is higher than the maximum number of threads one ICLI server can create.

- Multiple ICLI server instances should also be used with the same DB2 subsystem if you want to remove a single point of failure and thereby increase the availability of your SAP system. We strongly recommend dedicating one ICLI server to one SAP system. This is especially important in the MCOD scenario where you have multiple SAP systems in one DB2.
- To meet high availability goals, it is recommended not to share ICLI servers.

Further configuration-related information on ICLI is presented in the following sections:

- For more information on the maximum number of threads that can be created by a single ICLI server instance, see "Configuring the ICLI server and SAP application server on z/OS" on page 40.
- For possible configurations with multiple ICLI servers, see "Sample configuration structures" on page 11.
- For a more detailed description of the ICLI client-server architecture, see "ICLI overview" on page 95.

DB2 and operating system prerequisites

Table 2 indicates the DB2 versions and operating system releases pertaining to specific SAP technology releases. This information is preliminary as long as the products are not yet released or the particular release combination is not yet supported by SAP. This information is therefore subject to change without further notice. The matrix reflects the current plans only and is not meant to state at which point in time the planned support of any of the items will actually be available. Check the SAP NetWeaver Product Availability Matrix to see which database and operating system combinations are currently certified:

```
http:/service.sap.com/pam
```

See also SAP Note 407325 and APAR II11352 for more information.

Table 2. Release matrix as currently available or planned

SAP Technology	DB2 for z/OS	z/OS	Linux for zSeries	AIX	Windows	Solaris
32/64–bit		32–bit	64–bit	64–bit	32–bit	64–bit
4.6	6.1+	OS/390 2.10+	SLES 7 SLES 8	5.1+ (32/64–bit)	NT4.0 Windows 2000+	8+
6.20	6.1+	1.2+	SLES 7 SLES 8	5.1+	Windows 2000+	8+
6.30	7.1+	1.4+	SLES 8	5.1+	Windows 2000+	8+
6.40	8.1	1.4+	SLES 8	5.1+	Windows 2000+	–

Connection between application servers and the database server

While no network connection needs to be established between an SAP application server on z/OS and the database server, a number of connectivity hardware options are available for connecting SAP application servers on UNIX/Linux and Windows application server hosts to the database server.

SAP on DB2 uses the TCP/IP communication protocol. For more information on supported connectivity options and SAP-specific required and recommended settings for communication setups and parameters, refer to the *Connectivity Guide* and also to *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206, which contains the latest connectivity information.

The following table shows the connectivity hardware supported by TCP/IP. It covers direct (non-gateway) configurations only.

Table 3. Connectivity hardware supported by TCP/IP

Application server type	Connectivity type
AIX 5.x (64–bit)	Any fast connectivity type supported by IBM (for example, Gigabit Ethernet, Fast Ethernet)
Linux for zSeries[1]	Any fast connectivity type supported by IBM (for example, HiperSockets, Gigabit Ethernet, Fast Ethernet)
Windows[2]	Any fast connectivity type supported by both IBM and the Windows server vendor (for example, Gigabit Ethernet, Fast Ethernet)
Solaris	Any fast connectivity type supported by both IBM and Sun (for example, Gigabit Ethernet, FDDI, Fast Ethernet)
Notes:	
[1] For configuration-related information, see the IBM documentation *Linux for zSeries and S/390 Device Drivers and Installation Commands* (available at `http://www10.software.ibm.com/developerworks/opensource/linux390/index.shtml` for Linux kernel 2.4) and the *Connectivity Guide*.	
[2] According to Windows certification rules of server model.	

Figure 3 on page 9 depicts the most prevalent connectivity types supported by TCP/IP:
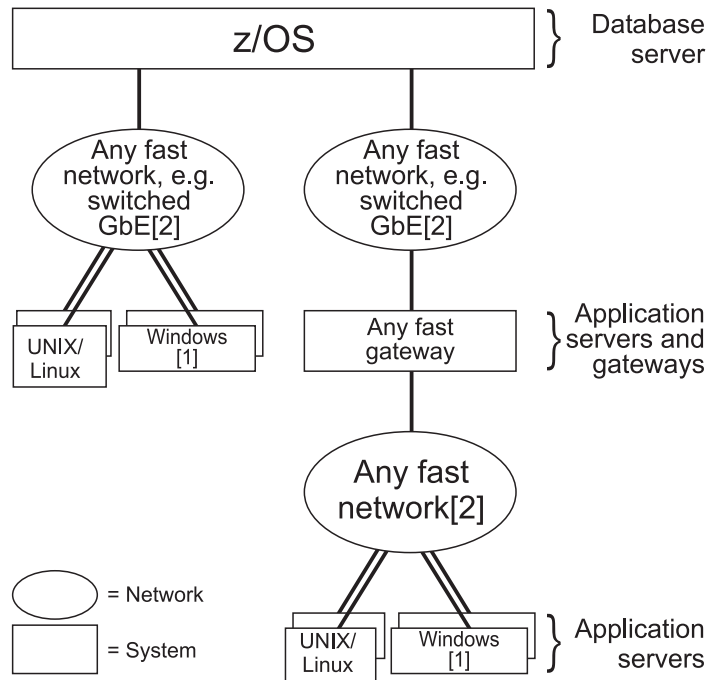
Figure 3. The most prevalent connectivity types supported by TCP/IP

SAP using TCP/IP supports both direct configurations and gateway configurations. The gateway, as well as the connection between the gateway and the application server, can be any connectivity type as long as it satisfies your performance needs.

The prime reason for using gateway (IP router) configurations is to share the connectivity between the gateway and z/OS among multiple application servers. However, gateway configurations tend to have a higher network turnaround time than direct configurations. This may cause longer elapsed times or response times with some workloads. For performance reasons, therefore, we do not recommend gateway configurations.

SAP on DB2 using TCP/IP supports any connectivity hardware that both the z/OS Communications Server and the TCP/IP product of the involved application servers and gateways support. Please make sure that your performance requirements and the performance capabilities of the connectivity hardware match. This includes (but is not limited to) OSA-Express Gigabit Ethernet, OSA-Express Fast Ethernet, OSA-Express ATM, ESCON, OSA-2 FDDI, OSA-2 Fast Ethernet, and OSA-2 ATM. All future connectivity hardware can be considered supported by SAP as soon as the above conditions are met.

Switched OSA-Express Gigabit Ethernet is strongly recommended in any environment where network performance is important. Gigabit Ethernet requires S/390 G5 or later hardware.

If the Linux for zSeries application server and SAP on DB2 database server run on the same zSeries server, HiperSockets is the preferred mode for TCP/IP LPAR-to-LPAR communication for an SAP Linux for zSeries application server in

such a setup, because of its superior performance characteristics as compared to all other modes of LPAR-to-LPAR communication. You can also communicate via a shared OSA-Express adapter, however.

Connectivity hardware

There are no simple recommendations regarding the type of connectivity hardware to use. Multiple factors (such as performance requirements, cost, availability requirements, and – last but not least – skills and network management tools) need to be considered.

For an estimate of the number of OSA-Express and/or ESCON and/or OSA-2 connections to use, contact your IBM ERP Competence Center or your IBM/SAP support group.

LAN connections (such as OSA-Express Gigabit Ethernet)

For the TCP/IP communications protocol, any sufficiently fast LAN connection is supported to z/OS.

> **Note:**
> Because of its performance characteristics, we particularly recommend using Gigabit Ethernet with the OSA-Express adapter.

A LAN connection consists of a physical LAN ring or segment, one or more OSA-2 or OSA-Express adapters on z/OS, a LAN adapter on each application server, and network control equipment to connect the adapters with the LAN ring or segment.

With LAN connections, network control equipment (such as a switch, hub, or concentrator) is highly recommended for best performance. Contact the manufacturer of your network control equipment to obtain detailed performance characteristics.

The choice of the number of separate LAN connections and how many application servers should connect to one LAN connection is independent of most other decisions. The main criteria are:
- Cost considerations
- Capacity considerations
- Availability considerations (that is providing alternative paths)

For details on the capacity of LAN connections in an SAP on DB2 system environment, refer to your IBM ERP Competence Center or IBM/SAP support group.

One LAN connection can be shared in many ways:
- by multiple communication protocols,
- by multiple ICLI server instances,
- by multiple DB2 subsystems, and
- by multiple SAP systems.

Considerations for the Linux for zSeries application server: For the Linux for zSeries application server, additional considerations apply if it runs in one LPAR and the SAP on DB2 database server runs in another LPAR within a single zSeries server. In such a setup, HiperSockets is the preferred mode for an SAP application

server for Linux for zSeries because of the superior performance characteristics of HiperSockets as compared to all other modes of LPAR-to-LPAR communication. It is also possible to set up connectivity between LPARs using a single shared OSA Express adapter. A shared OSA adapter provides inter-LPAR communication while eliminating the media transport latencies that are present when two non-shared adapters are used. Typically, the performance gain will only be observable as long as the shared adapter is not saturated. If your workload saturates the shared adapter, you should use two adapters as for all other platforms.

Connectivity recommendations

OSA-Express Gigabit Ethernet delivers significantly better network response times for workloads in an SAP on DB2 environment compared to other currently-available connectivity hardware. OSA-Express Gigabit Ethernet is strongly recommended in any environment where network performance is important.

For the Linux for zSeries application server, additional considerations apply if it runs in one LPAR and the SAP on DB2 database server runs in another LPAR within a single zSeries server. These are described in "Considerations for the Linux for zSeries application server" on page 10.

High availability considerations

An integrated high availability solution has been developed that combines the concepts of system automation, high availability, and transparent failover in a Parallel Sysplex. This solution covers all SAP components and related components that are running on z/OS. Critical and non-critical components are monitored and restarted automatically, if necessary, and alerts are posted.

Based on the IBM products *IBM Tivoli System Automation for z/OS* and *IBM Tivoli System Automation for Linux*, together with a redesign of the *SAP central instance concept*, this high availability solution exploits the new SAP standalone enqueue server, the enqueue replication server, dynamic virtual IP addresses (VIPA), shared HFS, and DB2 data sharing to guarantee a minimum of system outages along with a maximum of automation.

For comprehensive information on this automation and high availability solution, see the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.

Sample configuration structures

The five sample configuration structures presented in this section are meant to give you a feeling for the scalability and flexibility of SAP on DB2.

Economy configuration

Figure 4 on page 12 shows a case in which three small SAP systems – each running one application server – share as much network equipment as possible. This is why it has been dubbed an *economy configuration*.
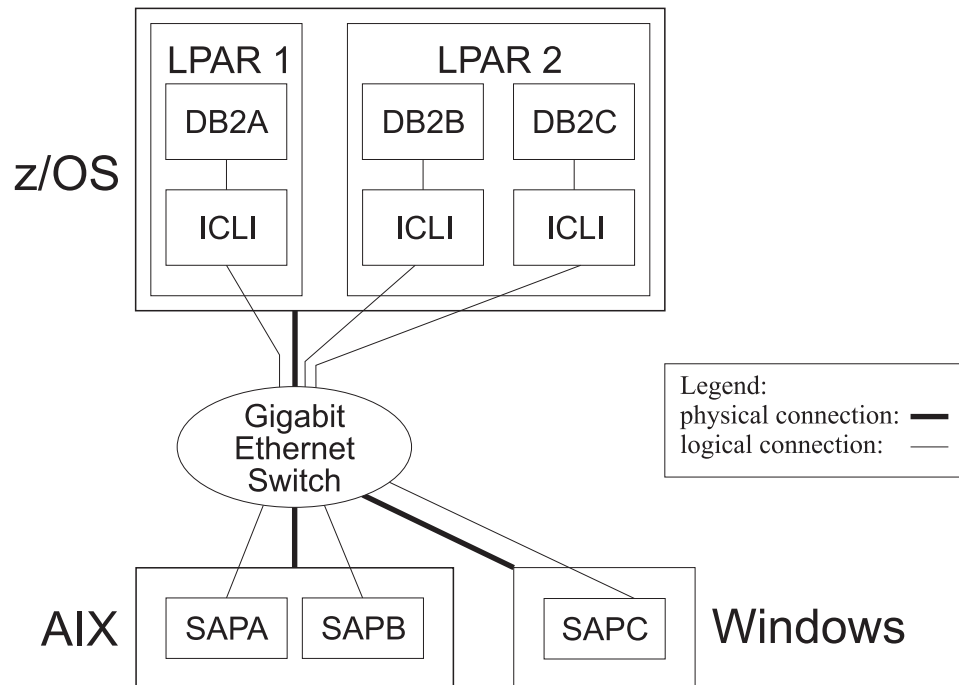
Figure 4. Economy configuration

In this case, an OSA-Express Gigabit Ethernet connection has been chosen.

The ICLI server instances are dedicated to SAP systems to facilitate administration. This allows an ICLI server instance to be stopped without affecting multiple SAP systems at the same time.

All three SAP systems share the same OSA-Express Feature via the switch, to which both the AIX and Windows systems are directly connected.

Linux for zSeries configuration

Figure 5 on page 13 depicts a zSeries environment in which Linux for zSeries application servers are running as guests in a z/VM environment and also in native logical partitions (LPARs).
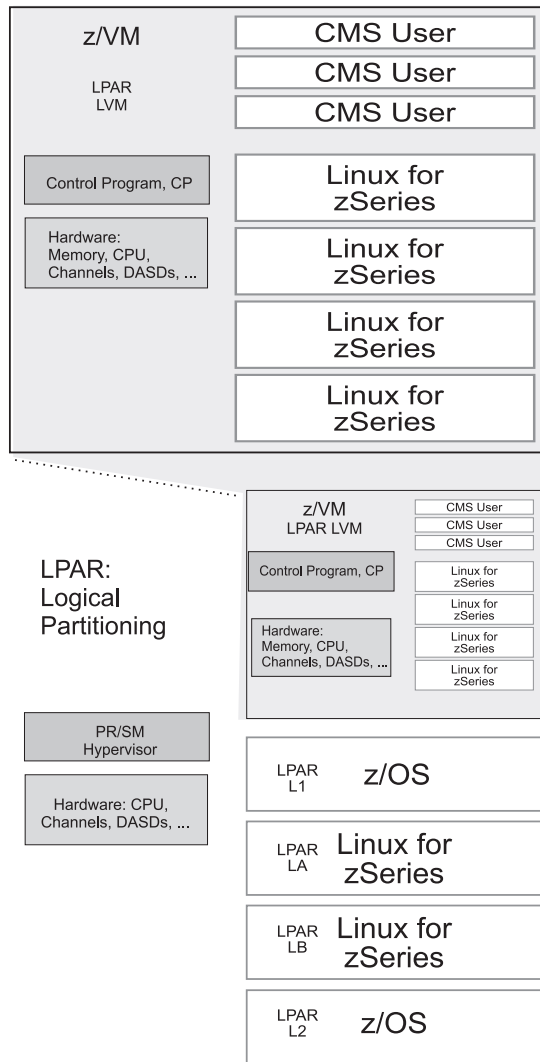
Figure 5. Linux for zSeries configuration in z/VM and LPAR mode

In the virtual machine model, all hardware resources of a computer system are managed by a hypervisor. VM's hypervisor is called the Control Program (CP). The zSeries hardware hypervisor is the Processor Resource/Systems Manager (PR/SM), which divides a machine into LPARs. The CPU and channel paths are the two resources that can be logically shared.

Figure 6 on page 14 illustrates a high-availability configuration of two SAP systems, SAP system A and SAP system B, implemented in the environment described above. Because all application servers reside on the same physical machine, they use HiperSockets to communicate with the corresponding z/OS database server. HiperSockets implement a very fast and highly available 'network in a box'. In such a configuration with two z/OS LPARs, we use DB2 data sharing to maximize availability of the DB server for each SAP system. For capacity reasons, the primary data sharing member for system B (DB2B1) and the scecondary (standby) data sharing member of system A (DB2A2) are running in LPAR L1, while the opposite is true for LPAR L2. Refer also to *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.
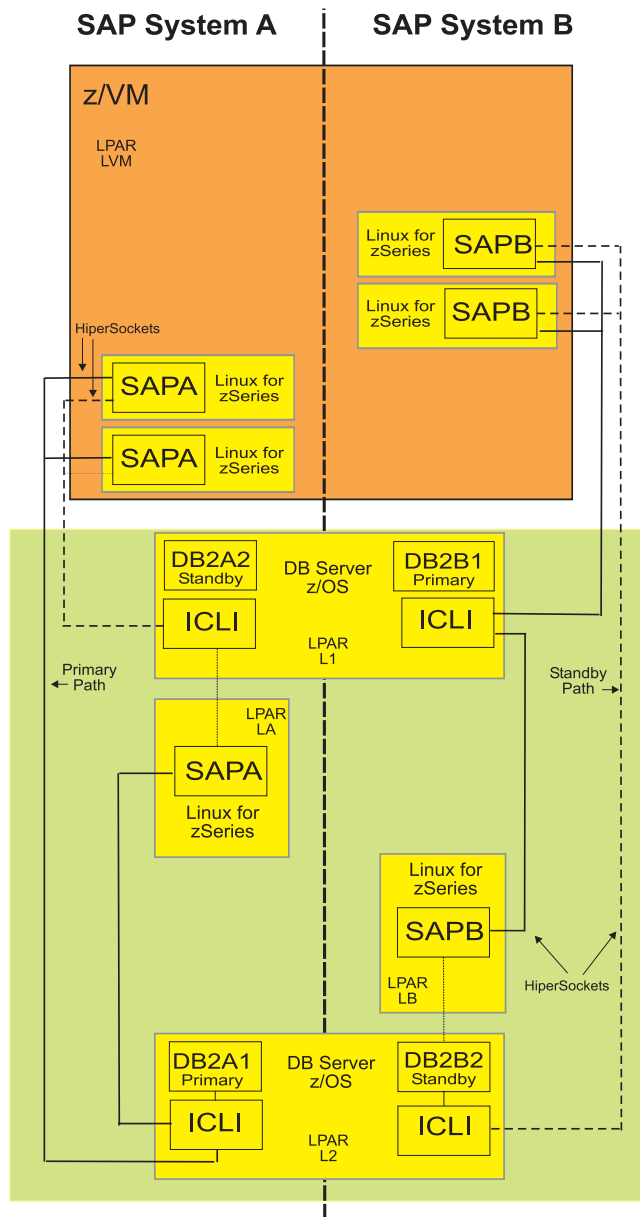
Figure 6. High availability configuration with z/OS and Linux for zSeries

Large configuration

Figure 7 on page 15 shows a large configuration exploiting DB2 data sharing. The term ″large configuration″ indicates that this SAP system has many application servers with several different parallel connections to different zSeries servers. Although not depicted in Figure 7 on page 15, the same configuration can be used if one or more SAP application servers reside on z/OS.
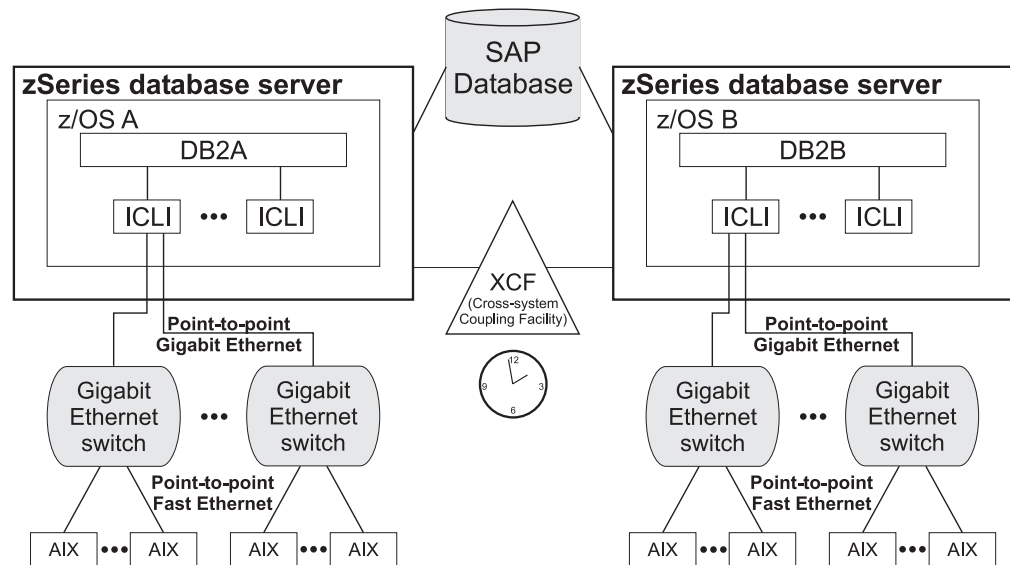
Figure 7. Large configuration

There is only one DB2 subsystem on each zSeries server because the main reason for using DB2 data sharing in this scenario is to expand the capacity of DB2 beyond that of a single zSeries server.

For capacity reasons and because of its superior latency, OSA-Express Gigabit Ethernet connections have been chosen. To achieve optimal latency, gateways have been avoided. And last but not least, the point-to-point configuration of the links improves throughput and latency. If the sharing of z/OS connectivity is not an issue, the switches can be omitted and a direct Gigabit Ethernet connection can be used instead. The optimal number of application servers per Gigabit Ethernet connection, and whether Fast Ethernet is sufficient for the particular kind of application server being used, should be determined as part of the SAP sizing.

More than one ICLI server instance per z/OS system is used because the number of SAP work processes on the SAP application servers attached to each z/OS system most likely exceed the maximum number of threads one ICLI server instance can create.

More than one dedicated OSA-Express Gigabit Ethernet connection is used per ICLI server instance because the capacity of a Gigabit Ethernet connection will probably be exceeded before the thread limit of the ICLI server instance is reached.

The sysplex failover support in the SAP system can be used to further improve availability. For details, refer to *SAP on zSeries: High Availability for SAP on zSeries Using Autonomic Computing Technologies.*

z/OS and AIX support duplicated Gigabit Ethernet adapters in order to bridge network outages as well. This has not been used in the scenario in order to keep it simple. For details, refer to *SAP on zSeries: High Availability for SAP on zSeries Using Autonomic Computing Technologies..*

MCOD configuration

Figure 8 on page 16 outlines a sample MCOD (Multiple Components in One Database) configuration. It encompasses two SAP systems (CRM, SBW) that require different DB2 settings and employ failover support. As opposed to the SAP CRM

system "CRM", the SAP BW system "SBW" requires DB2 parameters tailored to BW as described in the *SAP installation guides*. Most of the DB2 parameters (including the BW-relevant ones) can be set differently for the DB2 subsystems that form a data sharing group (note that DB2 V8 allows using a single set of parameters that ensures optimal performance for both OLTP and OLAP workload). Therefore, the basic idea of this topology is to have one data sharing group that

- serves both SAP systems *and*
- establishes an affinity between SAP systems and data sharing members.

This means that each member is connected to no more than one SAP system. The figure shows the four DB2 subsystems DB2A, DB2B, DB2C, DB2D, which form a data sharing group. During regular operations:

- CRM is connected to DB2A on LPAR1. In case of a failover, it will connect to DB2C on LPAR2.
- SBW is connected to DB2D on LPAR2 and fails over to DB2B on LPAR1 when necessary.

Thus, normally only one DB2 subsystem is busy per LPAR. While such a setup does help to reduce the total number of necessary LPARs, it has the disadvantage that both SAP systems will be connected to DB2 subsystems in the same LPAR when a failover occurs. For optimal performance and availability, you should consider running each data sharing member in a separate LPAR or zSeries server.
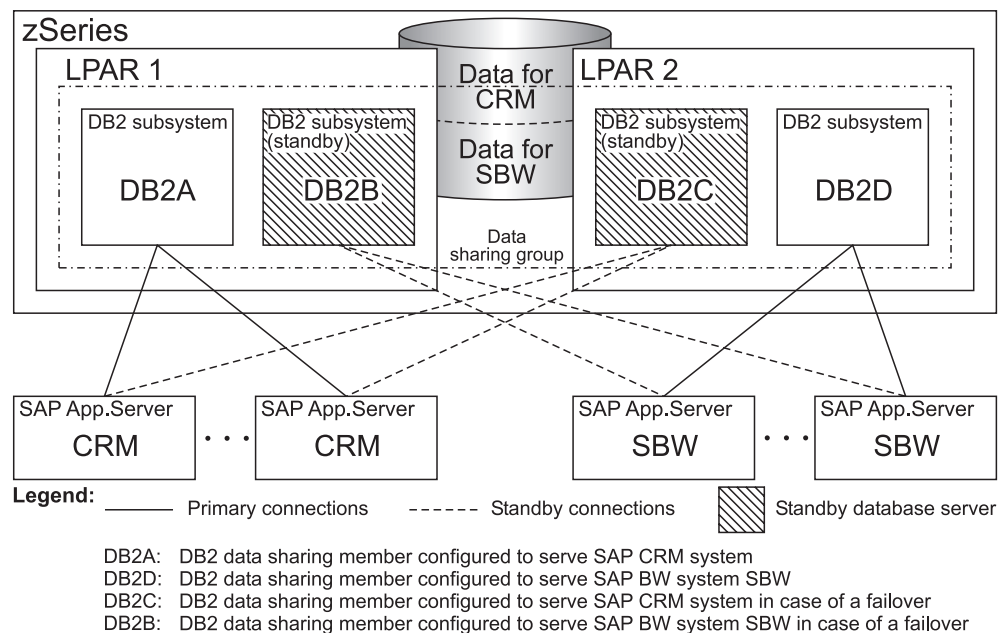
Figure 8. MCOD configuration

Configurations with SAP application servers on z/OS

In this section, two possible configurations with SAP application servers on z/OS are depicted and their corresponding advantages and disadvantages are described.

Heterogeneous configuration
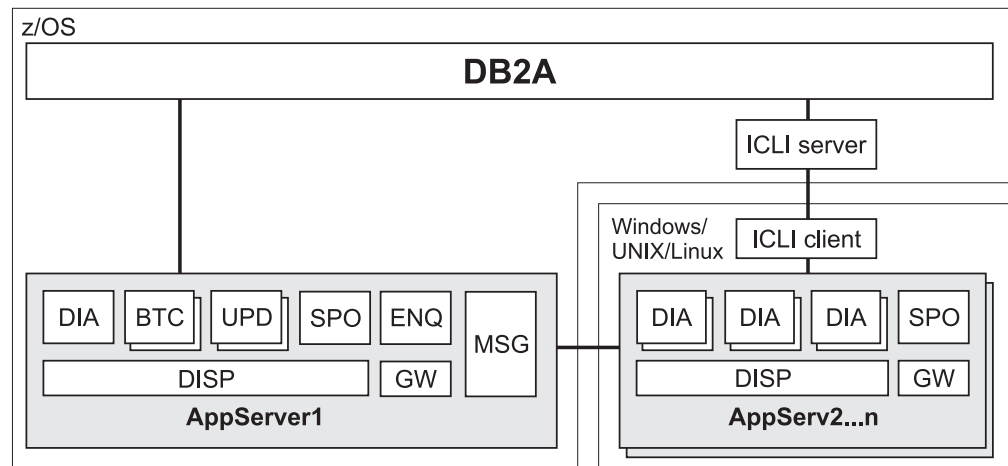
LEGEND:
AppServ1	Application server instance running on z/OS; Central Instance
AppServ2...n	Application server instances running on other platforms
DISP	Dispatcher
DIA	Dialog work process
BTC	Batch work process
UPD	Update work process
SPO	Spool work process
ENQ	Enqueue server
MSG	Message server
GW	Gateway server

Figure 9. Heterogeneous configuration

Figure 9 shows the configuration in a heterogeneous environment. The central instance **AppServ1** is running on z/OS in the same LPAR as DB2. Dialog instances (**AppServ2...n**) on other platforms (AIX, Linux for zSeries, Solaris, Windows) are connected to the database via a network. In the case of the dialog instances, the ICLI is used as a communication interface to transmit database requests between SAP work processes and DB2. **AppServ1** does not need the ICLI because the database is called directly.

The following factors were taken into consideration in designing this sample configuration:
- availability
- type of workload
- integration
- number of instances in one LPAR.

Availability: z/OS was chosen as the central instance's location for high availability reasons. The central instance includes a message server and an enqueue server. Usually, the SAP file system's shared directories reside on the same system and are mounted by the dialog instances.

If availability of the entire SAP system is critical, refer to *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.

Type of workload: Due to the proximity of application servers on z/OS and the database server, the largest performance improvements are seen for a workload that is database-intensive. This type of workload is typical for UPDATE and for many BATCH jobs.

On the other hand, the performance of CPU-bound workload is determined mainly by the processor speed. Proximity to the database server is not so performance-relevant. This is typical for DIALOG work processes that perform a lot of input checking and interaction with the graphical user interface.

At least one DIALOG process should be started on each instance. This allows the administrator to log in. It is also required for some RFC communication between SAP instances.

ALE or BAPI processing is performed in a DIALOG work process as well; however, it usually has characteristics similar to those of BATCH workload.

Integration: For integration with other non-SAP programs (e.g., using BAPI = business application program interface), it might be desirable to run the external program and an application server instance on the same system. This facilitates administration and provides a performance advantage if a huge amount of data is exchanged.

However, if the external program is not supported on the chosen application server platforms, it can still connect to an SAP application server. The communication is always effected via the SAP gateway, and it is always TCP/IP-based.

An SAP system can have several SPOOL work processes. For high volume printing, especially when printing to a host printer, one or more SPOOL work processes should be started by the SAP application server instance. Since the SAP system gets the spool data from the database, any network traffic for printing is avoided.

Number of instances in one LPAR: Running multiple application server instances on z/OS against the same database will not improve performance. On the other hand, as long as it does not cause paging, it will also not decrease performance. More real storage is required because, in this case, SAP buffer pools are allocated multiple times. SAP buffer pools can be shared only among work processes belonging to the same instance.

Sysplex configuration
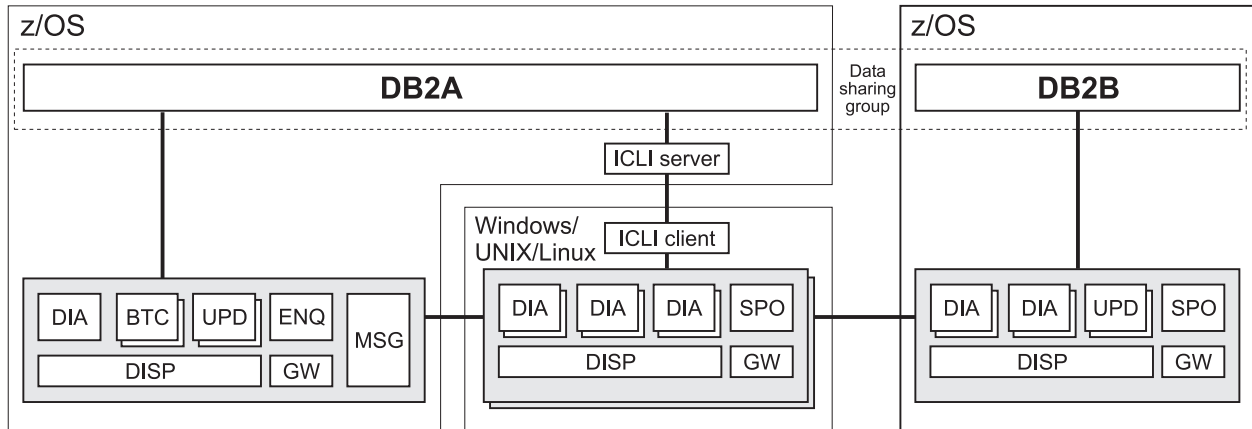
Figure 10. Sysplex configuration

Figure 10 shows a zSeries Parallel Sysplex configuration (see Figure 9 on page 17 for the legend). The central instance is located in one z/OS LPAR. In a second LPAR, a distributed application server instance is running DIALOG, UPDATE, and SPOOL work processes. By exploiting the DB2 data sharing feature, it is possible to add zSeries systems to a sysplex, thereby increasing the capacity of application server instances on z/OS. In such a configuration, too, the discussed considerations and performance advantages apply.

For sysplex configurations that support high availability of the SAP system, see also *High Availablity for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206 . Additional information can be found at the SAP on DB2 Web site under:

`http://www.ibm.com/servers/eserver/zseries/software/sap/sapesa/zsap-scenarios.html`

Part 2. Preparation

Chapter 2. z/OS security, group IDs and user IDs

This chapter addresses topics related to the security of your SAP on DB2 system environment regarding user IDs and authorizations on z/OS and DB2, and to physical security.

The user IDs and group IDs mentioned in the following sections contain variable parts. The variable text that you need to replace with the valid value for your installation is enclosed in angle brackets (<>). The following variables are used:

Table 4. System-specific variables used in this chapter

Variable	Description
`<SCHEMA>`	The DB2 schema used by the SAP system you plan to install. In a non-MCOD environment, the schema name usually is SAPR3. Refer to the *SAP installation guides* for details on choosing the DB2 schema.
`<SAPSID>` `<sapsid>`	The SAP system identifier (in uppercase or lowercase letters) used for the SAP system you plan to install. Refer to the *SAP installation guides* for details on choosing the DB2 SAP system identifier. SAP requires that you use the name `<sapsid>adm` for the user ID that the SAP application server runs under. If you are using a separate user ID to run the ICLI server, we recommend using `<SAPSID>ICLI`, although you are free to choose any z/OS user ID that is suitable for your installation.
`<sapsid>adm`	The user ID used to run the SAP application server.
`<SAPSID>ICLI`	The user ID used to run the ICLI server.

Overview

In the SAP client/server environment, you need to consider security from the presentation servers to the database server. This includes network, AIX, Linux, Solaris, Windows, SAP system, z/OS, and DB2 security:

- Each AIX, Linux, or Solaris system that runs SAP servers (application, update, message, and so on) has UNIX (-style) IDs and files that must be secured. To run SAP, only the user ID `<sapsid>adm` is needed. When SAP is installed on UNIX or Linux, this user ID is created by the SAP installation tool and does not need to be created during the preparation.
- The SAP system has a security system that is used to grant user IDs access to transactions, data, and resources (such as printers).
- z/OS Security Server (RACF) or a similar security product protects resources and authorizes users for z/OS and Unix System Services. Several user IDs need to be created (for details, refer to "z/OS group IDs and user IDs" on page 24).
- DB2 controls the access to database resources, such as tables and views. All transactions of a single SAP system use the same DB2 schema. An SAP application will change its SQLID to this DB2 schema when it accesses the database. Therefore, no SAP user will require a user ID on DB2.
- Physical security must be ensured in such a way that nobody can tamper with the z/OS, AIX, Linux, Solaris, or Windows systems, and the connections

between them. The ICLI server uses a RACF PassTicket compatible service to secure the connection establishment from an ICLI client. Once a secure connection is established, the ensuing communication between the database server (ICLI server) and the application servers (ICLI clients) is unencoded.

UNIX and Linux security

UNIX (-style) security (AIX, Linux, Solaris, z/OS UNIX System Services) is implemented using UNIX(-style) user IDs (UIDs) and group IDs (GIDs). The UIDs and GIDs are numbers. The names of the groups and users are symbols which stand for the current GID and UID numbers. If two user names share the same UID number, they are the *same user* to a UNIX(-like) system. Keep the following in mind when adding or checking users and UIDs or GIDs:

- The UID number used by the <sapsid>adm user must not be assigned to any other user ID.
- The SAPSYS group must contain only SAP <sapsid>adm user IDs.
- The GID number used by the SAPSYS group must not be assigned to any other group.

In a UNIX(-like) environment, if you have several SAP application servers on different UNIX(-like) systems which share data (for example /usr/sap/trans), it is important for NFS that the UID and GID numbers match on all systems. That is, <sapsid>adm must have the same UID on all systems, and SAPSYS must have the same GID on all systems. This can be implemented manually on each system, or by using the NIS (Network Information System) shared passwd and group files.

On z/OS, users and groups are defined by using the z/OS Security Server (RACF). The UNIX(-style) attributes of a user or group are specified in the OMVS segment. NIS is not supported on z/OS.

z/OS security

This section describes the security setup using the z/OS Security Server (RACF). If you use a different security product, the following sections will give you an idea of the kind of security you must set up before using your product.

z/OS group IDs and user IDs

The following z/OS group IDs and user IDs need to be created as part of the preparation:

Group ID <SCHEMA>

A z/OS group ID <SCHEMA> must exist on z/OS, since <SCHEMA> is the creator of all SAP objects in the DB2 subsystem. The ICLI server sets its current SQLID to <SCHEMA>. The group ID <SCHEMA> must be defined as a DB2 secondary authorization ID of both <SAPSID>ICLI and <sapsid>adm. This can be done after the DB2 subsystem setup is complete. To define DB2 secondary authorization IDs, you have to change the default sign-on exit routine. This is described in the *SAP installation guides*.

The group ID <SCHEMA> needs several DB2 privileges. Use the JCL job FOME620G (FOME640G) described in "DB2 authorization for the ICLI server and SAP application server on z/OS" on page 134 to grant these privileges. The DB2 subsystem setup must be complete before the job can be run.

Note:
In an MCOD environment, you have to ensure that *each* group ID <SCHEMA> is defined as a DB2 secondary authorization ID of <SAPSID>ICLI and <sapsid>adm, respectively.

Group ID SAPSYS to run the SAP application server on z/OS

The group ID SAPSYS is a z/OS UNIX System Services group ID that must be defined to run the SAP application server on z/OS. You define the group to RACF with an OMVS segment specifying the group ID (GID). In a heterogeneous environment with SAP systems on other UNIX platforms, the GID must be the same on all systems. This is important for NFS.

User ID to install an SAP application server on z/OS

To install an SAP application server on z/OS, superuser authorization is needed. Use this user ID whenever the SAP installation guide refers to the user ID "root".

Define a user ID with the following attributes:

- The RACF Userid Profile should have an OMVS segment with a UID of 0 and its DEFAULT GROUP set to SAPSYS.
  Partial example: ALU (root-user) DFLTGRP(SAPSYS) OMVS(UID(0)) ...

- If the profile BPX.DAEMON of RACF's FACILITY class is defined on your system, allow READ access to BPX.DAEMON and READ access to BPX.DAEMON.HFSCTL.
  Example: PE BPX.DAEMON CLASS(FACILITY) ACCESS(READ) ID(root-user)

- READ access to the profile BPX.SUPERUSER of RACF's FACILITY class.
  Example: PE BPX.SUPERUSER CLASS(FACILITY) ACCESS(READ) ID(root-user)

- READ access to the profile BPX.FILEATTR.PROGCTL of RACF's FACILITY class.
  Example: PE BPX.FILEATTR.PROGCTL CLASS(FACILITY) ACCESS(READ) ID(root-user)

- READ access to profile IRR.RADMIN of RACF's FACILITY class. If the profile did not previously exist, define it first. This profile authorizes the user to verify RACF settings. The SAP installation tool attempts to do such a RACF verification and will fail with RACF return code 8 and reason code 24 if this profile is not set up correctly.
  Example: PE IRR.RADMIN CLASS(FACILITY) ACCESS(READ) ID(root-user)

- READ access to the DB2 data set you specify as STEPLIB, that is, the load library <HLQ>.SDSNLOAD.
  Example: PE '<HLQ>.SDSNLOAD' ACCESS(READ) ID(root-user)

- Set the SPECIAL or at least the AUDITOR attribute for the user ID:
  - Both attributes enable the SAP installation tool to verify the security setup of the installation user ID itself and that of <sapsid>adm.
  - If the attribute SPECIAL is defined, the SAP installation tool creates the user ID <sapsid>adm, if necessary, and checks that the access permissions to the RACF profiles are set as described in this section, correcting them if necessary.
    Example: ALU (root-user) SPECIAL
  - If the attribute AUDITOR is defined, the SAP installation tool verifies that the user ID <sapsid>adm exists and that the access permissions to RACF profiles are set as described in this section. If <sapsid>adm does not exist or the access permissions are insufficient, the installation utility generates error messages; the errors have to be corrected by the administrator.
    Example: ALU (root-user) AUDITOR

Make sure that this user ID is also permitted to allocate up to 2 GB of storage. If using **rlogin**, the parameter *MAXASSIZE* applies. If using the z/OS UNIX shell via TSO log-on, the TSO *REGION* size must be set to 2 GB. In particular, log on to this

user ID to run the SAP installation tool; do not use the **su** command to switch to *UID=0* because this will not grant you the required RACF authorization.

For further information on these security profiles, refer to chapter "Defining Superuser Attributes for Users" in *z/OS UNIX System Services Planning*.

User ID <sapsid>adm to run the SAP application server on z/OS

The user ID <sapsid>adm — where <sapsid> is the SAP system identifier — is a z/OS UNIX System Services user ID. This user ID is needed to run the application server on z/OS. You can leave the creation of the user ID, its home directory and all RACF definitions to the SAP installation tool. The automatic creation requires that you have given the SPECIAL attribute to the installation user ID. The SAP installation tool does not define a TSO segment for this user ID, but you can add it manually later on (see also "User ID to install an SAP application server on z/OS" on page 25).

When creating the user ID manually, define the following attributes:
- RACF definition with the OMVS segment specifying the UID and GID; as the GID, use the number you have specified for SAPSYS. In a heterogeneous environment with SAP application server instances on other UNIX platforms, the UID must not be 0 and the UID and GID must be the same on all systems. This is important for NFS.
- The home directory is /u/<sapsid>adm. The home directory may be shared with application servers on other systems (for example, z/OS, UNIX).
- As the startup program, specify the C-shell: /bin/csh (see also "The C-shell, tcsh, and Korn shell" on page 63).
- Optionally, specify maximum values for *ASSIZEMAX*, *CPUTIMEMAX*, *PROCUSERMAX*, *FILEPROCMAX*, and *THREADSMAX*. This is necessary if the system-wide settings are left at a lower value but the settings are increased for individual users. See "Specifying limits for individual users" on page 43.
- If the <sapsid>adm user enters the z/OS UNIX shell via TSO log-on, the TSO region size must be set to 2 GB and the TIME parameter of the TSO logon procedure must be set to NOLIMIT.
- Since a user ID defined in RACF is always given in uppercase, but UNIX expects the user ID in lowercase, you must define a lowercase alias for the user ID in the file that was specified as USERIDALIASTABLE in parmlib member BPXPRMxx. See "Selecting the UNIX System Services parameters" on page 41. If the file does not exist, create it. The following sample entry was created in USERIDALIASTABLE for the user ID <sapsid>adm of an SAP system with the system identifier C11:

```
C11ADM    c11adm
```

> **Note:**
> The table is column-oriented. The uppercase user ID must start in column 1 and the lowercase alias in column 10. The in-storage copy of the table is regularly refreshed by the system. Using the OMVS operator command **SET OMVS** allows an immediate activation of the changes to that table.

You can verify the definition using the command **id <sapsid>adm**. The user name should now appear in lowercase characters.
- For the z/OS UNIX environment, the default language setting, *LANG=C*, and the default code page, IBM-1047, are to be used; the default is used if the environment variable *LC_ALL* is not set.

- READ, WRITE, EXECUTE (**rwx**) permissions to its home directory
  /u/<sapsid>adm, to /usr/sap/<SAPSID>, to /sapmnt/<SAPSID>, and to all of their
  subdirectories.
- READ access to the profile BPX.MAP of RACF's FACILITY class.
  Example: PE BPX.MAP CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>adm)
- READ access to the profile BPX.WLMSERVER of RACF's FACILITY class: WLM
  registration of the SAP application server can be enabled by setting the
  appropriate SAP profile parameter.
  Example: PE BPX.WLMSERVER CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>adm)
- READ access to the profile SUPERUSER.FILESYS.PFSCTL of RACF's UNIXPRIV class.
  Example: PE SUPERUSER.FILESYS.PFSCTL CLASS(UNIXPRIV) ACCESS(READ)
  ID(<SAPSID>adm)
- READ access to the profile ERBSDS.SMFDATA of RACF's FACILITY class. This
  allows the operating system monitor SAPOSCOL to read the SMF records
  written by RMF.
  Example: PE ERBSDS.SMFDATA CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>adm)
- Define the profile BPX.SAFFASTPATH of RACF's FACILITY class. For performance
  reasons, this profile is strongly recommended. For further information, see
  "Establishing the FastPath support for system authorization facility (SAF)" in
  z/OS UNIX System Services Planning.
  Example: RDEFINE FACILITY BPX.SAFFASTPATH UACC(NONE)
- If the application server is started as a started task, the user <sapsid>adm and
  group SAPSYS must be associated with the started task. This can be achieved by
  using RACF's STARTED class.

> **Using <sapsid>adm as the ICLI server user ID:**
> In order to keep the administration of user IDs simple, you can use
> <sapsid>adm for both the SAP application server and ICLI server. In this case,
> you need to add the following settings for <sapsid>adm:
>
> - If the ICLI server is started as a started task for user ID <sapsid>adm,
>   associate this user ID (and *not* <SAPSID>ICLI) with the started task. This can
>   be achieved by using RACF's STARTED class.
> - Give <sapsid>adm READ access to the profile BPX.STOR.SWAP of RACF's
>   FACILITY class (see also "Ensuring optimal ICLI server performance" on
>   page 72).

User ID to run the ICLI server

You have to install at least one ICLI server on z/OS if you want to run SAP
application servers on non-z/OS operating systems. The user ID (<SAPSID>ICLI)
needed to run an ICLI server must have special authorizations that are described
in the following paragraphs.

If you want to run multiple ICLI server instances, you have the following options:
- They can all use the same user ID. This is the preferred option. You can still
  identify individual ICLI servers if you use a different plan name for each. You
  can also indirectly define performance goals for individual ICLI servers by
  specifying these goals for each application server that is connected to that ICLI
  server. See Table 7 on page 84 for the list of available WLM work qualifiers.
- You can create different user IDs for some or all ICLI servers. However, this may
  have negative side effects, because it increases EDM Pool storage usage and
  decreases performance through global cache hit ratio degradation. If you want to
  avoid these negative effects, consider the following alternatives:

– If you intend to use different ICLI server user IDs just to be able to identify the various instances more easily, consider using different plan names for the instances instead.

– If your reason for using different user IDs is that you want to define explicit performance goals for different ICLI servers, consider using an implicit classification via application server names instead.

For more information on virtual storage considerations and dynamic statement caching, refer to the SAP documentation *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*.

The user ID <SAPSID>ICLI is a z/OS UNIX System Services user ID and, consequently, must have a user ID (UID), group ID (GID), and a home directory:

- You should *not* grant <SAPSID>ICLI UNIX System Services superuser (*UID*=0) privileges, because then any RACF security settings you made for <SAPSID>ICLI would be overridden and all security checks on the BPX profiles would always be passed. This means especially that, irrespective of the attributes you set for the profile BPX.WLMSERVER of RACF's FACILITY class, the ICLI server would always run according to the WLM classifications for SAP workload and all WLM settings would have to be made (see also "Workload Management (WLM) — Classification of SAP workload" on page 82).

- For GID, set <SAPSID>ICLI's default RACF GROUP to <SCHEMA> (see "Group ID <SCHEMA>" on page 24).

- Give <SAPSID>ICLI R, W, X permissions on its home directory.

If the ICLI server is started using a started task:

- <SAPSID>ICLI needs to be associated with the started task. This can be achieved by using RACF's STARTED class.

- The user ID employed in starting the procedure must have EXECUTE permission on the JCL data set. Note that this user ID does not need to be <SAPSID>ICLI.

In addition, it is highly recommended that you give <SAPSID>ICLI the following attributes:

- READ access to the profile for daemon administration, BPX.DAEMON, of RACF's FACILITY class (or your equivalent security product). This will prevent z/OS log-on. If the profile BPX.DAEMON is not defined on your system, you can define it in the FACILITY class with no universal access: UACC(NONE). The complete process for defining security for a daemon process such as the ICLI server is described in *z/OS UNIX System Services Planning*.

- READ access to the profile BPX.WLMSERVER of RACF's FACILITY class (or your equivalent security product) to run the ICLI server attached to WLM. For a detailed description of how to run ICLI servers attached to WLM, see "Workload Management (WLM) — Classification of SAP workload" on page 82.
  Example: PE BPX.WLMSERVER CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>ICLI)).

- READ access to the profile BPX.STOR.SWAP of RACF's FACILITY class. (for more information, see "Ensuring optimal ICLI server performance" on page 72). This will allow the ICLI server to make its address space non-swappable and to benefit from the resulting performance improvement.
  Example: PE BPX.STOR.SWAP CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>ICLI)

- READ access to the profile ERBSDS.SMFDATA of RACF's FACILITY class if you intend to run the operating system monitor SAPOSCOL under <SAPSID>ICLI. This allows SAPOSCOL to read the SMF records written by RMF.
  Example: PE ERBSDS.SMFDATA CLASS(FACILITY) ACCESS(READ) ID(<SAPSID>ICLI)

User ID for selected SAP system installation tasks

During the SAP system installation, a user ID for job submission and for FTP access to z/OS is needed. The user ID is needed for remote SAP application servers as well as for SAP application servers on z/OS. This user ID must have the following attributes:

- READ permission to the ICLI client executables (`FOME*` located in `SYS1.SFOMDATA`).
  Example: `PE 'SYS1.SFOMDATA' ID(<SAPSID>ICLI) ACCESS(READ)`
- Authority to submit JCL jobs that invoke DB2 utilities such as RUNSTATS
- Authority to create and read data sets under its own High Level Qualifier (HLQ) on the default volume
- DB2 SYSADM authority

After the DB2 subsystem setup, the JCL job `FOME620G` (`FOME640G`) described in "DB2 authorization for the ICLI server and SAP application server on z/OS" on page 134 is used to grant DB2 privileges to <SCHEMA> and <SAPSID>ICLI. The JCL job `FOME620B` (`FOME640B`) described in "DB2 binding for ICLI and the SAP application server on z/OS" on page 135 needs to be run for each DB2 subsystem whenever a new version of the ICLI server is obtained and installed .

A user ID with DB2 SYSADM authority is needed to run these jobs. This user ID is needed only during the preparation of the SAP system installation or when a new ICLI PTF is installed.

> **Note:**
>
> If you have defined the RACF profile `BPX.DAEMON` of RACF's `FACILITY` class, you must add the FTP server user ID to this class profile. This enables the FTP daemon to submit the job on behalf of the user who requested the JCL submission service. For setting up a daemon, refer to "Setting Up for Daemons" in *z/OS UNIX System Services Planning*.

User IDs for selected SAP CCMS functions

The Computing Center Management System (CCMS) functions that involve submitting jobs on z/OS, performing DB2 utilities, and viewing system console log output require a TSO user with the authorities necessary to perform these tasks. The TSO user ID must be the same as that of the SAP system user that is using the CCMS functions, and must not be more than seven characters long.

These TSO users are not necessary during SAP setup and can be created after successful SAP system installation.

For further details, refer to the SAP documentation *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*.

Configuring ICLI PassTicket signon

The ICLI server, by default, expects the ICLI client to use PassTicket signon, which requires RACF Version 2.2 (or higher) or a compatible security product to be installed on z/OS. Any security product compatible with RACF PassTicket can be used, because the ICLI server uses a SAF (Security Authorization Facility) exit to invoke the validation function of the security product. On UNIX, Linux, and Windows, no separate product is needed.

If *no* security product is installed, you must disable the PassTicket signon function by setting the environment variable *ICLI_TRUSTED_CONNECTIONS* to 1 for the ICLI server *and* all of its clients. In this case, you will receive warning messages ICLC2120W on the client side and ICLS1049W on the server side.

In the following sections, the configurations needed to enable PassTicket signon are described for the security product RACF. Even if you are using a different security product, the description should give an idea of what you have to do.

Configuring PassTicket signon for ICLI servers

The default setting of the environment variable *ICLI_TRUSTED_CONNECTIONS* for both the client and the server ensures that PassTicket signon is enabled right from the beginning. However, before PassTicket signon can actually be used, you have to define a RACF profile entry for the class PTKTDATA for each ICLI server within the scope of the specific RACF.

You can use the following JCL job to create the RACF profile entry for an ICLI server:

```
//<JOBNAM>  JOB   USER=<SYSADM>,NOTIFY=<SYSADM>,CLASS=A,MSGCLASS=Q,
//               MSGLEVEL=(1,1)
//*RACF definitions for using PassTicket validation
//*
//DUMMY EXEC PGM=IKJEFT01
//*
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
  SETROPTS RACLIST(PTKTDATA)
  SETROPTS CLASSACT(PTKTDATA)
  RDEFINE  PTKTDATA <profile_name> UACC(NONE) -
          SSIGNON(KEYMASKED(<signon_key>))
 SETROPTS REFRESH RACLIST(PTKTDATA)
//*
```

Before you can run the job, replace the following placeholders with the appropriate values:

- <SYSADM>: The user ID with the authority to define and activate RACF class profiles.
- <profile_name>: The profile name of the class PTKTDATA.

  The profile name is composed of the application name and the user ID:

  <application_name>.<user_ID> (for example: ICLI0001.E1EICLI), where:

  - <application_name> is any character string, with a maximum length of eight characters. If you use ICLI as a prefix, followed by four characters of your choice, you can track SAP-related RACF data more easily.
  - <user_ID> is the user ID defined in RACF. It is recommended that you use the user ID (<SAPSID>ICLI) associated with the ICLI server (E1EICLI in the example).
- <signon_key>: An arbitrary string of 16 hexadecimal characters specifying the RACF-secured signon key.

For details on determining profile names, refer to *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

Configuring PassTicket signon for ICLI clients

The information needed to perform PassTicket signon is stored in a so-called *PassTicket signon file*. If you are only using one ICLI server within the scope of a

specific RACF, only one PassTicket signon file is needed. This signon file will be created by the SAP installation tool. In the following cases, you have to create PassTicket signon files manually:

- You are running multiple ICLI servers within the scope of one RACF. You need to define a separate PassTicket signon file for each profile entry you have created for the class PTKTDATA.
- You disabled PassTicket signon initially and enable it after the SAP installation tool has run.

PassTicket signon information is stored by default in file:

- `/usr/sap/<SAPSID>/SYS/exe/run/iclipt`

  or

  `/sapmnt/<SAPSID>/SYS/exe/run/iclipt`

  (for SAP application servers on UNIX or Linux)
- `\\<SAPGLOBALHOST>\SAPMNT\<SAPSID>\SYS\exe\run\iclipt`

  (for SAP application servers on Windows)

PassTicket signon files must have the following structure (lines starting with "/*" are comments and will be ignored):

```
/* start of file iclipt
<user_ID>
<application_name>
<signon_key>
/* end of file iclipt
```

The entries must be identical to the corresponding values in the profile of the RACF class PTKTDATA that was defined for the ICLI server. For details, see *z/OS SecureWay Security Server RACF Security Administrator's Guide*.

During the installation with the SAP installation tool, the user is prompted for these values and the signon file is generated automatically. Make sure that the proper permission bits are set for the file and ensure that the access is restricted (after installation, READ permission for user ID `<sapsid>adm` only).

If you are not using the default sign-on file `iclipt`, or if you are using more than one sign-on file, you use the ICLI client environment variable *ICLI_SIGNON_FILE* to assign a sign-on file to an ICLI client. If the environment variable *ICLI_SIGNON_FILE* is set and points to an existing file, this file will be used. Otherwise, the default sign-on file `iclipt` will be used.

All PassTicket signon files used within one RACF must differ in at least the user ID or the application name. This is necessary because a PassTicket is only good for a single signon to RACF. To avoid any problems that can result from two clients trying to sign on at the same time to different ICLI servers within one RACF using the same PassTicket information, you have two alternatives:

- Make sure that the clients use a different signon file for each ICLI server containing different passticket information.
- Use the NO REPLAY PROTECTION option when defining passtickets. For details, see the *Security Server RACF Security Administrator's Guide*, SA22-7683

Refused signon

In case the ICLI server refuses a signon attempt from the ICLI client, it issues error message ICLS1055E and does not create a new ICLI thread.

Chapter 3. Configuring z/OS

This chapter describes the setup and configuration steps you need to complete on the z/OS side before you can start installing the SAP system. Be aware that some of the steps in this process may require an IPL of your system, so you need to plan these changes carefully before you begin the SAP system installation.

Checking the PTF status to ensure proper code levels

Before starting your SAP installation, you should make sure that you have installed all prerequisite PTFs on your system. The list of required PTFs is available in SAP Note 81737, which is maintained by IBM and SAP. The APARs and service levels specified in the SAP Note are considered a required minimum for your system. Not applying this minimum service is an unnecessary and incalculable risk to your SAP system. Past experience has shown that problems may not be apparent immediately but can occur later, which means that you can never feel safe if you do not stay current with your z/OS and DB2 maintenance levels. The list is updated regularly, so check it periodically for new entries and apply them.

> **Note:**
>
> We strongly recommend that you set up the Automated PTF Check as described in the *SAP Database Administration Guide* after your SAP installation is complete. This PTF check tool can automatically retrieve the most current version of SAP Note 81737 from SAPNET and will check your z/OS and DB2 PTF levels against the prerequisites listed there. You should perform this check on a regular basis to stay current with your z/OS and DB2 maintenance levels.

z/OS setup for DB2

Storage management considerations

An SAP system includes a very large number of data sets, and the target volumes should have appropriately sized VTOC and VVDS: 400 and 180 tracks, respectively. This recommendation is based on a 3390-3 DASD unit.

DB2 logs for installation must tolerate up to 2 GB/hour. Archiving is required and at least 20 GB of archiving space must be available. Most of this space can be reclaimed once SAP on DB2 is up and running. The preferred archiving medium is disk, with tape silos representing the second-most preferred medium.

With regards to the VSAM ICF catalog size, allocate 50 cylinders for the primary extent and 5 cylinders for the secondary extent for each SAP system.

Setting up SMS

Use the Storage Management Subsystem (SMS) to manage your storage and to better cope with your growing storage needs. SMS is highly recommended for the DB2 VSAM datasets. SMS can also be utilized for the required HFS datasets used for /usr/sap/..., /sapmnt/..., and /u/.... During the installation of the SAP system, the SAP installation tool inquires if SMS is being used. In order for the

above to be done, it is highly recommended to install and configure SMS. In such a case, SMS storage groups need to be defined for SAP DB2 and SAP HFS. For more details, see *z/OS DFSMS: Implementing System-Managed Storage*.

Setting up Resource Recovery Services (RRS)

Introduction

ICLI and the SAP application server on z/OS exploit the Recoverable Resource Manager Services Attachment Facility (RRSAF). The RRSAF is a DB2 facility that relies on the z/OS component RRS (z/OS Transaction Management and Recoverable Resource Manager Services). RRS is required for each SAP system.

z/OS RRS provides system-wide services for coordinating two-phase commit operations across z/OS products. Before you can run an RRSAF application, z/OS RRS must be started. z/OS RRS runs in its own address space and can be started and stopped independently of DB2. RRS is a z/OS system logger application that records events related to protected resources. System logger is a z/OS component that allows an application to log data. You can log data from one system or from multiple systems across the sysplex. A system logger application can write log data into a log stream.

There are two types of log streams: coupling facility and DASD-only. You can use either one of them for setting up RRS, regardless of whether your system is data sharing or non-data sharing. Typical data sharing installations use the coupling facility log stream, while typical non-data-sharing installations use the DASD-only log stream.

Example of a minimal setup for RRS

1. **Add subsystem entries to parmlib IEFSSNnn:**
   ```
   SUBSYS SUBNAME(LOGR)
   INITRTN(IXGSSINT)
   SUBSYS SUBNAME(RRS)
   ```
2. **SMS:**

   Edit SMS routines, add a data class for log streams, with shrlevel(3,3) and volume count 1. The default HLQ is IXGLOGR. Edit selection routines to put log stream data sets and logger couple data sets in the new data class and appropriate storage classes.
3. **Activate SMS changes.**
4. **Other parmlib entries:**

   Ensure that IEASYSnn has the following statements:
   ```
   COUPLE=nn
   PLEXCFG=MONOPLEX
   SYSNAME=xxxx
   ```

   where nn is the COUPLEnn member, and xxxx is your system name.
5. **Create COUPLEnn member, with content similar to:**
   ```
   COUPLE SYSPLEX(SAP2)
     PCOUPLE(SYS1.XCF.CDS1SAP2)
     ACOUPLE(SYS1.XCF.CDS2SAP2)
   DATA
     TYPE(LOGR)
     PCOUPLE(LOGR.SAP2.DS1)
     ACOUPLE(LOGR.SAP2.DS2)
   ```
6. **Allocate data sets:**

   Allocate data sets for sysplex couple data sets and logger couple data sets.

7. **IPL:**

   IPL to activate the sysplex and system logger. The system logger (IXGLOGR) should appear in a **D A,ALL** command.

8. **Set up RRS:**

   a. Determine naming conventions for the RRS log streams and log stream data sets. The default log stream names are `ATR.gname.logstreamname`, while the default data set names are `IXGLOGR.ATR.gname.logstreamname`, where `gname` (group name) defaults to the sysplex name. Note that the group name is supported by RRS and allows for separation of groups such as production and test RRS groups within a single sysplex. However, APPC does not support group names; it is thus not possible to separate APPC into test and production groups within the sysplex.

   b. Ensure that the log stream data sets are defined properly to DFSMS, i.e., automated class selection (ACS) routines (see *z/OS MVS Setting Up a Sysplex*), storage class, group, etc. and SHARE OPTIONS(3,3).

   c. Set up RRS proc: see SYS1.SAMPLIB(ATRRRS).

Installing the RRS panels: You must install the RRS panels (e.g., to your main ISPF panel), because they present the only mechanism for determining the state of the various resource managers. The installation of the panels is described in *z/OS MVS Programming: Resource Recovery*.

To display resource manager information, select option 2 on the RRS panel and press **ENTER**:

```
                              RRS
   Option ===> _____


   Select an option and press ENTER:

    1 Browse an RRS log stream
    2 Display/Update RRS related Resource Manager information
    3 Display/Update RRS Unit of Recovery information
    4 Display/Update RRS related Work Manager information
    5 Display/Update RRS UR selection criteria profiles
    6 Display RRS-related system information
```

Preventing CF structures from filling up: When off-load processing hangs because the data set director is full, the CF structures will fill up. To avoid this effect:

1. Utilize the program IEAMDBLG on a nightly basis to delete unneeded data or to copy it to a DASD or tape medium outside of the log stream.

2. Utilize the attributes *RETPD*(xx) and *AUTODELETE*(YES/NO) per log stream as required. These parameters permit automatic deletion of log stream data after a given retention period (retpd) as desired.

WLM considerations for DB2 subsystems

Note: All discussions in the book concerning WLM refer implicitly to WLM goal mode. Compatibility mode is no longer supported as of z/OS V1R3.

In goal mode, the WLM application environments are required and a WLM service definition must be created and installed. For more information, see "Workload Management (WLM)" on page 36 and "Workload Management (WLM) — Classification of SAP workload" on page 82.

Workload Management (WLM)

To manage the performance of your z/OS system (including system and business applications) using WLM, you can configure and manage your performance settings.

You can adapt the performance of your workload in business terms by specifying business importance and response time or velocity goals for your applications. WLM enables you to specify priorities for different kinds of SAP workload that are directly based on definitions you make for your SAP systems.

The basic setup required for running a system in WLM mode is described in the following sections.

WLM setup

Settings required for the entire system

To successfully run your system in WLM mode, you must ensure the following:

- The subsystems/address spaces **VTAM**, **IRLM**, and **RRS** must be mapped to the default service class **SYSSTC**.
- **MSTR** and **DBM1** must be assigned a high-velocity goal.
- TCP/IP, OMVS, and NFS (if used) must be assigned velocity goals lower than those of the subsystems/address spaces mentioned above.
- The velocity goals of the ICLI server and the SAP application server on z/OS must be lower than those of all the subsystems/address spaces mentioned above. The required minimum WLM setup for the ICLI server and for the SAP application server on z/OS (if it is to run WLM-enabled) is described below.

Minimum WLM configuration for SAP workload

Since the ICLI server is WLM enabled by default and always runs under WLM subsystem type *SAP*, you must implement a minimum WLM configuration for SAP workload. Otherwise, the enclaves created for SAP workload would run with the service class SYSOTHER, which would result in poor performance for the ICLI server work threads and the SAP processes associated with the enclaves.

The initial WLM setup for the ICLI server is simple:

- You define a special WLM service class (e.g., SAPMED). This service class should have a velocity goal that meets the requirements mentioned in "Settings required for the entire system."
- You make it the default service class for WLM subsystem type *SAP*.

As a result, all workload running under WLM subsystem type *SAP* will run with the velocity goal of this service class. (For an explanation of the terms *WLM subsystem type* and *WLM service class*, refer to *z/OS MVS Planning: Workload Management*).

For the SAP application server on z/OS, WLM is disabled by default. This means that its processes do not run with the velocity goals specified for WLM subsystem type *SAP*, but rather with those specified for WLM subsystem type *OMVS*.

If you want to enable WLM to manage the workload of an SAP application server on z/OS using the velocity goals defined for the WLM subsystem type *SAP*, you need to switch on WLM management for the SAP application server. This is achieved by setting the SAP profile parameter rdisp/prio/wlm/enabled = 1.

For SAP kernel release 6.20, see the section "Switching on WLM Management" in the SAP document *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* for details on how to switch on WLM management for the SAP application server.

When you are using the SAP 6.40 kernel and an SAP application server instance on z/OS, you can consider using additional WLM work qualifiers as introduced with SAP 6.40 to prioritize the workload of an SAP application server on z/OS. For more information on this topic, see the sections "Workload Management (WLM)" and "Workload Management - Classification of SAP Workload" in the *SAP Planning Guide* for SAP basis release 6.40.

Preparing SAP application servers for WLM

Performance management by WLM for the ICLI server and the SAP application server on z/OS gives you the following options:

- You can define specific performance goals for workload created by a specific SAP application server instance. This is possible because each application server can be uniquely identified by its SAP system ID (or database attach name), the name of the host on which it is running, and/or the SAP system number. This also means that you can specify different performance goals for workload generated by different SAP application servers on non-z/OS hosts even if these application servers are serviced by the same ICLI server.
- You can define different performance goals for different types of work. This is possible by using the SAP work classification type, e.g., you may give SAP dialogs a higher priority than SAP batch jobs.
- You can assign performance goals with a high degree of granularity; that is, you can, for example, assign different priorities to SAP dialogs depending on the SAP application server they belong to. You can achieve this by making sure that the SAP workload of each SAP application server can be uniquely identified as described above.

In previous SAP releases, you had to use different user IDs for ICLI server instances if you wanted to set different performance goals for workload running on different ICLI servers. While you can still proceed to use such a setup, it is not recommended to set up new SAP systems in such a way. This is because using different user IDs for some or all ICLI server instances increases EDM Pool storage usage and decreases performance through global cache hit ratio degradation. In order to use specific workload classifications for different ICLI server instances, you should use the new attributes *host name*, *SAP system number*, and DB2 *database attach name* instead. The following sections explain how you should set up your SAP system to ensure that you can fully exploit WLM for SAP workload in the future without having to reconfigure or even reinstall your system.

The ICLI server and the SAP application server on z/OS supply the following workload attributes:

- The *database attach name* defined for an SAP application server as described in the *SAP installation guides*. It is recommended to use the SAP system ID as the database attach name. For an SAP application server running on z/OS, this is always the SAP system ID. In the scenarios below it is assumed that the database attach name is always equivalent to the SAP system ID.
- The *host name*, which is one of the following:
  - the name of the non-z/OS host on which an SAP application server is running.
  - the name of the z/OS host on which the SAP application server is running.

- The *SAP system number* as specified by the SAP profile parameter SAPSYSTEM.
- The *SAP work classification type* (one of DIALOG, BATCH, UPDATE, UPDATE2, GENERIC, SPOOL, ENQUEUE).
- The *user ID* used to start an ICLI server or an SAP application server on z/OS. The user ID is mainly there to support older WLM setups created when the user ID and the SAP work classification type were the only attributes supplied by the servers.

For SAP application servers on non-z/OS hosts, the ICLI client passes all of these attributes to the ICLI server, except for the user ID, which is an attribute of the ICLI server itself.

In order to be able to use WLM efficiently, there must be a high degree of differentiation within your SAP workload with respect to the attributes mentioned above. This means that the more you can distinguish within your SAP workload, the better you can assign specific priorities (performance goals) to different kinds of SAP workload. Therefore, when setting up your SAP systems, you should always consider the following:

- For each SAP application server instance in your environment, choose the database attach name (or SAP system ID) and the SAP system numbers in such a way that, together with the host name, these attributes identify each SAP application server instance uniquely.
- For performance reasons, avoid using different user IDs for ICLI server instances to distinguish their workloads (see also "User ID to run the ICLI server" on page 27).
- It is not possible to specify priorities related directly to SAP system users or SAP system user groups. However, you can define priorities for SAP system user groups indirectly as shown in "Scenario 3" on page 40.

Note that WLM does not manage the workload of SAP application servers on non-z/OS hosts directly. It manages the workload of the ICLI server threads working for the corresponding ICLI clients. This means that you set the performance goals for the workload of a "remote" SAP application server by defining the goals for the ICLI server that processes the workload on z/OS.

WLM scenarios: The scenarios below show how you can use the attributes database attach name, host name, and SAP system number to distinguish the SAP workloads created by different SAP application servers. This is required to be able to define specific performance goals for SAP workload created by a specific SAP application server. To simplify the scenarios, the term "SAP system ID" is used as a synonym for "database attach name". For more detailed information on setting performance goals for SAP workload, see "Workload Management (WLM) — Classification of SAP workload" on page 82.

Scenario 1: In the scenario depicted in Figure 11 on page 39, an SAP system is set up with a central instance and a dialog instance that runs batch jobs only. Both instances have the same SAP system ID (C11) and the same SAP system number (11). Both run against DB2 subsystem C11. Since the central instance is running on host AIXHOST1 while the dialog instance is running on AIXHOST2, the host name attribute can be used to distinguish between the instances' workloads.

If the workload on the central instance is to have a higher priority than that on the dialog instance, WLM can be set up in such a way that workload running on host AIXHOST1 has a higher priority than that on AIXHOST2.
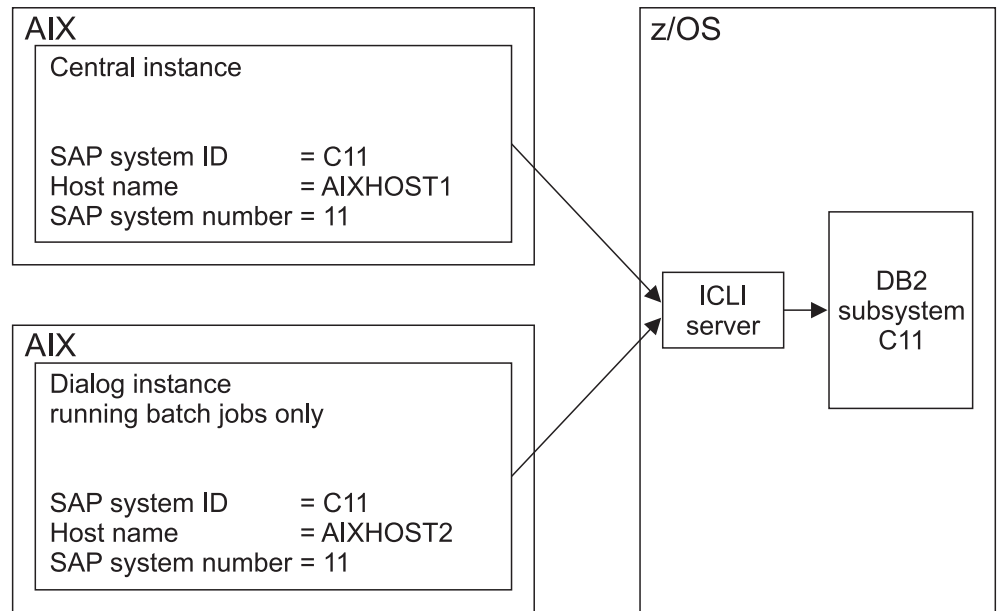
Figure 11. WLM scenario 1

Scenario 2: In the scenario depicted in Figure 12, a production system with SAP system ID C11 and a test system with SAP system ID C12 are running on the same host (AIXHOST), where the production system has SAP system number 11 and the test system has SAP system number 12. Since the production system is running against DB2 subsystem C11 and the test system is running against DB2 subsystem C12, and since the system numbers must also be different if the system IDs are, either the SAP system ID or the SAP system number can be used to distinguish between the workloads of the two systems.

If the workload on the production system is to have a higher priority than that on the test system, WLM can be set up in such a way that workload on the SAP system with the ID C11 has a higher priority than workload on the SAP system with the ID C12.
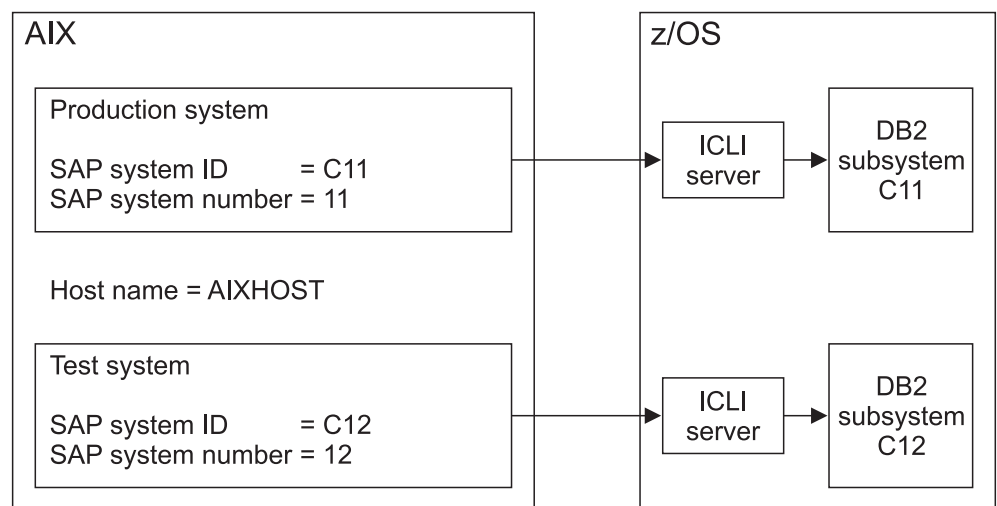

Figure 12. WLM scenario 2

Scenario 3:   In the scenario depicted in Figure 13, the two application server instances A and B have the same SAP system ID (C11) and they run on the same host (AIXHOST), but instance A has SAP system number 11 and instance B has SAP system number 12. Both instances are running against DB2 subsystem C11. In this case, the SAP system number can be used to distinguish between the systems' workloads.

If there are two user groups (A and B), where user group A can only log on to instance A, while users of group B can only log on to instance B, and the workload generated by user group A is to have a higher priority than that generated by user group B, WLM can be set up in such a way that workload running on instance A (with SAP system number 11) has a higher priority than workload on instance B (with SAP system number 12).

```
AIX                                            z/OS
 ┌─────────────────────────────────┐    ┌──────────────────────────┐
 │  Instance A                     │    │                          │
 │  SAP system ID       = C11      │    │                          │
 │  SAP system number = 11         │    │   ┌──────┐   ┌─────────┐  │
 │                                 │    │   │ ICLI │   │  DB2    │  │
 │  Host name = AIXHOST            │    │   │server│─▶│subsystem│  │
 │  Instance B                     │    │   └──────┘   │  C11    │  │
 │  SAP system ID       = C11      │    │              └─────────┘  │
 │  SAP system number = 12         │    │                          │
 └─────────────────────────────────┘    └──────────────────────────┘
```
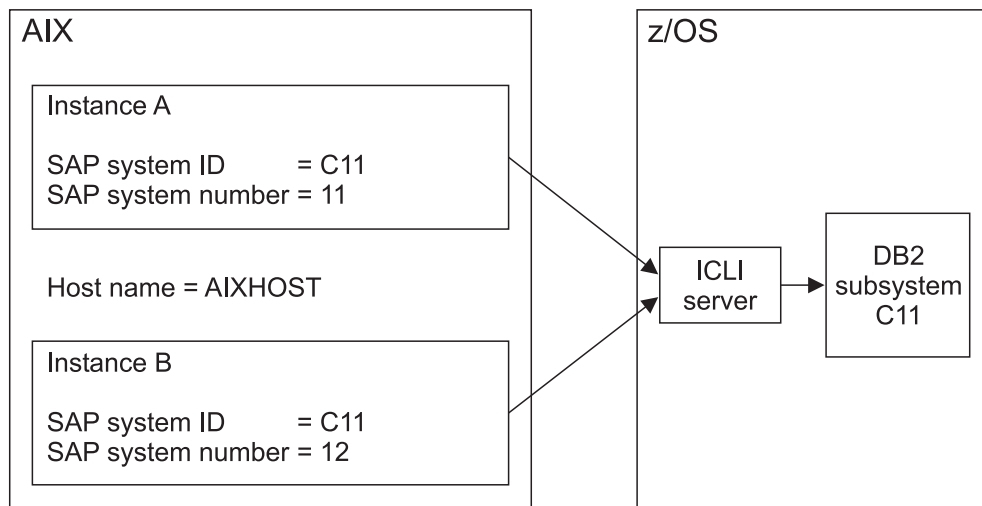
Figure 13. WLM scenario 3

Configuring the ICLI server and SAP application server on z/OS

The following configuration description applies to both the ICLI server and the SAP application server on z/OS. Since some of the changes described here only become active after an IPL, plan the configuration changes early in advance and schedule an IPL of the system before you begin with the actual installation of the SAP system.

Language Environment setup

When using the SAP 6.40 kernel, you need to ensure that you have set up your Language Environment to allow running applications that exploit Extra Performance Linkage (XPLink). In particular, make sure that the runtime library SCEERUN2 is included in the LINKLIST concatenation. This is because the ICLI server (see "ICLI client/server internals" on page 95)and the SAP application server on z/OS exploit the XPLink feature with the SAP 6.40 kernel. For details, see "Choosing your Language Environment run-time library access - Methods of Language Environment data set access - STEPLIB" in *z/OS Language Environment Customization* or "Running an XPLink application" in the IBM Redbook *XPLink: OS/390 Extra Performance Linkage*, SG24-5991.

Selecting the UNIX System Services parameters

The following parameters in the BPXPRMxx parmlib member need to be increased. For a description of all UNIX System Services parameters, see the IBM manual *z/OS UNIX System Services Planning*.

Table 5. Recommended values for parameters in the BPXPRMxx parmlib member

Parameter	Recommended value	z/OS default value	Maximum value
IPCSHMMPAGES	**524287**	25600	524287
MAXASSIZE	**2147483647**	209715200	2147483647
MAXCORESIZE	**2147483647**	4194304 (4 MB)	2147483647
MAXCPUTIME	**2147483647**	1000	2147483647
MAXFILEPROC	**2000**	2000	65535
MAXFILESIZE	**(omit statement)**	1000 (4 MB)	2147483647
MAXPROCUSER	**200**	25	32767
MAXTHREADS	**1000**	200	100000
MAXTHREADTASKS	**1000**	1000	32767
USERIDALIASTABLE	**/etc/ualiastable**		

UNIX System Services parameters in detail

The recommended values in Table 5 are sufficient for large SAP systems running ICLI servers and SAP application servers. Since no resources can be saved by setting lower limits, these values are appropriate for SAP systems of any size.

The values can be changed dynamically with the operator command **SETOMVS**, but, for some values, dynamic changing is limited to four times the original value. If a value change exceeds this limit, an IPL is required to activate the changes.

IPCSHMMPAGES
> This parameter limits the size of a single shared memory segment. It should be set to its maximum value of 2 GB. The largest segment as requested by the application server is the one for the SAP ABAP Program Buffer, which fills up several hundred MB.

MAXASSIZE
> This parameter limits the region size of an address space, that is, the amount of virtual storage a program can use. If you log on via **rlogin**, the *MAXASSIZE* parameter takes over the role of the TSO *REGION* size. If a user enters the z/OS UNIX shell via TSO log-on, the UNIX processes inherit the TSO settings for region size and maximum CPU time. Therefore, it must be ensured that the user's TSO session is running with REGION=0M (that is, unlimited) and TIME=NOLIMIT if the program is started in this way. The same applies if the ICLI server or SAP application server is started as a started task.
>
> When the *REGION* size is too small, the ICLI server abends and writes the following message to STDERR :
> ```
> 671088653 FSUM7744 Killed ./fome620s (or fome640s)
> ```
>
> In the z/OS console log, you will find the following z/OS Language Environment abend:
> ```
> IEF450I XXX13 STEP1 -ABEND=S000 U4093 REASON=XXXXXXXX
> ```
>
> where XXXXXXXX can be either 0000001C or 00000044.

The IEFUSI user exit can modify the *REGION* size of an address space. Changing the *REGION* size of address spaces in the OMVS subsystem category is strongly discouraged. Such user exits are defined in SYS1.PARMLIB(SMFPRMxx). IEFUSI must be omitted in the entry `SUBSYS(OMVS,EXITS(...))`.

The SAP application server may require up to 2 GB of virtual storage, depending on the SAP profile settings and the workload. At startup, the SAP application server therefore verifies that the region size is set to 2 GB and generates an error message otherwise.

The available virtual storage per process can be checked by running the SAP tool **memlimits**, which is part of the SAP application server. The available storage per process should be 1600 MB or more. Here is an example for the output of **memlimits**:

```
Check the maximum data size per process (malloc)
Check the available swap space (malloc in several processes)
Process 3827 allocating ... Size = 1752MB Total: 1752MB
Process 3829 allocating ... Size = 1752MB Total: 3504MB
```

MAXCORESIZE
> This parameter specifies the maximum size of a core dump file.

MAXCPUTIME
> This parameter limits the CPU time a process can use. It must be set to the maximum value to indicate unlimited CPU time.
>
> If you log on via `rlogin`, the *MAXCPUTIME* parameter takes over the role of the TSO *TIME* parameter. If a user enters the z/OS UNIX shell via TSO log-on, the UNIX processes inherit the TSO settings for region size and maximum CPU time. Therefore, it must be ensured that the user's TSO session is running with `REGION=0M` (this means, unlimited) and `TIME=NOLIMIT` if the program is started in this way. The same applies when the ICLI server or the SAP application server is started as a started task.
>
> The ICLI server and the SAP application server check at startup that the CPU time is not limited.

MAXFILEPROC
> This parameter limits the number of file handles for files and TCP/IP sockets per process (see "Setting the maximum number of INET and UNIX sockets" on page 45).

MAXFILESIZE
> This parameter limits the size of a file. Omit this parameter or set it to the value of NOLIMIT to allow the file size to be unlimited.

MAXPROCUSER
> This parameter limits the number of processes a user can start. When running an SAP application server on z/OS, the value must be increased, since the number of SAP work processes plus other processes exceeds the default value.

MAXTHREADTASKS and MAXTHREADS
> These parameters limit the number of threads the ICLI server can start. The default values for these parameters may be too low if you want to run a higher number of ICLI server threads. This is the case if the number of SAP work processes connecting to an ICLI server is high.

USERIDALIASTABLE
>This table allows you to map z/OS user IDs to lowercase UNIX(-style) user
>IDs (see section "User ID <sapsid>adm to run the SAP application server
>on z/OS" on page 26). You can enter any alias table name of your choice.

Specifying limits for individual users: The following parameters can also be
specified for individual users:

- *MAXASSIZE*
- *MAXCPUTIME*
- *MAXPROCUSER*
- *MAXFILEPROC*
- *MAXTHREADS*

Instead of increasing the system limits, the attributes of the user `<sapsid>adm` can
be specified as *ASSIZEMAX, CPUTIMEMAX, PROCUSERMAX, FILEPROCMAX*,
and *THREADSMAX* in the OMVS segment in RACF, for example:

```
ADDUSER C11ADM OMVS(CPUTIMEMAX=2147483647)
```

Checking the settings of the UNIX System Services parameters

The current settings can be checked using the operator command **D OMVS,O**.

Example:

```
OMVS 000E ACTIVE OMVS=(67,AL,0F)
CURRENT UNIX CONFIGURATION SETTINGS:
MAXPROCSYS     =        512    MAXPROCUSER    =        200
MAXFILEPROC    =       2000    MAXFILESIZE    = NOLIMIT
MAXCPUTIME     = 2147483647    MAXUIDS        =        200
MAXRTYS        =        500    MAXPTYS        =        256
MAXMMAPAREA    =       4096    MAXASSIZE      = 2147483647
MAXTHREADS     =       2000    MAXTHREADTASKS =       1000
MAXCORESIZE    = 2147483647    MAXSHAREPAGES  =   32768000
IPCMSGQBYTES   =     262144    IPCMSGQMNUM    =      10000
IPCMSGNIDS     =        500    IPCSEMNIDS     =        500
IPCSEMNOPS     =         25    IPCSEMNSEMS    =         25
IPCSHMMPAGES   =     524287    IPCSHMNIDS     =        500
IPCSHMNSEGS    =        100    IPCSHMSPAGES   =     262144
SUPERUSER      = BPXROOT       FORKCOPY       = COW
STEPLIBLIST    = /system/steplib
USERIDALIASTABLE= /etc/ualiastable
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS  =       1000    SHRLIBRGNSIZE = 67108864
SHRLIBMAXPAGES = 4096          VERSION = 12FREA
SYSCALL COUNTS = NO            TTYGROUP = TTY
SYSPLEX = YES                  BRLM SERVER = N/A
LIMMSG = NONE                  AUTOCVT = OFF
RESOLVER PROC = DEFAULT
```

Defining the TCP/IP port range

The port range from which the ICLI connection port must be chosen depends on the communication stack you will use on z/OS (see "Defining the BPXPRMxx parmlib member" in the *Connectivity Guide* for more information on the Integrated INET stack and the Common INET stack).

If you are using the recommended Integrated INET stack, you can skip this section.

If you are running the Common INET stack, which allows you to have multiple stacks, you have to divide the set of usable ports between these stacks. For a description of the general port usage schema, see "Considerations for selecting an ICLI connection port" on page 99. To define the port range, you use the parameters *INADDRANYCOUNT* and *INADDRANYPORT* (in the BPXPRMxx parmlib member) and the parameter *PORTRANGE* (specified in PROFILE.TCPIP). You must provide for enough port numbers in the range defined by *INADDRANYCOUNT* and *INADDRANYPORT*. Make sure that the value for *PORTRANGE* specified in PROFILE.TCPIP exactly matches this range.

Since each of ICLI's work threads will use one port from this range, set *INADDRANYCOUNT* to a value that is equal to or greater than the number of SAP work processes that will connect to the ICLI server(s) on your z/OS system. Calculate the number as follows:

The maximum number of SAP work processes you ever plan to run against all ICLI servers on this z/OS system plus the number of additional ports reserved for standalone tools and reconnecting work processes. Reserve at least 20 additional ports.

Setting the maximum number of INET and UNIX sockets

When you are running an ICLI server on z/OS, make sure that the maximum number of AF_INET sockets is at its default value of 2000 or higher. The total number of sockets needed by all ICLI servers that run on this z/OS system is equal to the maximum number of SAP database connections that will ever be handled by these ICLI servers.

NFS client configuration

If an NFS client on z/OS is used (this applies, for example, if your SAP files reside on AIX but you run an SAP application server instance or the SAP upgrade tool **R3up** on z/OS), you have to specify the NFS client parameter *DISABLELLA(Y)*. This parameter is in the BPXPRMxx parmlib member.

```
FILESYSTYPE TYPE(NFS) PARM('DISABLELLA(Y)') ...
```

For further information on the parameter, see APAR OW35419.

If you are running in a UNIX System Services Shared HFS environment, it is advisable to run the NFS client in a colony address space under the Master Subsystem. This will allow JES2 to be stopped without shutting down the NFS client, which propagates the *UNMOUNT* to all LPARs (see APAR OW48709). Add the following keywords to the current FILESYSTYPE statement:

```
ASNAME(proc-name,'SUB=MSTR')
```

JES parameters

The z/OS Scheduler Work Area (SWA) must be allocated above the 16 MB line. In the JES parmlib member, the STCCLASS must be defined with SWA=ABOVE. The current definition can be checked using the operator command **$T STCCLASS**. This will facilitate parallel execution of various installation and administration jobs (e.g., RUNSTATS) and will provide for configurations with multiple subsystems with the same z/OS. Furthermore, a shortage of memory below 16 MB will restrict the number of ICLI threads that can be created by an ICLI address space.

For SAP system installation, at least 15 MB of JES spool space is required. You must therefore check your spool space before you start the installation. (If spool space is low, you can make more space available by clearing all garbage on the current spool space or by adding a new spool volume.)

> **Caution:**
> For the RUNSTATS step of the SAP installation tool, five Class A initiators are required to run five RUNSTATS jobs in parallel. If there are too few Class A initiators, individual RUNSTATS jobs will fail with RC=8 and NO ERROR MESSAGE. Following the conversion to SMS, management batch jobs may suffer a variety of virtual storage abends such as abend878 rc10 or abend80a. If such a situation occurs, see APAR II07347 for recommendations.

Linkage Index and Address Space ID considerations

Linkage Indexes (LX) and Address Space IDs (ASIDs) are z/OS control structures that are used in inter-address space communication. Their number is limited and decreases with each DB2 subsystem shutdown. Be sure to read the following considerations, because incorrect settings or inappropriate practices can make an IPL necessary.

Linkage Index (LX) considerations

The total number of LXs per z/OS image cannot exceed 2048.

There are two types of LXs: system LXs and non-system LXs. The guaranteed number of system LXs is specified in the NSYSLX parameter (in the IEASYSxx member of SYS1.PARMLIB). The default value for NSYSLX is 165, which is appropriate for most installations including the SAP system. When an address space that uses the LXs terminates, these LXs become temporarily unavailable. The duration of the unavailability depends on the LX type:

- The system LXs can be reused the next time the address space is started, providing the corresponding application is coded to enable such a reuse. This is the case for DB2, IRLM and RRS. Otherwise, the system LXs are unavailable until the next IPL.
- The non-system LXs are unavailable as long as there is any active address space which has cross memory connections with the terminated address space.

Let's see what this means for DB2 in SAP system environments:

- The DB2 subsystem itself uses one system and one non-system LX.
- RRS uses one system LX. At RRS shutdown, the LX becomes unavailable until the next start of RRS or the next IPL.
- IRLM uses one system LX. At RRS shutdown, the LX becomes unavailable until the next start of the same IRLM or the next IPL.

At DB2 shutdown, the system LX becomes unavailable until the next start of the same DB2 or until the next IPL. Therefore, this LX is not a concern with regard to exhausting the number of available LXs. On the other hand, since SAP uses RRS to attach to DB2, the non-system LX becomes unavailable until the RRS is shut down (because RRS preserves the cross memory binds to DB2).

If you do not shut down, you can have 1883 DB2 shutdowns before an IPL becomes necessary. Note however, that the number of shutdowns before an IPL is needed decreases if multiple DB2 subsystems reside in one z/OS image.

Address Space ID (ASID) considerations

Whenever a DB2 subsystem shuts down, the Address Space IDs (ASIDs) of the associated address spaces cannot be used (for integrity reasons) until the next IPL. For most SAP customers that would be 4 ASIDs, corresponding to the MSTR, DBM1, IRLM and SPAS address spaces.

The total number of ASIDs is determined by the value of the *RSVNONR* parameter in PARMLIB member IEASYSnn. There is an implicit limit for *RSVNONR*. Namely, the sum of *MAXUSER*, *RSVNONR* and *RSVSTRT* must not exceed 32767. *MAXUSER* and *RSVSTRT* are other parameters in IEASYSnn. Increasing the sum results in a small increase of the ASVT (Address Space Vector Table) size which resides below the 16 MB addressing range line: each ASVT entry occupies only 4 bytes.

Here is an example:

If *MAXUSER* is set to 512 and *RSVSTRT* to 100 (common values for these parameters), *RSVNONR* can be set to 32155. The ASVT size will be 125K which is acceptable. However, this will ensure 32155/3 = 10718 DB2 shutdowns before an IPL is necessary. Again, the number of shutdowns decreases with multiple DB2 subsystems per z/OS image.

Service class definitions

Setting the appropriate service class definitions is a prerequisite for satisfactory performance. Be sure to read "Workload Management (WLM) — Classification of SAP workload" on page 82 and apply the recommendations documented therein.

Environment variables

The ICLI server and SAP application server set their runtime options for the z/OS Language Environment to optimally use the storage and avoid heap and stack fragmentation. The environment variable _CEE_RUNOPTS would be used to override these internal defaults, but this is not recommended.

Because of undesired side effects, the system environment variable _BPX_SHAREAS=REUSE must not be used when running any SAP utilities. In particular, before starting the SAP installation tool or **R3up**, make sure that _BPX_SHAREAS is set to **NO**.

During SAP system installation, the default user profiles of the user ID <sapsid>adm will be tailored to include the appropriate settings.

Maximum number of threads

The maximum number of threads that can be created by a single ICLI server instance under optimal conditions is about 500. To reach the maximum number of threads, ensure that the following recommendations are met:

- DB2 thread limit

  Make sure that DB2 supports the required number of threads by setting the DB2 system parameters *IDBACK* and *CTHREAD* accordingly. For a detailed description of the DB2 system parameters, refer to the *SAP installation guides*.

- *REGION* size

  Set the region size to 2 GB to ensure that this parameter does not limit the number of threads. Furthermore, consider the memory available below the 16 MB line. For every thread that is created, a certain amount of memory is needed below the 16 MB line. Make sure that the private area available to the ICLI server is not restricted.

- Setting of CEE runtime options

  Make sure that the value for the Language Environment runtime option *THREADSTACK* is set to the IBM-supplied default. Setting this option to a different value might result in a dramatic reduction in the number of threads that can be created by the ICLI server.

Additional setup for the application server on z/OS

TCP/IP setup

The SAP system requires a consistent, case-sensitive TCP/IP setup. Furthermore, the SAP system requires that the host name be no more than eight characters in length. There are different places in which SAP uses or compares the host name. These instances must all be identical. During the installation of the SAP system, the length of the host name and the consistent setup are verified by the SAP installation tool:

- The host name as defined in TCP/IP. The host name is returned by the command **hostname -s** . The same value is also returned by the C function **gethostname()**.

- The host name returned by the name resolution service for the specified IP address. Name resolution is done with the command **nslookup** *ip-address*. The same value is also returned by the C function **gethostbyaddr()**.
- The SAP profile parameter or environment variable *SAPDBHOST*
- The host identifier in several SAP profile parameters, e.g., *rdisp/enqname*

Since the domain name server usually returns the host name in lowercase characters, our recommendation is to define the TCP/IP host name in lowercase as well.

On z/OS, the host name can be defined in different places. If the host name is not explicitly defined by the keyword **HOSTNAME**, the system uses the node name instead. The node name is the parameter specified for starting VMCF (Virtual Machine Communication Facility). It is defined either as part of the VMCF subsystem definition in SYS1.PARMLIB(IEFSSNxx) or as a parameter of the VMCF startup procedure. We recommend leaving the node name as is and using the keyword **HOSTNAME** to specify the appropriate host name. The example below shows the **HOSTNAME** definition in the TCP/IP system parameter file SYS1.TCPPARMS(TCPDATA):

```
DOMAINORIGIN domainname.de
SAPF:    HOSTNAME  hostname1
SAPG:    HOSTNAME  hostname2
```

In this example, SAPF and SAPG are the MVS node names as specified in SYS1.PARMLIB(IEFSSNxx). For the name resolution, the local /etc/hosts file or the TCP/IP site info data set and/or the domain name server are used and should be checked for consistency.

For performance reasons, it is required that localhost be resolved locally to 127.0.0.1. You must therefore define it in the file /etc/hosts or the TCP/IP site info data set:

```
127.0.0.1    localhost
```

The consistency of the definition can be verified by comparing the output of the TSO commands **HOMETEST** and **NSLOOKUP**.

Sample output of **HOMETEST**:

```
EZA0620I The TCP/IP system parameter file used will be "SYS1.TCPPARMS(TCPDATA)".
EZA0621I The FTP configuration parameter file used will be "SYS1.TCP.FTP.DATA".
EZA0602I TCP Host Name is: hostname1.domainname.de

EZA0605I Using Name Server to Resolve hostname1.domainname.de
EZA0611I The following IP addresses correspond to TCP Host Name:
         hostname1.domainname.de
EZA0612I 10.199.19.54

EZA0614I The following IP addresses are the HOME IP addresses defined in
         PROFILE.TCPIP:
EZA0615I 10.199.19.54
EZA0615I 10.156.19.54
EZA0615I 10.156.120.85
EZA0615I 127.0.0.1

EZA0618I All IP addresses for hostname1.domainname.de are in the HOME list!

EZA0622I Hometest was successful - all Tests Passed!
```

Sample output of TSO command **NSLOOKUP** 10.199.19.54:

```
EZB3170I Server: ...
EZB3172I Address: ...

EZB3170I Name:    hostname1.domainname.de
EZB3172I Address: 10.199.19.54
```

When TCP/IP resolves a port name, it checks if the HFS file /etc/services exists. If the file is found, TCP/IP restricts its search to this file. Otherwise, the MVS data set <TCPIP>.ETC.SERVICES is used for port name resolution. (The qualifier <TCPIP> stands for the high level qualifier under which the TCP/IP product was installed.)

During the installation of the SAP system, the SAP installation tool expects /etc/services to exist and appends the required port name definitions, for example:

```
sapdp13          3213/tcp          # disp+work
sapgw13          3313/tcp          # gateway
sapmsD6A         3613/tcp          # msg server
```

If /etc/services is not yet used at your installation, you have the following options:

- If you decide to use /etc/services from now on, copy the MVS data set <TCPIP>.ETC.SERVICES to /etc/services before running the SAP installation tool.
- Otherwise, the SAP installation tool will recognize that the file does not exist and will create a new file named /etc/services.sap containing the missing port names. A warning message will be displayed. The administrator has to add those port names to <TCPIP>.ETC.SERVICES manually.

The hostname command for application servers on z/OS

The SAP startup shell script for the application server uses the UNIX command **hostname** to build the profile name.

On z/OS, **hostname** returns the fully qualified host name (including the domain name), while **hostname -s** returns the short host name (without the domain name). The SAP shell scripts expect the short host name. The SAP installation tool therefore provides an alias definition in the profiles of user <sapsid>adm:

```
alias hostname='hostname -s'
```

File system setup

ASCII/EBCDIC considerations

To use the SAP J2EE Engine (which is available as an add-in for 6.20) on z/OS, you must set up the user environment and NFS mounts to exploit the Enhanced ASCII function before you can begin the installation.

The SAP Web application server on z/OS runs in ASCII mode. That means that all data inside the application server is represented in ASCII. The SAP J2EE server also expects all of its files to be ASCII.

On the other hand, the z/OS UNIX System Services shell and utilities expect EBCDIC data when processing a file (e.g., when editing it or when running a script).

In previous releases (and when using a 6.20 kernel), this situation was resolved by storing files on z/OS in the EBCDIC code page. If files were shared between ASCII UNIX, Linux, or Windows systems, a conversion was needed. An ASCII/EBCDIC

converter shipped with ICLI ensured that the SAP application server saw all data in ASCII, although the file system provided it in EBCDIC.

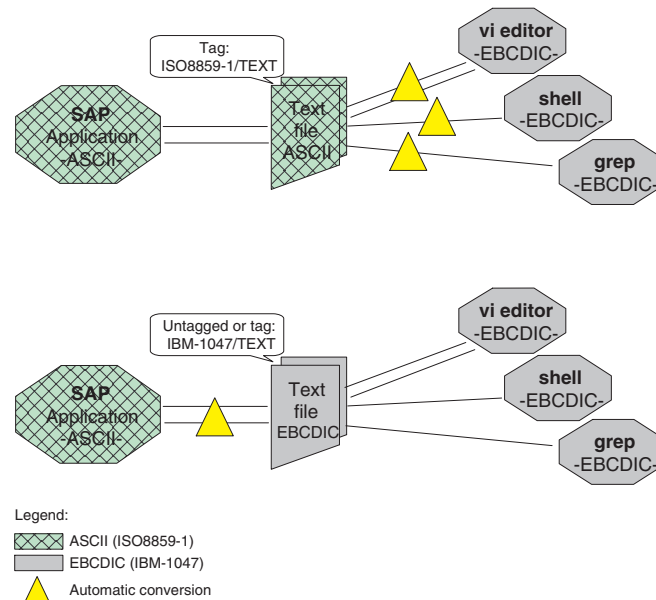With the introduction of the SAP J2EE server on z/OS, file handling has changed.

Figure 14. Automatic conversion provided by the Enhanced ASCII feature

Enhanced ASCII feature overview: Beginning with z/OS V1R2, the operating system provides a feature to support ASCII programs and ASCII files on z/OS. With this so-called Enhanced ASCII feature, ASCII-compiled programs do not need to care about the fact that they are running in an EBCDIC environment. ASCII programs are able to access files on a local EBCDIC file system or on a remote ASCII system without caring about conversion themselves. This is achieved with a sophisticated automatic conversion mechanism which is transparent to the application programs (see Figure 14). The mechanism is based on file tagging. Files which are located in the z/OS hierarchical file system (HFS) or in zFS can be tagged. The file tag consists of two parts:

CCSID (Coded Character Set ID)
>	This is the code page the file was written in, for example 819 (ISO8859-1, ASCII) or 1047 (IBM-1047, EBCDIC)

TXTFLAG
>	This indicates whether a file is pure text and therefore can be converted safely to another code page. If text and binary data are mixed, the TXTFLAG must be off.

The USS command **ls** supports an additional option **-T** to show the file tag. The new command **chtag** is used to manually set or change a file tag. For compatibility, untagged files are considered to be EBCDIC files.

When the file tag CCSID is different from the program CCSID, the following table shows whether automatic conversion is performed:

Table 6. Automatic conversion as a function of program type and file tag

Program type	File tag	Automatic conversion?
EBCDIC	Untagged	No
EBCDIC	ASCII/TEXT	Yes
EBCDIC	ASCII/MIXED	No
ASCII	Untagged	Yes for SAP programs. In general, it is application-dependent. No if file is open in binary mode.
ASCII	EBCDIC/TEXT	Yes
ASCII	EBCDIC/MIXED	No

Automatic conversion, or autoconversion, is enabled if the environment variable *_BPXK_AUTOCVT* is set to ON, or system-wide by specifying AUTOCVT(ON) in the BPXPRMxx parmlib member. Thanks to autoconversion, files do not need to be converted—they just need to be tagged correctly. The application that creates an ASCII file is responsible for the correct tagging of the file. File tagging is also supported by NFS, DFS, SMB, and FTP.

Enhanced ASCII setup on z/OS: To ensure the desired behavior, the following setup is required to use the SAP J2EE Engine on z/OS. This setup will be mandatory for SAP Web Application Server releases after 6.20 and it is also necessary when using the SAP 6.40 downward compatible kernel with your SAP 6.20 system. The setup is downward compatible with previous SAP releases.

- The _BPXK_AUTOCVT variable controls the autoconversion, while the _TAG_REDIR_* variables control the tagging if standard input/output/error is redirected with <, >, or |. Therefore

  – Include the following lines in the .profile file in the home directory of the <sapsid>adm user:

    ```
    export _BPXK_AUTOCVT=ON
    export _TAG_REDIR_IN=TXT
    export _TAG_REDIR_OUT=TXT
    export _TAG_REDIR_ERR=TXT
    ```

  – Include the following lines in the .cshrc file in the home directory of the <sapsid>adm user:

    ```
    setenv _BPXK_AUTOCVT ON
    setenv _TAG_REDIR_IN TXT
    setenv _TAG_REDIR_OUT TXT
    setenv _TAG_REDIR_ERR TXT
    set _TAG_REDIR_IN=TXT
    set _TAG_REDIR_OUT=TXT
    set _TAG_REDIR_ERR=TXT
    ```

  Note: The ″=″ sign is required for 'export' and 'set' but is not valid for 'setenv'.

- As before, SAP files that are physically located in the z/OS hierarchical file system (HFS) or zFS can be stored in EBCDIC code page IBM-1047. Alternatively, SAP files can be stored in ASCII code page ISO8859-1; in this case they must be tagged as ISO8859-1/TEXT.

- For file systems shared between z/OS and ASCII systems, NFS tagging must be used instead of the fixed conversion used in the past. Furthermore, autoconversion must be switched on (refer to "NFS setup" on page 54).

SAP system files physically located on ASCII UNIX, Linux, or Windows systems are stored in ASCII code page ISO8859-1.

Files used by the SAP J2EE Engine are stored in ASCII and tagged accordingly.

Note: This also applies to files with extensions .jar, .zip, .class, .java, etc., because Java on z/OS expects binary files to be in ASCII and SAP tools need these files to be correctly tagged.

The ASCII code page ISO8859-1 and the corresponding EBCDIC code page IBM-1047 contain the same character set, namely Latin-1. Data can therefore be converted back and forth, and a round trip results in an identical file. For SAP kernel releases through 6.20, and ASCII/EBCDIC converter (fomeliba.dll) has been linked into the SAP Web application server and SAP utilities on z/OS. Newer SAP kernels fully exploit the Enhanced ASCII runtime, and the converter becomes obsolete.

Downward compatibility: The described setup is downward compatible with previous SAP releases. If files or file systems (such as NFS mount points) are shared between previous SAP systems and the J2EE server, autoconversion must be enabled for the old SAP systems as well. This is achieved by setting the environment variable in the `<sapsid>adm` profiles as previously described.

Therefore, even if you are not using the SAP 6.40 downward compatible kernel or the SAP J2EE Engine on z/OS, this setup is *strongly recommended* in order make a switch to these components easier. Under these conditions, however, the setup as described in the previous edition of this book will continue to work.

ASCII/EBCDIC converter for SAP kernel release 6.20: For SAP kernel release 6.20 and SAP utilities based on that release, you still need the ASCII/EBCDIC converter for all file input and output. Since the converter is shipped as part of ICLI, you must install the ICLI PTF as indicated in SAP Note 81737 for what is referred to as the 'LIBASCII release' in that note. The converter is a dynamic link library named fomeliba.dll, which is installed in the ICLI installation directory /usr/lpp/icli/sbin. A symbolic link to this DLL is established in directory /usr/lib during the installation process. The SAP installation tool and R3up adapt the profiles .dbenv_<hostname>.csh and .dbenv_<hostname>.sh in order to guarantee that the environment variable LIBPATH will contain the ICLI installation directory within the environment of the user ID <sapsid>adm.

SAP directory structure

As part of the installation, you have to define file systems. The file system and directory structure for the SAP system are described in the document *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*. This document also describes how to set up a heterogeneous UNIX environment. This applies, for example, if the central instance runs on z/OS while dialog instances are on AIX, Linux, or Solaris.

The file systems are to be allocated with at least the sizes specified in the SAP installation guide mentioned above. The sizes are checked by the SAP installation tool. If a file system needs to be extended, the following command can be used:

```
confighfs -x <size_MB>M <pathname>
```

Superuser authority is required for this command. In the case of a multi-volume file system, use the option *-xn* instead of *-x*.

SAP recommends that some directories be shared among all application server instances of an SAP system. These directories are /usr/sap/<SAPSID>/SYS/global and /usr/sap/SAPSID/SYS/profile. They point to /sapmnt/<SAPSID>/global and /sapmnt/<SAPSID>/profile, which are usually located on the central instance, too. Instances running on the same UNIX/Linux platform as the central instance can also share the executables in directory /sapmnt/<SAPSID>/exe. If different UNIX/Linux platforms are mixed, it is suggested that multiple exe directories in the file system be created on the central instance, one for each platform, e.g.: /sapmnt/<SAPSID>/AIX/exe and /sapmnt/<SAPSID>/OS390/exe. The appropriate directory is to be mounted on the dialog instance.

Selecting the network file system type

Shared HFS is the recommended method for sharing file systems between z/OS systems.

NFS is needed for sharing files among z/OS and other UNIX(-like) systems. This allows z/OS to take the role of the NFS server or client, depending on the physical location of the files. NFS is also needed to access CD-ROMs mounted on a workstation or PC.

SMB is needed when files are to be shared between z/OS and Windows. SMB lets z/OS behave like a Windows file server. It is an IBM product for z/OS that is equivalent to the freeware Samba, which does not support file tagging.

Shared HFS

The shared HFS feature, which was introduced in OS/390 V2R9, allows sharing of hierarchical file systems among z/OS systems in a Parallel Sysplex. For each mounted file system, one system becomes the owner while the other systems ship function requests to the owning system.

Note: The term *shared HFS* is somewhat confusing, because it seems to imply that it is related to the HFS and only the HFS. However, the Shared HFS is a logical layer above the physical file system implementation. All available file system implementations, i.e., HFS, zFS, NFS (the client), and TFS (the temporary file system), are supported as physical file systems.

When using the SAP application server in a sysplex, it is recommended to define the global SAP directories on a shared file system which is mounted with the option AUTOMOVE. In particular, the shared directories are /usr/sap, /usr/sap/<SAPSID>/SYS, /sapmnt, and /usr/sap/trans.

Example for /sapmnt/C11:

```
MOUNT FILESYSTEM('OMVS.&SYSPLEX.LOCAL.SAPMNT.C11')
MOUNTPOINT('/sapmnt/C11')
TYPE(HFS)
MODE(RDWR)
  AUTOMOVE
```

For performance reasons, the directories used during SAP installation and release upgrade, and the instance directories, must be mounted locally, that is, the local system must be the owner of the file system. We suggest setting up an additional mount point (/$SYSNAME/sapmnt) for locally owned file systems and defining the instance directory /usr/sap/<SAPSID>/<INSTNAME> as a symbolic link to a subdirectory of /$SYSNAME/sapmnt. This approach allows you to use the same instance name and number for application server instances on different z/OS systems while using shared file systems.

Notes:

1. It is recommended that the $VERSION file system be mounted read-only. This file system is referred to by */usr* and contains all system files. For file systems that are mounted read-only, the system attempts to access them locally. Cross-system messaging, which occurs for read/write mounted file systems, is thereby avoided.

2. If you intend to run the NFS client in conjunction with the shared HFS feature, see APARs OW43826 and OW48709 for special setup considerations and restrictions.

NFS setup

For the NFS client on z/OS, specify parameter *DISABLELLA*`(Y)`. This parameter is coded in the BPXPRMxx parmlib member as described in "NFS client configuration" on page 45.

NFS mounts between z/OS and an ASCII system must use the new autoconversion concept.

The file tagging function of the NFS server assumes that the Unicode Conversion Services are installed and activated on the system. The following operator command displays the active conversion tables:

```
D UNI,CONV

CONVERSION: 01047-00819-LRE
            00819-01047-LRE
```

For complete documentation on NFS setup, refer to *z/OS Network File System Customization and Operation* and *z/OS Network File System User's Guide*.

- **NFS client on z/OS (EBCDIC), NFS server on UNIX or Linux:** If an ASCII file system is mounted on z/OS the `TAG(TEXT,819)` operand is used:

```
MOUNT ... PARM('...'),TAG(TEXT,819)
```
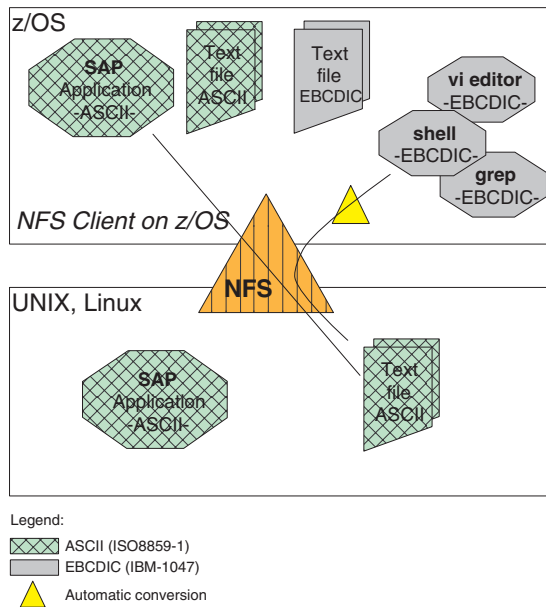
Figure 15. Automatic ASCII-EBCDIC conversion with NFS client on z/OS

In this case, the conversion is done by z/OS UNIX System Services, not the NFS client.

- **NFS server on z/OS, NFS client on ASCII UNIX or Linux:** If a z/OS file system is mounted on an ASCII system (e.g., AIX, Linux) the following mount parameter is used:

```
TEXT,cln_ccsid(819),srv_ccsid(1047)
```
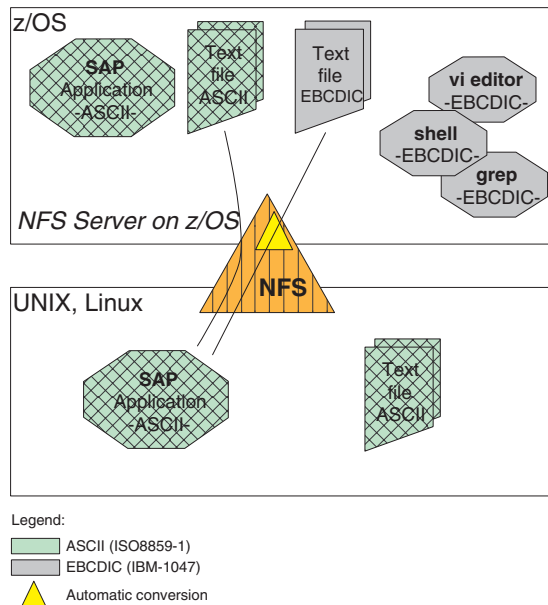
Figure 16. Automatic ASCII-EBCDIC conversion with NFS server on z/OS

In this case, the conversion is done by the NFS server.

- **NFS server and client on z/OS:** If a z/OS file system is mounted on another z/OS system (i.e., NFS mounts between LPARs), the following mount operands are used:

```
MOUNT ... PARM('...,TEXT,cln_ccsid(819),srv_ccsid(1047)'),TAG(TEXT,819)
```

The TEXT keyword and the ccsid specifications are appended to the pathname and sent to the NFS server, while the TAG operand is processed locally by the NFS client. On z/OS, the NFS attributes of a mounted file system can be displayed using the command:

```
usr/lpp/NFS/showattr <mountpoint>
```

Example of a TSO mount command:

```
MOUNT FILESYSTEM('NFS001')
      TYPE(NFS)
      MODE(RDWR)
      MOUNTPOINT('/usr/sap/trans')
      PARM('aix01:/usr/sap/trans')
      TAG(TEXT,819)
      NOSECURITY
```

The following example shows two mount definitions in the mount map to be processed by the z/OS automounter:

- The first mount definition is for an AIX file system and specifies ASCII/EBCDIC translation.
- The second mount definition specifies a file system on a different LPAR.

```
name            usr_sap_trans
type            NFS
filesystem      NFS001
mode            rdwr
duration        30
delay           30
parm            aix01:/usr/sap/trans
tag             text,819
security        no

name            sapmnt_C11
type            NFS
filesystem      NFS002
mode            rdwr
duration        30
delay           30
parm            mvs01:/HFS/sapmnt/C11,TEXT,cln_ccsid(819),srv_ccsid(1047)
tag             text,819
security        no
```

Sample mount command on AIX:

```
mount 'mvs01:/HFS/sapmnt/C11,text,cln_ccsid(819),srv_ccsid(1047)' /sapmnt/C11
```

Security considerations: The NFS server attribute `SECURITY(EXP,EXP,EXP)` means that normal UNIX-type security applies. If `SECURITY(SAF,SAF,SAF)` is specified, the command **mvslogin** must be executed and the z/OS password typed in before the files can be accessed.

SAF security is needed to gain root authority when installing the Web application server on z/OS. It is not needed during normal SAP operation. In fact, SAF security must not be used when running the IBM High Availability Solution for SAP as described in *High Availability for SAP on zSeries Using Autonomic Computing Technologies* (SC33–8206).

> **Example:**
> Assume that the customer's SAP file system is on z/OS and he would now like to install a dialog instance on AIX. To do this, he must invoke the SAP installation tool as the root user. In order for the SAP installation tool to access the files, the customer must first issue the command **mvslogin** (thus obtaining UID=0 authorization on z/OS; a warning message about mismatching GID can be ignored). During the installation, the SAP installation tool creates the AIX user ID `<sapsid>adm` and switches to that user ID, which leads to a permission error when accessing z/OS files. The AIX user `<sapsid>adm` must now execute **mvslogin** to gain the appropriate authorization before the SAP installation tool is able to continue. In this case, the UID and GID of the AIX and z/OS user ID must match.
>
> If, however, we assume that the shared files reside on an AIX file system and they are to be accessed from z/OS, security is controlled by the export file on AIX. In this case, there is no need to perform a log-in before attempting to access the files; however, the UID and GID must match. Because the installation process runs under superuser authority, the exported file systems must allow root access.

SMB setup

The SMB protocol is integrated into the operating system. This feature provides functionality similar to that offered by Samba. Refer to *z/OS Distributed File Service SMB Administration* and *z/OS Distributed File Service Administration*.

The following description and samples have not been tested on z/OS releases prior to V1R3. For such releases, the definitions, commands, or table names may differ. For information pertaining to these releases, refer to *z/OS Distributed File Service SMB Administration* for the appropriate z/OS release.

In order to be able to share data between z/OS and Windows systems, you have to perform the following task:

- **Defining the shared directories and the file systems to be exported**

  Set up the following SMB tables, which are usually located in /opt/dfslocal/var/dfs:

  - smbtab to define the shared directories
  - devtab and dfstab to define the file systems to be exported.

  For details, refer to the examples below and to the chapter "Sharing Files" in *z/OS Distributed File Service SMB Administration*.

  For the examples presented below, the following assumptions are made:

z/OS HFS data set name	UNIX System Services mount point
OMVS.FCSPLEX.LOCAL.SAPMNT	/sapmnt
OMVS.FCS2.LOCAL.SAP	/usr/sap
OMVS.FCSPLEX.LOCAL.SAPMNT.EZ1	/sapmnt/EZ1
OMVS.FCS2.LOCAL.SAP.EZ1	/usr/sap/EZ1

  Example of an smbtab table:

```
****************************** Top of Data ******************************
/dev/ufs3  SAPMNT ufs "/USR/SAP" r/w 0 /
/dev/ufs4  SAPMNTEZ1 ufs "/SAPMNT/EZ1" r/w 0 /
****************************** Bottom of Data ****************************
```

  Example of a devtab table:

```
****************************** Top of Data ********************************
define_ufs 2
OMVS.FCSPLEX.LOCAL.SAPMNT TEXT
define_ufs 3
OMVS.FCS2.LOCAL.SAP TEXT
define_ufs 4
OMVS.FCSPLEX.LOCAL.SAPMNT.EZ1 TEXT
define_ufs 5
OMVS.FCS2.LOCAL.SAP.EZ1 TEXT
****************************** Bottom of Data *****************************
```

  Example of a dfstab table:

```
****************************** Top of Data ******************************
/dev/ufs2 hfs2 ufs 2 0,,2
/dev/ufs3 hfs3 ufs 3 0,,3
/dev/ufs4 hfs4 ufs 4 0,,4
/dev/ufs5 hfs5 ufs 5 0,,5
****************************** Bottom of Data ***************************
```

- **Logon considerations**

  To ensure that an application server can access shared resources without user intervention, SMB supports two different methods of user authentication without

prompting the user for a password (for details, refer to "Sharing Files, Logon considerations" in *z/OS Distributed File Service SMB Administration*):

– RACF DCE Segments for SMB encrypted password support:

To enable the SMB server to use encrypted passwords, you must set up the user ID <sapsid>adm and the SAP installation user ID <root> on z/OS for SMB encrypted password support:

1. Activate the RACF class KEYSMSTR:

   `SETROPTS CLASSACT(KEYSMSTR)`

2. Define a DCE.PASSWORD.KEY entry in the class KEYSMSTR:

   `RDEFINE KEYSMSTR DCE.PASSWORD.KEY SSIGNON(KEYMASKED(nnnnnnnnnnnnnnnn))`

3. Define the RACF DCE segment for the user IDs <sapsid>adm and <root>:

   ```
   ALTUSER <sapsid>adm DCE
   ALTUSER <root> DCE
   ```

4. Log on as <sapsid>adm and issue the following z/OS UNIX System Services command:

   `smbpw <login_password> <login_password>`

   where <login_password> is the password of <sapsid>adm on Windows.

   Note: This step must be repeated whenever the password is changed on Windows.

5. Log on as <root> and issue the following z/OS UNIX System Services command:

   `smbpw <administrator_password> <administrator_password>`

   where <administrator_password> is the password specified for the SAP installation user ID (<administrator>) that has administrator rights on Windows.

   Note: This step must be repeated whenever the password is changed on Windows.

6. Enable the SMB server to use encrypted passwords by setting the following environment variable for the server process DFSKERN in file envar, which is normally located in /opt/dfslocal/home/dfskern:

   `_IOE_SMB_CLEAR_PW=NOTALLOWED`

– Using passthrough authentication

Passthrough authentication permits the SMB server to use a domain controller to authenticate a client trying to log on to the SMB server. The SMB server will log on to the domain controller using the account name (user ID) and the challenge response from the client. If the logon to the domain controller is successful, the client is allowed to establish a session with the SMB server and the PC user ID will be mapped to a local (RACF) user ID. For details see APAR II13046.

• **Mapping SMB user IDs to z/OS user IDs**

ID mapping entries must be created in table smbidmap for the following user IDs:

– SAPService<SAPSID> must be mapped to <sapsid>adm

– <sapsid>adm must be mapped to <sapsid>adm

– the user ID <administrator> must be mapped to the user ID <root>

Example of smbidmap table definitions:

```
******************************* Top of Data ********************************
SAPServiceEZ1
ez1adm

ez1adm
ez1adm

administrator
sup8
****************************** Bottom of Data ******************************
```

For details on how to specify the entries in smbidmap, refer to "Mapping SMB user IDs to z/OS user IDs" in *z/OS Distributed File Service SMB Administration*.

The following environment variable for DFSKERN must point to the mapping file:

```
_IOE_SMB_IDMAP=<path_of_mapping_table>/smbidmap
```

If dfs is already started, issue the following command on z/OS to reload the user ID mapping table smbidmap:

```
modify dfs,send dfskern,reload,smbmap
```

- **Setting the environment variables for the process dfskern**

  The following environment variables must be specified in envar, which is normally located in /opt/dfslocal/home/dfskern:

  ```
  _IOE_SMB_IDMAP=<path_of_mapping_table>/smbidmap
  _IOE_SMB_CLEAR_PW=NOTALLOWED
  _IOE_SMB_COMPUTER_NAME=<tcp>
  _IOE_SMB_DOMAIN_NAME=<Windows_Workgroup>
  _IOE_MOVE_SHARED_FILESYSTEM=ON (if you are using shared HFS)
  _IOE_HFS_FILETAG=QUERY
  _IOE_DYNAMIC_EXPORT=ON
  _IOE_PROTOCOL_SMB=ON
  _IOE_PROTOCOL_RPC=OFF
  _IOE_EXPORT_TIMEOUT=OFF
  ```

  Example of an envar table:

  ```
  ******************************* Top of Data ********************************
  _IOE_SMB_COMPUTER_NAME=fcs2vipa
  _IOE_SMB_DOMAIN_NAME=WORKGROUP
  _IOE_SMB_IDMAP=/etc/smbidmap
  _IOE_MOVE_SHARED_FILESYSTEM=ON
  _IOE_HFS_FILETAG=QUERY
  _IOE_DYNAMIC_EXPORT=ON
  _IOE_EXPORT_TIMEOUT=OFF
  _IOE_PROTOCOL_SMB=ON
  _IOE_PROTOCOL_RPC=OFF
  _IOE_SMB_CLEAR_PW=NOTALLOWED
  _IOE_SMB_ABS_SYMLINK=ON
  _IOE_MVS_IDMAP_SAF=NO
  _IOE_MVS_SERVER=DFS
  #_IOE_MVS_DFSDFLT=DFSDFLT   don't specify else default and no prompting
  _IOE_DFS_MODIFY_PATH=/opt/dfslocal/home/dfcntl/modify.rendezvous
  _IOE_MVS_IDMAP=/opt/dfslocal/home/dfskern/mapid.output
  _IOE_LFS_SYNC_INTERVAL=30
  _EUV_SVC_MSG_LOGGING=CONSOLE_LOGGING
  DCE_START_SOCKET_NAME=/opt/dfslocal/home/dfcntl/ioepk.soc
  TZ=MEZ-1MES
  _EUV_AUTOLOG=NO
  NLSPATH=/usr/lib/nls/msg/%L/%N
  LANG=En_US.IBM-1047
  LIBPATH=/usr/lib:/usr/lpp/ocsf/lib
  ****************************** Bottom of Data ******************************
  ```

- **Starting the SMB server**

When all of these tasks have been completed successfully, issue following command on z/OS to start the Distributed File System (DFS):

```
start dfs
```

Mounting CD-ROMs

The SAP code and database are delivered on CD-ROMs. The contents must be accessible by the SAP installation tool and the upgrade program **R3up**. This can be achieved either by mounting a CD-ROM drive of a workstation on z/OS via NFS, or by copying the contents to the z/OS file system.

Using a CD-ROM drive on a UNIX/Linux system: On AIX 5.1 or higher you can mount the CD-ROM with option `-o upcase` in order to have all file names in upper case.

The CD-ROM file system is to be exported for NFS. Make sure that an NFS server is running on the UNIX/Linux system.

On z/OS, mount the CD-ROM file system from the UNIX/Linux system using the *TAG* option as described in "NFS setup" on page 54.

If multiple CD-ROM drives (or a jukebox) are available, several CD-ROMs can be mounted at the same time.

The CD-ROM file system is not case sensitive; which means that the files can be accessed in upper- or lowercase. The SAP utilities always use uppercase names.

Optionally, copy the CD-ROMs to a local z/OS file system. For each CD-ROM, allow for 600 MB space. Before you copy a CD-ROM, make sure that it is mounted on the UNIX system such that files and directories are listed in uppercase characters:

- On AIX, use mount option *-o upcase*, for example,

  ```
  mount -o upcase -r -v cdrfs /dev/cd0 </CD-mountdir>
  ```

  where `</CD-mountdir>` specifies the mount point of the CD-ROM.
- On Solaris, use mount option *-o ro,nomaplcase*, for example,

  ```
  mount -F hsfs -o ro,nomaplcase /dev/dsk/c0t6d0s2 </CD-mountdir>
  ```

  where `</CD-mountdir>` specifies the mount point of the CD-ROM.

Issue the following commands to copy the entire CD-ROM:

```
cp -Rp </CD-mountdir> </targetdir>
chmod -R 777 </targetdir>
```

where `</CD-mountdir>` specifies the local or NFS mount point of the CD-ROM and `</targetdir>` specifies the path of the target directory.

Using a CD-ROM drive on a Windows system: When using a CD-ROM drive on a Windows system, the following two alternatives are available for using the SAP installation tool or **R3up** on z/OS:

- NFS server
- SMB

These two alternatives are described in the following two sections.

NFS server: The Windows system must have third-party software installed that provides the functionality of an NFS server. The z/OS NFS client must be configured. For a description of the configuration, refer to *z/OS Network File System Customization and Operation*. The CD-ROM drive is to be exported on the Windows system and mounted on z/OS.

To mount the CD-ROM, use the TSO **MOUNT** command.

For a description of the setup, see "NFS setup" on page 54.

SMB: For z/OS software requirements and a description of the setup, see "SMB setup" on page 58.

When using SMB, the CD-ROMs cannot be accessed directly by z/OS. Instead, the contents must be copied to the z/OS file system.

Provide a directory structure on z/OS for a copy of the CD-ROMs. For each CD-ROM, allow for 600 MB space. Define the directories as network share.

On the workstation, map the share as a network drive and copy the CD-ROM using the command:

```
xcopy <cdrom> <dest> /s /e
```

where `<cdrom>` is the CD-ROM drive and `<dest>` specifies the destination network drive and path.

Transferring files with FTP

z/OS file tagging is also supported by FTP. Tagging and automatic conversion is enabled by using the SBDATACONN (abbreviated as SBD) parameter. This parameter is specified with an FTP **site** or **locsite** command. If you invoke the FTP command on an ASCII system (such as AIX) and connect to z/OS, enter the following command:

```
site sbd=(iso8859-1,iso8859-1)
```

Or, if you invoke FTP on z/OS for data transfer to an ASCII system, use the **locsite** command instead of **site**:

```
locsite sbd=(iso8859-1,iso8859-1)
```

Transferring SAP files from an ASCII system (such as UNIX, Linux, Windows) to z/OS: When the SBDATACONN parameter is specified, files transferred to z/OS remain unconverted and are tagged as ISO8859-1/TEXT.

Note: If you specify the FTP option BINARY, the files are also transferred without any conversion, but they are tagged as BINARY. This excludes them from the automatic conversion. You can change and correct the tag afterwards.

Transferring SAP files from z/OS to an ASCII system: When the SBDATACONN parameter is specified, a file tag is recognized and the data is converted from the code page of the file to ISO8859-1. If the file tag indicates an ASCII file, the data is transferred without conversion. The following FTP message informs the user about the file tag found and the respective conversion:

```
125-Tagged EBCDIC file translated with table built using file system cp=IBM-1047,
    network transfer cp=iso8859-1
```

The C-shell, tcsh, and Korn shell

In the SAP system environment, the Berkeley UNIX C-shell is used as the default shell. The first line of a shell script tells the system which shell is to be used; C-shell scripts begin with the following line:

```
#!/bin/csh -f
```

C-shell logon profiles

The C-shell reads the following four profiles:

- `/etc/csh.cshrc`
- `/etc/csh.login`
- `$HOME/.cshrc`
- `$HOME/.login`

When the C-shell is invoked with the **-c** option, the files `/etc/csh.login` and `$HOME/.login` are not processed because this is not a "real" user login. In particular, this means that the profiles are not processed when a shell script or program is invoked via BPXBATCH in a started task or via the System Automation command INGUSS. Therefore, make sure that all relevant settings needed for the startup of the SAP system are defined in the profiles `/etc/csh.cshrc` and `$HOME/.cshrc`.

Creating a link for tcsh

An enhanced but compatible version of the C-shell (tcsh) is provided as part of UNIX System Services. For more information on tcsh, refer to *z/OS UNIX System Services Command Reference*.

To use tcsh for C-shell scripts, the following link must be established. You must be superuser to perform this command, and the file system in which `/bin` is located must be mounted in the read-write mode:

```
ln /bin/tcsh /bin/csh
```

Creating a link for the Korn shell

Additionally, we also suggest setting up a link for the Korn shell (ksh); this is because UNIX shell scripts may refer to it:

```
ln /bin/sh /bin/ksh
```

Java on z/OS

To run any Java program written by SAP, such as the Upgrade Assistant, the IBM Developer's Kit for z/OS, Java 2 Technology Edition at SDK level 1.4.1 is required. Java is installed in the following directory:

```
/usr/lpp/java/J<java_version>
```

where `<java_version>` is the version number of the JDK installed on your system, for example:

```
/usr/lpp/java/J1.4
```

The directory /usr/lpp/java/J<java_version>/bin must be included in the *PATH* variable. The availability of Java can be checked using the following command:

```
java -version
```

Setup of a heterogeneous SAP system

Choosing consistent time settings

Time settings on the database server

IBM recommends running the z/OS system clock with GMT (Greenwich Mean Time) and setting up the time zone and daylight saving time information properly. If you run the z/OS system clock with GMT, it is not necessary to stop z/OS or DB2 when the standard time is switched to daylight saving time or back, and the z/OS system clock does not have to be changed.

Time settings on the application servers

The system time on all application servers should be GMT (sometimes also called UTC, ″Universal Time Coordinated″) as well. Furthermore, the time zone definitions should be the same on all systems. If the time settings differ, the timestamp of files on NFS and the time displayed in SAP dialogs could be off by one or more hours and remote command execution might be impacted.

On all involved systems, log on as `<sapsid>adm` and verify the time settings:

- Compare the time zone definitions. The actual names of the time zone can be different, what matters are the offset to GMT and the optional settings of daylight saving time:

  `echo $TZ`
- Compare system time (GMT or UTC):

  `date -u`
- Compare local time:

  `date`

For details on setting the time zone for the ICLI server, see the description of the environment variable TZ (Appendix C, "ICLI environment variables," on page 157). For recommendations on how to handle the SAP system during a switch to daylight saving time and back, refer to SAP Note 353529 and related SAP Notes.

Chapter 4. Setting up instances of the ICLI server

The ICLI server and client software is part of the scope of delivery of z/OS. While the client software is automatically transferred to its execution directory on the remote application server host and set up by the SAP installation tool during the installation of the SAP system, the ICLI server component has to be set up manually. The tasks that need to be completed before the remote SAP application servers are installed are described in the following sections.

> **Note:**
> Whenever an ICLI PTF is installed on z/OS after SAP system installation, make sure that you follow the manual procedures for re-binding the ICLI server and downloading the ICLI client described in Chapter 8, "Applying service to ICLI and the SAP application server on z/OS," on page 133.

Prerequisites

The following prerequisites must be satisfied before you can set up an ICLI server instance:

- z/OS user ID's have been created as described in Chapter 2, "z/OS security, group IDs and user IDs," on page 23.
- The user ID <SAPSID>ICLI has READ permission on the HFS files or z/OS data sets used as environment files, if any.
- TCP/IP support has been set up as described in the *Connectivity Guide*.
- The DB2 subsystem is set up as described in the *SAP installation guides*.
- The appropriate PTF service level for the ICLI is installed on z/OS. Check SAP Note 81737 for the APAR number of the current version of the ICLI.

  If necessary, install the correct PTF service level on z/OS and the application server hosts (see SAP installation note 493836 for important installation-related information).
- The required z/OS UNIX System Services parameters have been activated as described in "Configuring the ICLI server and SAP application server on z/OS" on page 40.
- The SAP installation tool executes the necessary DB2 BIND and GRANT jobs for the default plan (for the name of the plan, refer to the *SAP installation guides*). If you will not be using this plan in your start procedure for the ICLI server, you will have to perform DB2 binding as described in "DB2 binding for the ICLI server" on page 135.

Running the ICLI server setup program

The easiest way to set up an ICLI server instance is to use the setup tool `fome620i` (`fome640i`). Using the parameters you specify when you invoke it, it will perform the following tasks:

- It creates the so-called instance working directory (or "working directory") in the HFS, into which the ICLI server instance writes its message and trace files and where its start parameters are kept.

- In the instance working directory, it creates the following start procedure templates, which you can use to start the server instance or tailor them to your specific requirements:
  - the UNIX System Services start script `iclistart`.
  - the JCL batch job `icli.jcl`.
  - the JCL job `iclitask.jcl` and the environment file `iclienv`, in which the start parameters for this started task are defined.

To set up multiple instances of the ICLI server, you can either run the setup tool repeatedly or run it just once and create the working directories for the other instances manually and copy and edit the start procedure you want to use accordingly.

The setup tool `fome620i (fome640i)` is located in the directory `/usr/lpp/icli/samples`. To run the instance setup tool, invoke it with the following command line options (example shown for 6.20):

```
►►── fome620i─┬─<working_dir>──<steplib>──<plan>──<conn_port>──[- q]─┬──────────────►◄
                                                                     └─<options>─┘
```

<working_dir>
> The working directory for the new instance. It is recommended that you use a directory under the HFS home directory of `<SAPSID>ICLI` as the working directory.

<steplib>
> The DB2 load library `DB2_SDSNLOAD` where the DB2 modules reside that are needed by the ICLI server.

<plan>
> The plan name for the new instance.

<conn_port>
> The port number of the connection port for the new instance. For details on selecting the connection port for an ICLI server, see "Considerations for selecting an ICLI connection port" on page 99.

-q   Quiet, display no informational messages.

<options>
> Command line options, zero or more of the following:
> - *-help*: Display help text.
> - *-?*: Display help text.

Example:
```
fome620i  /usr/icli/CUS_1 SYS1.SDSNLOAD CUS1PLAN 5001
```

Configuring the start procedures for the ICLI server

There are three ways to start the ICLI server:
1. Starting a JCL job or procedure as a started task.
2. Submitting a JCL job to batch.
3. Executing a UNIX System Services shell script.

If you have used the setup program `fome620i` (`fome640i`) to set up an ICLI server instance, template files for these start procedures have been created in the instance working directory. You can tailor them to your specific requirements as described in the following sections.

In the start file templates, the variable *ICLI_TRACE_LEVEL* set to its default value (0) is provided as an example. Other ICLI server environment variables would be specified in the same way, each on a separate line. The complete list of ICLI server environment variables can be found in "ICLI server environment variables" on page 159.

If you are not using the start procedure templates to start the server but want to create you own, you should ensure that:

- the messages written to STDERR and STDOUT are redirected to files in the working directory as shown in the sample template files `iclitask.jcl` or `icli.jcl` below. There, the output is redirected to the files `icliserv.out` and `icliserv.err`.
- the environment variable TZ is set to your local time zone in your started task or JCL batch job, so that the timestamps of the ICLI server messages and the ICLI client messages will match.

Note:

If you want to be able to invoke the smooth stopping method for the ICLI server from *UNIX System Services*, you should check if the default values of the two environment variables determining the smooth stop behavior meet your requirements, because they need to be set before the ICLI server is started (see also "The smooth stopping method" on page 106).

Configuring the JCL started task

If you plan to start the ICLI server as a started task, you can use the JCL job template `iclitask.jcl` and the template for its environment file `iclienv`, which were created by the setup tool in the instance working directory, and tailor them to your requirements (for example, you will have to add a valid job card).

To create the started task and its environment file shown in the examples below, the setup tool was invoked with the following command line options (example shown for 6.20):

```
fome620i /usr/iclirun01/CUS_1 SYS1.SDSNLOAD CUS1PLAN 5001
```

Following is the JCL job `iclitask.jcl` created by the setup tool:

```
//* ICLI server started task JCL
//ICLI    EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=0M,
//        PARM='PGM /usr/sbin/fome620s -PLAN CUS1PLAN -LOGDIR /usr/icli
//           run01/CUS_1 -PORT 5001'
//STDENV  DD PATH='/usr/iclirun01/CUS_1/iclienv'
//STEPLIB DD DISP=SHR,DSN='SYS1.SDSNLOAD'
//STDERR  DD PATH='/usr/iclirun01/CUS_1/icliserv.err',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRWXU)
//STDOUT  DD PATH='/usr/iclirun01/CUS_1/icliserv.out',
//        PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//        PATHMODE=(SIRWXU)
//SYSUDUMP DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

The environment file `iclienv` is defined as follows:

```
ICLI_TRACE_LEVEL=0
STEPLIB=SYS1.SDSNLOAD
TZ=MEZ-1MES,M3.5.0,M10.5.0
```

Edit the files to alter or add start parameters or environment variables.

Since the JCL *PARM* parameter has a limit of 100 characters, use the ICLI server environment variable *ICLI_PARAMETERS* to specify additional command line parameters in file `iclienv`.

The ICLI server environment variables are described in detail on page 159; the ICLI server command line options are listed on page 101.

In the JCL job, the *REGION* size is set to 2 GB (`REGION=0M`). For more information on this parameter, see "Maximum number of threads" on page 47.

To start the ICLI server instance with the started task:

1. Copy `iclitask.jcl` into a member of a partitioned data set (PDS) that is in your STC PROCLIB concatenation.

   If you run multiple ICLI server instances, use one PDS member for each instance.

   The current directory of the ICLI server process is the home directory of the user ID `<SAPSID>ICLI`.

   Access rights of the started task:
   All access rights of the started task are based on the user ID `<SAPSID>ICLI` associated with the started task in RACF's `STARTED` class. If you have not done so already, associate the user ID with the started task by using RACF's `STARTED` class.
   ---

2. Start the procedure/job.

Configuring the JCL batch job

If you plan to start the ICLI server by submitting a JCL job to batch, you can use the JCL job template `icli.jcl`, which was created in the instance working directory by the setup tool, and tailor it to your requirements (for example, you will have to add a valid job card).

To create the JCL job shown in the example below, the setup tool was invoked with the following command line options (example shown for 6.20):

```
fome620i /usr/iclirun01/CUS_1 SYS1.SDSNLOAD CUS1PLAN 5001
```

The following depicts the JCL job `icli.jcl` created by the setup tool:

```
//* ICLI server JCL job
//* Hot Info:
//* - REGION=0M in job header results in a REGION size of
//* - 2GB. This is recommended as described in
//* - SAP Web Application Server 6.20 on IBM eServer
//* - zSeries: Planning Guide, Configuring z/OS UNIX System
//* - Services - Selecting the z/OS UNIX System Services
//* - Services parameters for the ICLI -
//* - Maximum number of threads - REGION size.
//ICLI    EXEC PGM=BPXBATCH,TIME=NOLIMIT,REGION=0M,
//         PARM='PGM /usr/sbin/fome620s -PLAN CUS1PLAN -LOGDIR /usr/icli
//              run01/CUS_1 -PORT 5001'
//STDENV  DD *
ICLI_TRACE_LEVEL=0
STEPLIB=SYS1.SDSNLOAD
TZ=MEZ-1MES,M3.5.0,M10.5.0
/*
//STDERR  DD PATH='/usr/iclirun01/CUS_1/icliserv.err',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRWXU)
//STDOUT  DD PATH='/usr/iclirun01/CUS_1/icliserv.out',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRWXU)
```

To customize the template, proceed as described for the JCL started task in section
"Configuring the JCL started task" on page 67. You start the ICLI server by
submitting the JCL job.

Configuring the UNIX System Services shell script

If you plan to start the ICLI server by executing a UNIX System Services shell
script, you can use the start script template iclistart that was created in the
instance working directory by the setup tool. You can either use it as is or adapt it
as needed.

To create the script shown in the example below, the setup tool was invoked with
the following command line options (example shown for 6.20):

```
fome620i /usr/iclirun01/CUS_1 SYS1.SDSNLOAD CUS1PLAN 5001
```

The following UNIX Systems Services start script iclistart is created by the setup
tool:

```
#!/bin/sh
# ICLI server start script
export ICLI_TRACE_LEVEL=0
export STEPLIB='SYS1.SDSNLOAD'
nohup /usr/sbin/fome620s -PLAN CUS1PLAN -LOGDIR /usr/iclirun01/CUS_1 -PORT 5001
2>/usr/iclirun01/CUS_1/icliserv.err 1>/usr/iclirun01/CUS_1/icliserv.out &
```

The defaults should allow you to run immediately and to use the smooth stopping
method with its default values (see also "The smooth stopping method" on page
106). The STDERR and STDOUT output is redirected to the instance working
directory.

If you need to alter or add start parameters or environment variables, edit the file
accordingly.

Since the shell script contains environment variables, it must not contain sequence
numbers.

The ICLI server environment variables are described in detail on page 159; the ICLI server command line options are listed on page 101.

To start the ICLI server instance with the start script:
- Switch to the UNIX System Services shell using the user ID <SAPSID>ICLI.
- Make the instance working directory the current directory.
- Execute the shell script by issuing:
  ```
  ./iclistart
  ```

Verifying the installation

In order to verify some of the setup steps performed so far you can use the following mechanism; it will allow you to rule out a number of configuration errors without having to start SAP:

1. If you are not using the default DB2 schema SAPR3, specify the environment variable *ICLI_SAP_SCHEMA*.
2. Temporarily add the command line option *-CHECK* <db2_subsystem> to your ICLI server start procedure (for details on *-CHECK*, see "ICLI server command line options" on page 101).
3. Start the ICLI server with your start procedure.

If the check was successful, a message similar to the following is written to STDERR:

```
This is DB2 version 7.
DB2 CCSIDs: ASCII 819, EBCDIC 500
FETCH for A2E translation table was OK.
FETCH for E2A translation table was OK.
Translation tables are OK.
Check for DB2 subsystem <E1E> and SAP SCHEMA <SAPR3> was OK.
ICLI server ended.
```

Using the option *-CHECK* allows you to verify in particular that:
- the ICLI server is installed and it can access its log directory
- DB2 modules can be loaded by the ICLI server
- the DB2 subsystem is up and accessible
- RRS is started
- the DB2 plan used by the ICLI server is accessible
- the ICLI server can set its SQLID to the specified schema
- the ICLI can use the Instrumentation Facility Interface of DB2 to query the DB installation parameters
- the ICLI can retrieve the codepage translation tables from DB2
- the codepage translation tables are bijective, i.e., the mapping is the same in both directions

If any of the above checks fails, you will receive an appropriate error message in the ICLI message log file. Once you have performed the check, you should remove the option *-CHECK* from your start procedure.

Chapter 5. Performance tuning considerations before installing an SAP system

The objective of performance monitoring and tuning is to achieve the goals set for the response times of SAP transactions, system throughput, and system availability. Setting acceptable and achievable goals and determining the optimal system resources necessary to meet the goals are very important factors in this equation. These capacity planning tasks need to be skillfully done before performance monitoring and tuning activities start. The IBM Competence Centers are the best source of information on capacity planning for SAP system environments. In the rest of the chapter, we assume that the system resources have been provided according to the official sizing guidelines.

There are a number of steps you should complete before SAP system installation. They are described in the subsequent sections. Note that many of them also apply during any stage of an SAP system's life cycle, so include them in your regular performance monitoring and tuning procedures.

For a detailed description of additional performance monitoring and tuning measures and basic WLM setup configurations that you can implement even in the preparation phase, see Chapter 6, "Performance tuning considerations," on page 77.

Ensuring proper z/OS and DB2 code levels

Staying current with the z/OS and DB2 maintenance levels is an important prerequisite for ensuring optimal SAP system performance. For more information on how to ensure proper z/OS and DB2 code levels, see "Checking the PTF status to ensure proper code levels" on page 33.

Selecting appropriate z/OS settings

ICF catalog performance

There are two major tuning knobs with respect to the ICF catalog performance: Global Resource Serialization and Catalog Address Space. The term "catalog" in this section refers to a VSAM catalog rather than the DB2 catalog.

Global Resource Serialization (GRS) is the strategic multi-system serialization solution for z/OS. The Catalog Address Space (CAS) is where the catalog operations are performed. The way you use these facilities significantly affects the catalog operations such as **data set locate**, **data set open and extend**, etc. Here are some general recommendations:

- Stay current with z/OS (more specifically, DFSMS) releases and maintenance.
- Set *RESMIL* to 1
- Do not specify GRS in IEAIPSxx
- Increase *CATMAX* above the number of active catalogs.
- Set *STRNO* between 5 and 7 for highly active catalogs.
- Place shared catalogs in the Catalog Data Space Cache.
- Remove *SYSZVVDS* and *SYSVTOC* from Reserve Conversion RNL. Place them in Exclusion RNL instead.

Further details on GRS and CAS would go beyond the scope of this document. For more information, refer to the following documents in the z/OS library:

- *z/OS MVS Planning: Global Resource Serialization*
- *z/OS DFSMS: Managing Catalogs*

Switching off unnecessary traces

- When running the application server, the SAP installation tool, or the SAP upgrade tool **R3up** on z/OS, it is recommended to disable SMF record type 92 for performance reasons. This SMF record type is used to monitor all UNIX file system activities. An active SAP system can produce millions of SMF records.
- Check which SMF record types are activated by looking at the SMFPRMxx member and turn off those not needed. Be sure to keep types 70-79 and 100-102 active. Type 30 SMF records can cause a significant overhead, especially if *DDCONS* is set to YES (which is the default). If you do not need these records, do not collect them. Otherwise, specify *DDCONS*(NO) in the corresponding SMFPRMxx member. You can also improve the DB2 shut down time if you specify *SUBSYS*(STC,NODETAIL) in the same member.
- Switch off unnecessary system component traces such as the IRLM trace. Use the system command **D TRACE** to check which system component traces are active.
- Make sure that the ICLI trace is at its default, i.e. *ICLI_TRACE_LEVEL*=0.
- Switch off the CICS and IMS monitoring facilities traces
- Do not specify DSNTRACE even if used with DD=DUMMY. Do not specify DSNRRSAF even if used with DD=DUMMY.

Ensuring optimal ICLI server performance

For performance reasons, the ICLI server tries to make its address space non-swappable. In order to enable this feature, you need to grant the user ID that starts the ICLI server (<SAPSID>ICLI) READ access to the profile BPX.STOR.SWAP of RACF's FACILITY class (see *z/OS UNIX System Services Planning*). If you have not yet defined such a profile of the FACILITY class, you may do so using the following RACF command:

```
RDEFINE FACILITY BPX.STOR.SWAP UACC(NONE)
```

Use the following two commands to grant <SAPSID>ICLI READ access to this profile:

```
PERMIT BPX.STOR.SWAP CLASS(FACILITY) ID(<SAPSID>ICLI) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

If you choose not to grant <SAPSID>ICLI this READ access, ICLI message ICLS1078W will be issued. The ICLI server is fully functional but will not benefit from the additional performance improvement.

Checking the WLM setup

WLM application environments are required and a WLM service definition must be created and installed.

You must check the WLM setup of your system and determine which service classes apply before installing the SAP system or loading or upgrading the SAP database.

> **Caution:**
> An incorrect configuration will have a considerable impact on system performance.

A detailed description of WLM service definitions for the ICLI server and SAP application server on z/OS is provided in "Workload Management (WLM) — Classification of SAP workload" on page 82, where you will also find a sample WLM startup configuration for the ICLI server. You can use this configuration as is or tailor it to your specific requirements.

Part 3. Administration

Chapter 6. Performance tuning considerations

Performance monitoring and tuning in an SAP system environment is a complex and challenging task. An SAP system involves numerous elements, most notably the component itself, the network, AIX, Linux, Solaris, Windows, and z/OS operating systems, and the DB2 database management system. The various elements often require different performance metrics, tools, and monitoring and tuning procedures.

This chapter is not intended to be an all inclusive, one-stop performance management reference but rather a starting point, a summary of tuning steps that should be done prior to and soon after the installation of an SAP system, and a place to find pointers to further documentation.

In order to evaluate the effects of tuning and detect the development of new bottlenecks and performance deficiencies, you need to establish a base for performance evaluation. This can be done by collecting and storing the performance data over a longer period of time, but most importantly before and after any tuning activities.

Some essential aspects of performance monitoring and tuning were already covered in Chapter 5, "Performance tuning considerations before installing an SAP system," on page 71, but since they remain valid throughout your system's lifetime, you may want to reread the chapter before you proceed with the following sections.

Further information on performance tuning is available in numerous sources. Here are some selected ones:

- SAP Notes

  Search arguments:
  - Primary component: BC-DB-DB2
  - Category P(erformance)
- Redbooks

  Go to the corresponding home page at `http://www.redbooks.ibm.com` and search for ″SAP″.
- DB2 for OS/390 and z/OS Library
- General z/OS UNIX System Services tuning tips and hints are available at `http://www.ibm.com/servers/eserver/zseries/zos/unix`.

Performance monitoring tools

A complex system such as an SAP system requires sophisticated tools for performance monitoring and tuning. The performance monitoring suite that is a constituent part of the SAP Computing Center Management System (CCMS) includes a number of monitors for various parts of the system, such as:

- Work Process Load monitor
- Application Server Buffers monitor
- SQL Trace monitor
- Network monitor
- Operating System monitor

- DB2 monitor

Apart from providing crucial input for evaluating and tuning the performance of an SAP system, the single point of control and a common "look and feel" make these monitors very comfortable to use.

You can reach the monitors if you start from the SAP main panel and navigate to them by following the path **Tools**, **Administration**, **Monitoring**. Alternatively, you can use the shortcuts, that is, transaction codes. For example, to get to the DB2 monitor, call SAP transaction ST04, or to control the SQL Trace, call SAP transaction ST05.

For performance monitoring, you have to set up RFCOSCOL to serve all performance monitoring requests.

A full description of the SAP performance monitoring suite is provided in the SAP Online Documentation. The DB2 monitor integrated into the suite is described in *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*. For those of you who will be monitoring and tuning an SAP system on a more permanent basis, there are a number of courses on the topic offered by the SAP education centers. In any case, the best learning technique is to use the monitors and explore their capabilities in a working system.

The network performance statistics monitor is described in the following sections. The z/OS operating system monitor (SAPOSCOL) specific to the DB2 for z/OS platform is described in the *SAP Database Administration Guide*.

Generating and analyzing network performance statistics

The ICLI client and server support the generation of statistical information on the performance of the network between SAP application servers and database servers. While the SAP application server is running, you can switch the networks performance statistics on and off in two ways via SAP transaction ST04:

- by defining the time in seconds after which the statistical information is externalized and reset.
- by defining the number of ICLI request/response pairs after which the statistical information is externalized and reset.

The statistical information will appear in the SAP developer trace files dev_w*.

You can also turn on the statistics counters *before* the SAP application server is started by setting the ICLI client environment variable ICLI_NET_STATS_INTERVAL or ICLI_NET_STATS_SECONDS and in addition ICLI_TRACE=1, but then you can no longer change the settings at runtime via transaction ST04.

If you set *both* variables, the value of *ICLI_NET_STATS_SECONDS* takes precedence over the value of *ICLI_NET_STATS_INTERVAL*.

Scope of the network statistics function

The scope of the statistical data collected depends on how the statistics function is turned on:

- If you turn it on for one SAP application server by setting one of the environment variables before startup, the network statistics are generated for all SAP work processes of this SAP application server.

- If you turn it on for one SAP application server using SAP transaction ST04, the data will be collected for all SAP work processes of this SAP application server *and* for all others connected to the same ICLI server. This behavior allows you to generate a complete picture of the network performance for one ICLI server.

  Note that with SAP transaction ST04 the performance data is measured as long as the network performance statistics function is turned on.

We recommend that you use 300 (seconds) as the sampling time or 5000 (packets) as the request/response pair sampling rate. Usually, these values will not have a noticeable performance impact. You should monitor the growth of the SAP developer trace files (`dev_w*`). If they grow too fast with your current value, you should increase it.

Network turnaround time

The ICLI network performance statistics provide several statistics values derived from the network turnaround time. This is illustrated in the following figure:
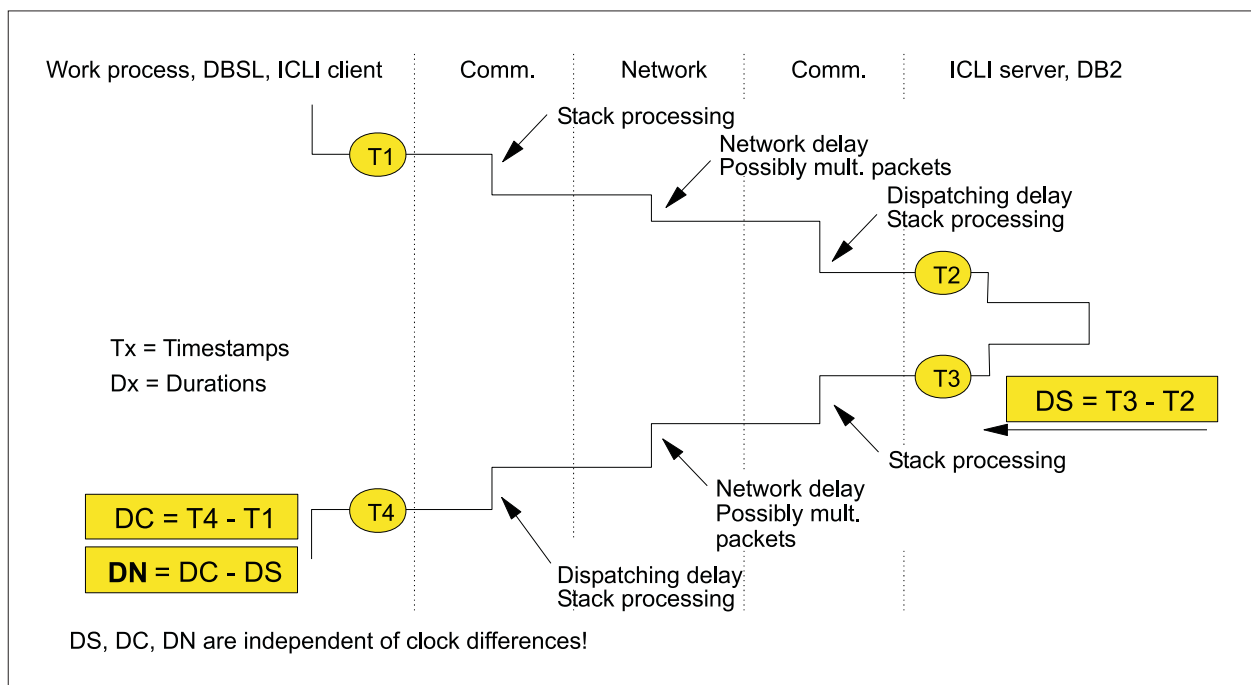
Figure 17. ICLI network performance statistics based on network turnaround time

The *network turnaround time* (DN) is defined as the sum of the request transmission time and the response transmission time.

The *request transmission time* is defined as starting when the first data packet of a request leaves the client (T1) and ending when the server receives the last data packet of that request (T2).

The *response transmission time* is defined as starting when the first data packet of a response leaves the server (T3) and ending when the client receives the last data packet of that response (T4).

Since the clocks on the client and server sides are not synchronized, only the time differences on each side can be utilized: *DC* is the time difference (*T4-T1*) on the client side, *DS* is the time difference (*T3-T2*) on the server side, and the network turnaround time (*DN*) is *DC-DS*.

This means that any processing time on the database server, for instance, inside of the ICLI server and DB2, is not part of the network turnaround time.

Network turnaround time may include several traversals of the network, for instance, if the request or response is split into multiple packets, or in the case of retries (in which case one or more TCP/IP acknowledgement timeouts are also included). It also includes the processing time within the TCP/IP protocol stacks on the application server side and the database server side, as well as dispatching delays on both sides.

Statistics counters

For each request/response pair, the network turnaround time is measured and maintained in an accumulated manner in a set of statistics counters that reside in memory in the ICLI client. There is one set of statistics counters for each SAP work process.

Once the number of request/response pairs transmitted has reached the number specified in the environment variable *ICLI_NET_STATS_INTERVAL* or the number of seconds specified in the environment variable *ICLI_NET_STATS_SECONDS* has elapsed, the statistics counters are externalized as messages that appear in the SAP developer trace files (`dev_w*` files in the directory `/usr/sap/<SAPSID>/<Instance_Name>/work` on AIX, Linux, and Solaris and `\\<SAPLOCALHOST>\saploc\<SAPSID>\<Instance_Name>\work` on Windows, respectively) and they are reset to 0.

In the case of *ICLI_NET_STATS_INTERVAL*, a value of the environment variable of 1000 means that one set of statistics counters is written for each work process and for each 1000 request/response pairs. In the case of *ICLI_NET_STATS_SECONDS*, a value of 3600 means that one set of statistics counters is written for each work process every hour.

The following is a sample of a "network statistics" message that is written to the developer trace files:

```
  ICLC2125I Network Statistics summary output:
BEG                 END                    #    AVG-NTT (ms)   ...   ...    TotalSent(KB)  TotalRecvd(KB)
2001/05/29 16:51:59  2001/05/29 16:52:08  1000          8.116   ...   ...    118.164        113.281


29.05.2001 16:52:08
ICLC2126I Network Statistics about sent packets (requests):
#PAC    #REQ   AVG(KB)   MIN(KB)   MAX(KB)   | <128  | <256  | <512  | <1024 |... <...| <31999| <64000
-----------------------------------------------------------------------------------------------------
2       500    0.118     0.118     0.118       500      0       0       ...0  ...0      0      0
3       500    0.118     0.118     0.118       500      0       0       ...0  ...0      0      0


29.05.2001 16:52:08
ICLC2127I Network Statistics about received packets (responses):
#PAC    #REQ   AVG(KB)   MIN(KB)   MAX(KB)   | <128  | <256  | <512  | <1024 |... <...| <31999| <64000
-----------------------------------------------------------------------------------------------------
2       500    0.113     0.113     0.113       500      0       0       ...0  ...0      0      0
3       500    0.113     0.113     0.113       500      0       0       ...0  ...0      0      0


29.05.2001 16:52:08
ICLC2128I Network Statistics about the time consumption of request/response pairs:
#PAC    #REQ   AVG(ms)   MIN(ms)   MAX(ms)   | <2    | <4    | <8    | <16   | <32   | >32
-----------------------------------------------------------------------------------------------------
2       500    7.096     3.380     106.548       0      30     350     118      0      2
3       500    9.136     3.122     202.333       0      10     350     138      0      2
```

The following values appear in the developer trace files (note that some of the values described in the following list are not shown in the sample above but do appear in the developer trace files):

BEG   Starting time of this sampling interval.

END   Ending time of this sampling interval.

#REQ or #

Number of request (sent)/response (received) pairs with the same number of data packets (#PAC value). The sum of all #REQ values of one message in the developer trace files is equal either to the value of the environment variable *ICLI_NET_STATS_INTERVAL* or to the sum of all data packets sent or received during the time given by *ICLI_NET_STATS_SECONDS*. In the example, there have been 500 request/response pairs that could be transmitted with 2 data packets (one in the request and one in the response) and 500 request/response pairs that needed 3 data packets for transmission. The third data packet has been caused by large requests for some request/response pairs and by large responses for others.

AVG-NTT (ms)

Average Network turnaround time in milliseconds.

AVG-REQTP (KB/s)

Average request throughput over all pairs in KB/s.

AVG-RSPTP (KB/s)

Average response throughput over all pairs in KB/s.

TotalSent (KB)

Overall number of kilobytes sent.

TotalRecvd (KB)

Overall number of kilobytes received.

#PAC Number of data packets transmitted between client and server for a request/response pair. The statistics counters are grouped by this number. The #PAC value counts all data packets of both request and response. Not counted are data packets caused by retries and acknowledgement packets.

AVG (KB)

Average request/response packet size in KB.

MIN (KB)

Minimum request/response packet size in KB.

MAX (KB)

Maximum request/response packet size in KB.

<128 (<n)

Number of packets less than 128 (or 'n') bytes.

AVG (ms)

Average time spent for a request/response pair in milliseconds (ms).

MIN (ms)

Minimum time spent for one request/response pair in milliseconds (ms).

MAX (ms)

Maximum time spent for one request/response pair in milliseconds (ms).

<2 (<n)

Number of packets with network turnaround time less than 2 (or less than *n*) in milliseconds (ms).

Measuring network turnaround time

Use SAP transaction ST04 to measure network performance:

- To get a rough feeling for your network's performance, use the function **ICLI Ping**. This function sends and receives data packets across the network the same way that your transactions are running and reports the time (in milliseconds) spent on the network without having to restart the ICLI server. To guarantee

that the returned number represents an average, a sample of 10 send/receives is
performed and the mean value is returned. Figure 18 depicts the ICLI Ping
process and contrasts it with the standard TCP/IP ping:
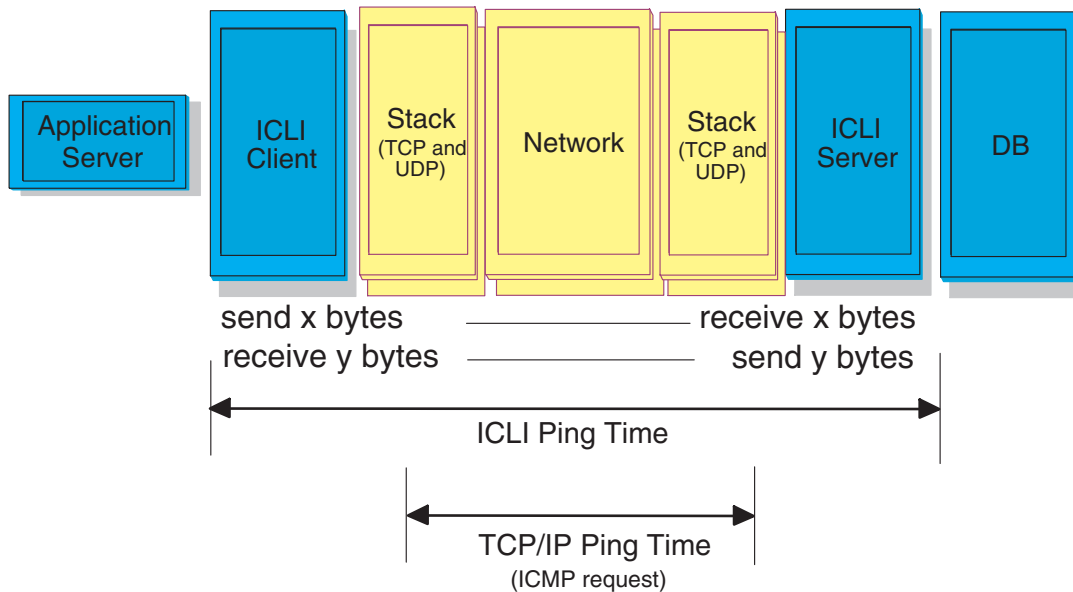
Figure 18. ICLI ping vs. standard TCP/IP ping

- To become more confident about your network performance, use the function
  Network Statistics. Run the statistics for at least one day and analyze the
  statistics record written to the dev_w* files thoroughly. Essential counters are the
  average response times for the packet sizes being sent.

Workload Management (WLM) — Classification of SAP workload

In the following sections, it is assumed that

- you are familiar with the concepts of WLM goal mode and the related
  terminology. For detailed information on the concepts and terminology of WLM
  (for example, the terms *classification rule*, *enclave*, *subsystem type, service class*,
  velocity goal, and *WLM work qualifier*) see:
    - *z/OS MVS Planning: Workload Management*
    - *z/OS MVS Programming: Workload Management Services*
    - the WLM Web site at `http://www.ibm.com/servers/eserver/zseries/zos/wlm`.
- your SAP system is running in WLM goal mode.

- you have set up your SAP application servers as described in "Preparing SAP application servers for WLM" on page 37, namely, you have set them up in such a way that the workload of each application server can be uniquely identified by its specific set of attributes.
- you have adopted the basic configuration described in "Minimum WLM configuration for SAP workload" on page 36.

In the following sections you will learn how you can extend the minimum WLM configuration for the ICLI server by classifying the different types of SAP workload running on your SAP system according to your business needs. Most of the information provided for the ICLI server is also valid for the SAP application server on z/OS. Exceptions to this rule are described in "Classifying the workload of SAP application servers on z/OS for WLM" on page 87.

ICLI server workload classification for WLM

As described in "Workload Management (WLM)" on page 36, the workload of an ICLI server is classified by the following attributes:
- Database attach name
- Host name
- SAP system number
- SAP work classification type (one of DIALOG, BATCH, UPDATE, UPDATE2, GENERIC, SPOOL, ENQUEUE)
- User ID

The ICLI server, which always runs on the z/OS server, consists of several different work threads, i.e., some ICLI-server-specific threads and "SAP-specific" threads, each of which serves one connection from an SAP work process. Each SAP application server work process that connects to an ICLI server (and thus the ICLI server thread that serves this SAP work process) is characterized by a set of the above mentioned attributes. You can utilize this fact by specifying performance goals for ICLI server threads according to their specific combination of attributes. Each combination of attributes corresponds to a so-called enclave, which means that:
- each ICLI server work thread is associated with (joined to) the enclave defined by the underlying attributes
- all ICLI server work threads that have the same set of attributes run in the same enclave.

An enclave is a logical transaction that can comprise multiple threads in one or more address spaces and is reported on and managed as a unit by WLM. The attributes classifying ICLI server workload are also used as attributes of the enclave in which the specific type of ICLI server workload is running.

An ICLI server thread enclave is created when the first SAP process with a specific set of attributes connects to the ICLI server. The enclave exists for the ICLI server's lifetime and is used for all ICLI server threads that have the same set of attributes.

WLM provides a set of so-called subsystem types, each of which consists of a set of WLM work qualifiers. While there are a number of IBM subsystem types (e.g., DB2, OMVS), *SAP* is the only predefined non-IBM subsystem type. This is the WLM subsystem type you will use to define performance goals for ICLI server threads (and for the SAP application server on z/OS as well).

For the subsystem type *SAP*, WLM uses the enclave attributes listed above as WLM work qualifiers to associate an enclave with a service class. The service class contains the performance goals for the enclave and thus for the workload running in the enclave. An association between enclaves and service classes is created by using classification rules in which WLM work qualifiers are mapped to service classes. You define these classification rules by using the WLM work qualifiers corresponding to the enclave attributes given above as shown in the following table:

Table 7. Enclave attributes and the corresponding WLM work qualifiers for WLM subsystem type SAP

Enclave attribute (SAP parameter)	WLM work qualifier	WLM work qualifier (full name)
Database attach name	PK	Package name
Host name	SPM	Subsystem parameter (see note)
SAP system number	PR	Procedure name
Work classification type	TN	Transaction name
User ID	UI	User ID
Note: When using the 6.40 kernel, the host name is defined in the account information (AI) work qualifier for an SAP application server (see "Classifying the workload of SAP application servers on z/OS for WLM" on page 87). The ICLI server uses SPM in all cases.		

In general, the full names of the WLM work qualifiers differ from the names of the enclave attributes, because WLM prescribes a fixed set of work qualifiers by which applications supporting WLM management can pass the attributes qualifying their workload.

It is not necessary to define classification rules for all workload qualifiers an ICLI server provides to classify its work load. You can, for instance, define classification rules for database attach names and work classification types only.

Defining the right performance goals

All ICLI server work threads run for a long period of time and handle requests from SAP work processes in an infrequent manner. The enclaves to which the work threads are dispatched are created dynamically at SAP startup.

This means that only velocity goals constitute the right kind of performance goals one should define for the service classes to which the enclaves are mapped. With velocity goals you can achieve goals like: "When this work is ready, be sure it runs without delay," "When that work is ready to be executed, keep it plodding along to ensure it will eventually finish," or gradations in between. So velocity is a measure of the acceptable processor and storage delays while work is capable of running. To familiarize yourself with the concepts of WLM and the related terminology, refer to *z/OS MVS Planning: Workload Management*, *z/OS MVS Programming: Workload Management Services* or visit the WLM Web site at http://www.ibm.com/servers/eserver/zseries/zos/wlm.

Configuring a WLM service definition

You must at least define a special WLM service class with appropriate performance goals and make it the default service class for WLM subsystem type *SAP*. As a result, all SAP workload will run in this service class. Otherwise, the enclaves created for SAP workload would run with the service class SYSOTHER, which would result in poor performance for the ICLI server work threads and the SAP

processes associated with the enclaves. Such a minimum WLM goal mode configuration for SAP workload is described under "Minimum WLM configuration for SAP workload" on page 36.

To extend the minimum configuration, define additional WLM service classes (e.g., SAPHIGH or SAPMED) with the performance goals required for your business (for more information, see "Defining the right performance goals" on page 84).

If you have not defined it yet, create the subsystem type *SAP* in your WLM environment.

Do not make the WLM service definition too complex, in order to ensure good WLM responsiveness. This can be achieved by observing these rules:
- Do not create too many WLM service classes. Reuse service classes whenever possible.
- Choose a WLM service class as default which is appropriate for most of you SAP system's workload.
- Specify classification rules only for enclaves whose workload should run with special priorities not provided by the default service class.

To assign enclaves to service classes, you need to provide work qualifiers for the subsystem. Since the enclaves use the work qualifiers USERID (UI), TRANSACTION_NAME (TN), PACKAGE_NAME (PK), PROCEDURE_NAME (PR), and SUBSYSTEM_PARAMETER (SPM), you need to provide strings which match the current enclave attributes provided by the ICLI server and SAP application server describing the workload at runtime. However, you do not need to specify classification rules for all WLM work qualifiers representing attributes of WLM enclaves in which SAP workload is processed. It depends on your business needs for which work qualifiers you specify classification rules.

In order to define the classification rules for an *ICLI server*, you map the enclave attributes of the ICLI server threads to the WLM work qualifiers as specified in the following list. Due to a limitation in WLM, strings can only be specified in uppercase.

PK   As the package name (PK), enter the database attach name for the ICLI server.

PR   Specify the SAP system number as the procedure name (PR) work qualifier.

SPM   Enter the name of the (non-z/OS) host on which the ICLI clients are running as the subsystem parameter (SPM) work qualifier. Due to a limitation in the ICLI client-server interface, only host names of up to 32 characters can be matched.

TN   Create a transaction name (TN) entry for each of the following eight strings:
`ENQUEUE, UPDATE, UPDATE2, DIALOG, SPOOL, BATCH, GENERIC.`

UI   For the WLM work qualifier UI (USERID), enter the user ID that is used to start the ICLI server.

You can use these attributes to define specific performance goals for workloads created by different SAP application server instances. For example, in "Scenario 1" on page 38 an SAP system is set up with a central instance and a dialog instance running only batch jobs. Both have the same SAP system ID (C11) and the same

SAP system number (11), but the central instance is running on host AIXHOST1 while the dialog instance is running on AIXHOST2. Both are running against DB2 subsystem C11.

If the workload on the central instance is to have a higher priority than the workload on the dialog instance, and dialogs on the central instance are to have a higher priority than batch jobs running there, this can be achieved by giving medium priority to the central instance's workload and low priority to the dialog instance's workload. Dialogs on the central instance can have high priority and batch jobs can have low priority.

Assume that you have defined the WLM service classes SAPHIGH, SAPMED, and SAPLOW for high, medium and low priority, a corresponding set of WLM classification rules would look as follows:

Table 8. Sample WLM classification rules

Level	Work qualifier	Work qualifier name	Service class
1	SPM	AIXHOST1	SAPMED
2	TN	DIALOG	SAPHIGH
2	TN	BATCH	SAPLOW
1	SPM	AIXHOST2	SAPLOW

In this example, we assume SAPMED as default service class for subsystem type *SAP*. The numbers in column **Level** define the hierarchy of the classification rules. In the example, we have a level 1 rule associating workload running on host AIXHOST1 with service class SAPMED and two level 2 rules associating DIALOG and BATCH workload to service classes SAPHIGH and SAPLOW, respectively. These level 2 rules only apply to workload running on host AIXHOST1. This nesting of classification rules allows you to associate workload matching a certain combination of work qualifier names to a service class.

Among the enclaves an ICLI server creates, there are some enclaves with `ENQUEUE` as an SAP work classification type (or transaction name). These enclaves exist only to provide consistent code logic, because the SAP `ENQUEUE` work process only connects to the database but does not perform any requests. In fact, it is not necessary to define classification rules for those enclaves. However, do not be surprised if such enclaves appear in your RMF report.

Note: Even if the ICLI server is started as a started task, it is not necessary to define any classification rules within the started task subsystem. Only **SAP** subsystem settings are required.

Choosing appropriate velocity goals and business priorities: Due to the very different customer requirements on SAP production systems, it would not make sense to recommend a concrete WLM service policy. It is an ongoing task of z/OS system administrators to determine the optimal settings. The following three questions, however, may help them do this:

What is and how heavy is the mixed workload on your z/OS system?
This influences the chosen velocity goals and business priorities of the service classes (into which the ICLI server enclaves are mapped) compared to other z/OS applications

How is your SAP system configured?
This includes whether and how your UPDATE2 process is defined and how many work processes of what type you have configured.

What kind of business do you run within the SAP system?
This influences the velocity goals and business priorities of the service classes (into which the ICLI server enclaves are mapped) with respect to each other.

Recommendations on assigning priorities to different kinds of SAP workload

The following list gives some general recommendations on how the different service classes should be related in terms of velocity goals and business importance. There are no recommendations as to the actual velocity goals you should choose compared to other workload (and service classes) you have in your system.

- Usually, UPDATE2 work processes can run at a lower velocity.
- If your applications use a lot of synchronous updates, make sure that the corresponding ICLI server enclave (UPDATE) has high velocity to prevent delays.
- Because the ENQUEUE work process connects to the ICLI server but works only on the application server, you can assign a discretionary (no velocity) goal for the respective enclave or even leave this enclave out of the service definition.
- If you have the emphasis on DIALOG work processes and short response times, make sure that the DIALOG enclave gets very high velocity compared to all others. For batch-oriented installations, assign high priorities to the BATCH enclave.
- The SPOOL enclave should usually have low velocity, with the exception of installations where you do a lot of high-priority printing.
- If a lot of SAP transports are planned or if you use other SAP tools very frequently, ensure that the GENERIC enclave has a sufficient velocity.
- **In any case, you must ensure that the service classes used for the ICLI server enclaves are given a lower priority than the service classes to which the subsystems/address spaces (see "Settings required for the entire system" on page 36) have been assigned.**

Classifying the workload of SAP application servers on z/OS for WLM

In general, classifying the workload of SAP application servers on z/OS for WLM is similar to classifying the workload for the ICLI server (see "ICLI server workload classification for WLM" on page 83). Therefore, only the differences compared to the ICLI server are described here.

The WLM work qualifiers for the SAP application server workload are the same as for the ICLI server workload, with the following exceptions:

- The work qualifier PK always represents the SAP system ID.
- For SAP kernel releases up to and including 6.20, the host name (work qualifier SPM) can be up to 255 characters long. For SAP kernel releases beginning with 6.40, including the 6.40 downward compatible kernel, the host name is specified to WLM by the first 32 characters of the WLM work qualifier AI (account information).

> **Note:**
> When switching from SAP kernel release 6.20 to 6.40, consider the
> following. For SAP application servers of SAP kernel releases 6.20 and
> earlier running on z/OS, the host name was specified using the WLM
> work qualifier SPM and could be up to 255 characters long. Corresponding
> classification rules will no longer work with SAP application servers of
> SAP kernel releases 6.40 and above on z/OS. If you have used the host
> name to classify workload of SAP application servers on z/OS for SAP
> kernel release 6.20 and below, remove such classification rules and replace
> them with the corresponding rules using the account information (AI) work
> qualifier (positions 1-32).

- The SAP work classification type (TN) may be one of DISPATCH, MSGSRV,
  GATEWAY, WEBSRV, DIALOG, BATCH, UPDATE, ENQUEUE, SPOOL, and
  UPDATE2.

The WLM enclaves for the SAP application server workload have the same
attributes as the WLM enclaves for the ICLI server workload. However, the
enclaves in which SAP ENQUEUE processes are running (TN = ENQUEUE) do not
have the SAP system number as an attribute. Therefore, you should not nest WLM
classification rules with SAP system numbers for work qualifier PR and TN =
ENQUEUE if this applies to SAP application server workload.

ENQUEUE work processes do CPU-critical and important work. The ENQUEUE
processes of SAP application servers on z/OS should have a relatively high
priority compared to the ICLI server thread, which handles the DB-related part of
the ENQUEUE work processes, which is not as CPU-intensive.

In order to enable WLM to manage the workload of an SAP application server on
z/OS according your classification rules, you need to switch on WLM management
for this SAP application server. This is described in *SAP Web Application Server
Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* under "Switching on
WLM Management".

Initial setup of classification rules for the SAP application server

Since the business requirements of our customers are highly volatile, we can only
give the following recommendations for an initial setup of classification rules
(SAPCRIT, SAPHIGH, SAPMED, and SAPLOW are used as examples of WLM
service classes for defining critical, high, medium, or low priorities):

- Initially, define the same medium priority for all SAP processes.
- If several SAP application server instances are running on your system, define
  critical priorities (SAPCRIT) for workloads qualified by the transaction names
  (TN) MSGSRV, DISPATCH and ENQUEUE, since these workloads are critical to the
  performance of an entire SAP system if the central instance is running on z/OS.
- You can dynamically update the classification rules according to your business
  needs, e.g., increase the priority for TN BATCH to SAPHIGH or decrease the
  priority for TN UPDATE to SAPLOW.

Sample WLM startup service definition for the ICLI server and SAP application server

This section describes a WLM service definition that contains all of the entries needed for a quick startup of a z/OS system dedicated to an SAP database server on z/OS.

However, this should only give you an idea of how to create a WLM service definition. If you simply adopt the given service definition, it is entirely possible that you won't run with the best conceivable performance.

Since the sample service definition dates from the time when the attributes *database attach name*, *host name*, and *SAP system number* were not yet available, you will find that they are not considered. Instead, it is assumed that different ICLI server user IDs are used to distinguish between the workload of different ICLI servers. While you can still proceed to use such a setup, it is not recommended to set up new SAP systems in such a way. This is because using different user IDs for some or all ICLI server instances increases EDM Pool storage usage and decreases performance through global cache hit ratio degradation. It is recommended that you use the new attributes to define different performance goals for workload generated by different SAP application servers. In this case, you have to customize the sample service definition accordingly.

Assumptions

The sample WLM startup definition is based upon the following assumptions pertaining to the definition and classification tables presented below:

- You only have one ICLI server, or all ICLI server instances run with the same user ID.
- You have an SAP application server on z/OS as the Central Instance (CI).
- Your database system name is SAPP.
- You have only one SAP system (i.e., a production system).
- You are running on a *dedicated system*, i.e., there is no other business application running.
- There is no batch (the WLM definition provided here includes no rules on how to classify JES).
- ICLI is started as a started task.
- You have only one **Service Policy**.
- Default **Service Coefficients** are used.
- No resource groups are defined.
- SAP batch work is not important compared to dialog work.

> **Caution:**
> It is absolutely necessary that you verify your concrete system setup and applications against these assumptions. If even only one of these assumptions is incorrect, this can have a strong impact on the performance of your system or application.

Startup service definition

The following tables describe the data you have to enter into the new (empty) WLM **service definition** on your system. This information about workloads, service classes, and so on appearing in the tables has been abridged, while the

input using the WLM ISPF application can be entered "as is." After you have completed your **service definition**, install it and activate the **service policy**.

Table 9. Workloads and service classes

Workload	Service Class	Period	Imp[1]	Goal		Remarks
				Velocity	Service Units	
STC	CRITICAL	1	1	85		For critical but low-volume work.
	HIGHSTC		2	65		This service class should comprise very important work, i.e., started tasks for communication.
	LOWSTC		3	35		Used for low-priority or non-classified work.
TSO	TSO	3	2	80% < 0.8 sec	1500	Verify that the given duration (service units) is adequate for your processor capacity.
			3	80% < 1.5 sec	9000	See above.
			4	25		
SAP	SAPCRIT	1	2	65		For MSGSRV, DISPATCH, and ENQUEUE work processes of SAP application servers running on z/OS as central instances.
	SAPHIGH			55		For your high-priority SAP work (usually related to DIALOG work processes)
	SAPMED			35		Normal SAP-related work
	SAPLOW		3	25		Less-important SAP-related work

[1]	Imp = (Business) Importance of work (1 = very important, 5 = least important)

Table 10. Classification groups

Classification Group	Name	Qualifier
Transaction Name Group	SYSTASKS	DUMPSRV
		RMFGAT
		RMF
		SMS
		SYSBMAS
		TRACE
		LLA
		JES2
		RACF
		TSO
		RRS
		PCAUTH
		NET
		VTAM*
User ID must be ″UserID Group″	SAPUTIL	ICLIRUN[1]
		<SAPIDs>[2]

[1]   This is the user ID you use to start the ICLI server.

[2]   All SAP-related user IDs used to execute SAP tools and utilities (e.g., SAPOSCOL).

Table 11. Classification rules

Subsystem	Q-Type[1]	Q-Name[1]	SVC
TSO		Default	TSO
OMVS		Default	LOWSTC
	UIG	SAPUTIL[2]	SAPMED
SAP[3]		Default	SAPLOW
	UI[4]	ICLIRUN[5]	
	TN[6]	GENERIC	SAPMED
		DIALOG	SAPHIGH
		UPDATE	SAPHIGH
		UPDATE2	SAPMED
		SPOOL	SAPLOW
		BATCH	SAPLOW
STC		Default	LOWSTC
	TNG	SYSTASKS	SYSSTC
	TN	SAPPMSTR	CRITICAL
		SAPPDBM1	CRITICAL
		SAPPDIST	HIGHSTC
		SAPPIRLM	SYSSTC
		SAPPDCX%	CRITICAL
		FTP*	LOWSTC
		TCPIP	HIGHSTC
		OMVS	HIGHSTC

[1] Q-Type (Q-Name) = work qualifier type (name) (the **Starting Position** field in the **Classification Rules** panel is always empty).

[2] The user IDs under this user ID group are used for SAP-related tools and components, including ICLI server UID and SAPOSCOL.

[3] In this example, only two of the five possible work qualifiers for subsystem type *SAP* are used.

[4] This startup service definition describes a setup in which the UI qualifier was used to allow a distinction between different ICLI server instances or groups of server instances. In earlier releases of SAP, this was the only way to specify different classification rules for different ICLI server instances. Since the introduction of the attributes *database attach name*, *host name*, and *SAP system number*, it is neither necessary nor recommended to use different user IDs for ICLI servers. If all of your ICLI server instances are started with the same user ID, ignore the UI field. If you want to set different performance goals for workload running on different ″remote″ SAP application servers, use the work qualifiers PK, PR, and SPM instead. For SAP application server instances on z/OS, it is recommended to classify the workload by database attach name, host name, and SAP system number in order to distinguish different instances, instead of using the user ID.

[5] This is the user ID you use to start the ICLI server.

[6] TN for SAP is a level-2 qualifier. Thus, it is used only if the level-1 UI matches the actual work qualifier. You can use it on level 1 if you don't want to classify on an ICLI user ID level (see also note [3]).

Monitoring considerations

Always keep in mind that it is absolutely necessary to monitor the actual workload and the goals on your running system using RMF (or an equivalent product) even

if you are using the concrete startup definition given above. This is because the parameter settings and the combination of applications running on your system are bound to be unique to *your* system. Therefore, based on the reports you gather using your performance monitoring tool, you should refine these basic definitions or classification rules to improve system responsiveness and performance. This is an iterative process.

Note that it is not possible to monitor WLM-related performance via the integrated z/OS operating system monitor, SAPOSCOL.

Displaying WLM settings within the SAP system

Call SAP transaction DB2W to display the current WLM settings of your z/OS database server. In order to use this SAP transaction, you must have set up SAPOSCOL and RFCOSCOL on your z/OS system and both components must be active.

Chapter 7. ICLI client and server

ICLI overview

The ICLI component contains a number of different modules that are responsible for the exchange of database information between an SAP System and the DB2 database. This section provides a short description of these modules and outlines the internal design of some of the components to help you understand the behavior of the complete system.

These are the main parts that are shipped with Component ID 5695OICLI:

- the ICLI server and ICLI clients
- the LIBASCII DLL (fomeliba.dll), which is used on z/OS as an ASCII/EBCDIC converter by SAP kernel releases 6.20 and below

The ICLI server and various ICLI clients are used to implement the remote SQL access from the SAP application server to the DB2 database on z/OS. Currently supported clients include AIX 5.x (64-bit), Linux, Solaris, and Windows. The ICLI components are not needed for SAP application servers running on z/OS because the database is accessed locally in this configuration.

The ASCII/EBCDIC converter DLL `fomeliba.dll` is a shared library which is needed only by SAP application servers on z/OS. It implements a layer that allows an SAP work process on z/OS to run as an ASCII application inside z/OS UNIX System Services. An SAP application server on z/OS expects the DLL in `/usr/lib`. Standard z/OS installation places the DLL in the HFS directory `/usr/lib/` as a symbolic link to `/usr/lpp/icli/sbin/fomeliba.dll`. See "File system setup" on page 49 for more details.

Note: As of SAP kernel release 6.40, including the 6.40 downward-compatible kernel, this DLL is no longer required. See "File system setup" on page 49.

ICLI client/server internals

The ICLI client and server allow SAP application servers to access a z/OS database server remotely across a network. They do not implement a general-purpose database call-level interface but are designed to cooperate in an optimal way with the SAP Database Interface (DBIF), also referred to in the following as the Database Service Layer (DBSL). The DBSL uses only a certain subset of the database SQL functionality and ICLI delivers exactly that subset.

Several mechanisms are employed to optimize performance:

- ICLI sends and receives only the data that is needed by the SAP DBSL.
- Database requests are bundled where possible.
- Array operations on INSERTs and SELECTs.
- DBSL combined update/insert operations are supported.
- ICLI prefetches database information to reduce network traffic (CLOB-, BLOB-length information)

For the SAP 6.20 kernel, the ICLI server load module `FOME620S` resides in an APF-authorized library. The z/OS installation by default places this module into `SYS1.LINKLIB`. The UNIX System Services executable `fome620s` exists in the HFS

directory /usr/sbin/, which is a symbolic link to /usr/lpp/icli/sbin/fome620s.
For this file, the sticky bit must be set (the command **ls -l** shows: -rwxr-xr-t),
which means that a load module with the same name (FOME620S) will be searched
for along the usual z/OS library concatenation for APF-authorized libraries.

When using the SAP 6.40 (downward compatible) kernel, there is no load module
FOME640S residing in SYS1.LINKLIB or another APF authorized library. Instead,
there is a UNIX System Services executable 'fome640s' in the HFS directory
/usr/sbin, which is a symbolic link to /usr/lpp/icli/sbin/fome640s. For this file
the extended attribute for the APF authorization must be set (the command ls -E
must show:

```
-rwxr-xr-x a-s- ..... fome640s
```

where the 'a' indicates that the executable is APF-authorized.

Whenever a new ICLI server is shipped via a PTF, the corresponding DBRM
modules FOME6201 (FOME6401) and FOME6202 (FOME6402) are also shipped. They
must be bound to the DB2 plan used by the ICLI server.

The ICLI clients are shared libraries for UNIX/Linux and a dynamic link library
(DLL) for Windows. They reside as binaries in the z/OS data set SYS1.SFOMDATA
and must be downloaded to the appropriate client (see "ICLI client download" on
page 138 for more information).

The ICLI client and server communicate via an internal protocol. The version of
this protocol is reported in the startup message of the ICLI server (ICLS1011I) and
is usually retained unchanged across the different ICLI PTF levels for an SAP
release. In certain cases, it might become necessary to change the protocol version.
In such a special case, you will be alerted by a corresponding ++HOLD ACTION
statement in the ICLI PTF and you must then download the ICLI clients
immediately after you have applied the ICLI PTF on z/OS. As long as the protocol
version does not change, you can run ICLI clients and ICLI servers even if they
have different internal versions. However, we recommend keeping both ICLI
clients and servers on the same level.

ICLI client/server communication

Figure 19 on page 97 shows how an ICLI connection between application server
and z/OS database server is established:
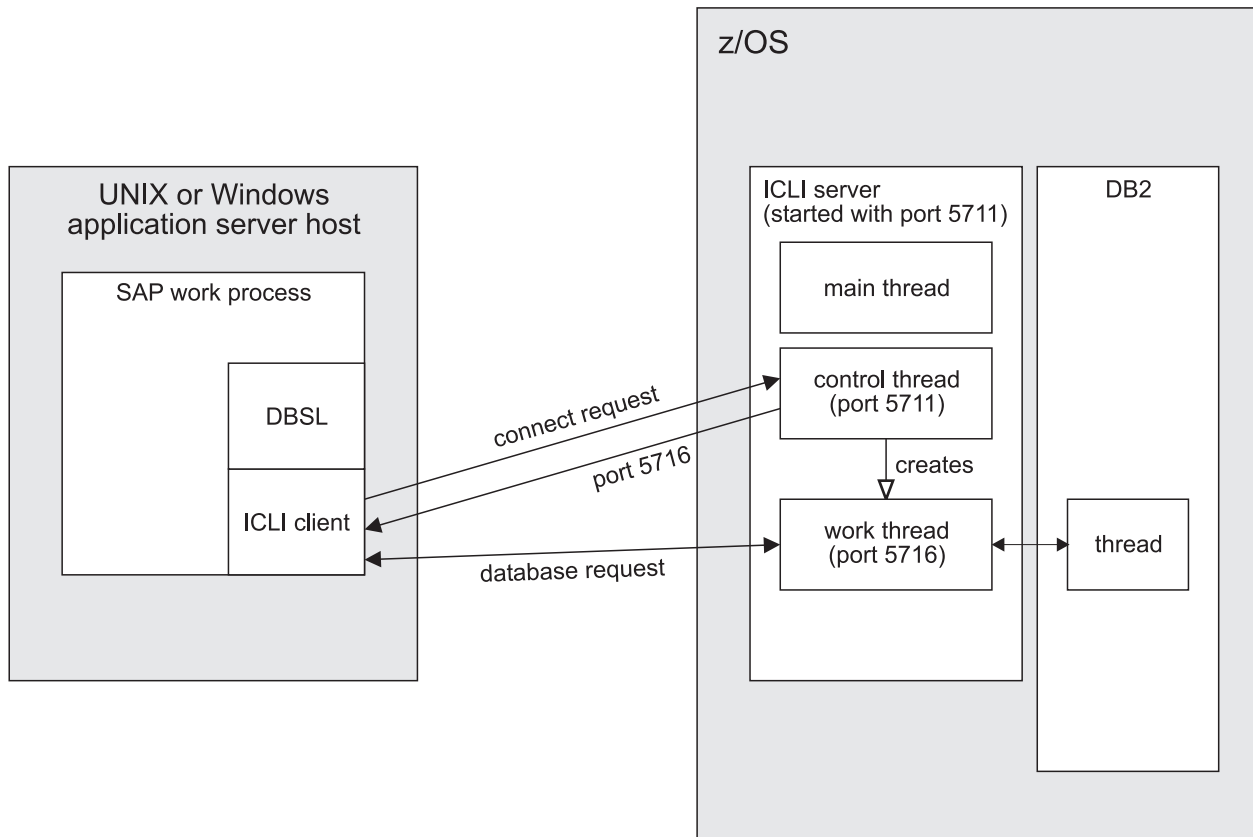
Figure 19. ICLI architecture

The process of creating a connection via ICLI consists of the following steps:

1. Every SAP program, for example an SAP work process, accesses the database via its DBSL interface. The DBSL passes the connect request to the ICLI client, which communicates it to the control thread of the ICLI server using the ICLI connection port number 5711.

2. The ICLI control thread creates an ICLI work thread, allocates a free port number (5716 in Figure 19) for further one-to-one communication, passes back the new port number and waits for further connection requests.

3. The ICLI client then issues a DB connect to the newly created work thread via port 5716. If PassTicket signon is active, the signon information sent from the client is checked by the ICLI work thread before any database connection is established. If the check fails, the ICLI client receives a response indicating that error, and the ICLI work thread terminates.

4. If the validation succeeds (or PassTicket signon is not active), the ICLI work thread connects to DB2, thereby creating a DB2 thread.

5. The ICLI client is now ready to accept database requests from the DBSL.

6. When a database request arrives, the ICLI client uses the received port number to send requests to the ICLI work thread.

7. The ICLI work thread executes the request and responds with the data from DB2.

ICLI code page conversion

DB2 stores SAP data in ASCII tables. This allows code page translations between SAP and DB2 to be avoided for SAP data. Only the SQL statement text is translated by the ICLI client into the proper EBCDIC code page before the statement is prepared by DB2. However, this happens quite rarely since DB2's Dynamic Statement Cache helps to effectively avoid SQL conversions.

Versions

The SAP DBIF and ICLI verify at startup whether the installed versions of the SAP system, ICLI client and ICLI server are compatible. Four numbers are used for this:

- SAP Version: This is a number identifying the release of the SAP kernel, for example 6.20.
- ICLI API Version: This is the version of the API between SAP DBIF and the ICLI client.
- ICLI Protocol Version: This is the version of the protocol between the ICLI client and server.
- ICLI Internal Version: This identifies different code levels of the ICLI client and server and is not used to verify whether the components are compatible.

Between SAP and the ICLI client, the SAP system version and the ICLI API version must be identical. Between the ICLI client and the ICLI server, the ICLI Protocol Version and the SAP version must be identical. As a result, an ICLI server and an ICLI client with different internal ICLI versions will work together as long as the ICLI Protocol Version is the same. In general, we recommend that you always keep the ICLI client version up-to-date (meaning on the same internal version level as the ICLI server) but you can also use this feature in the following way: apply an ICLI PTF, for example, during your z/OS maintenance window, and do the necessary rebind, but delay the download of the ICLI client to a later point in time (for example, your SAP maintenance window).

On the application server hosts, the SAP tools write their version information to SAP log files and to the dev_w<n> files, which are located in

/usr/sap/<SAPSID>/<Instance_Name>/work (for AIX, Linux, and Solaris) and

\\<SAPLOCALHOST>\saploc\<SAPSID>\<Instance_Name>\work (for Windows)

for SAP work processes, in the following format:

```
...
C SQL DRIVER VERSION is "17.09.0100"
C SQL DBMS NAME is "DB2/390"
C SQL DBMS VERSION is "06.01.0000"
...
```

The version reported as SQL DRIVER VERSION refers to the ICLI client. The format is vv.rr.ppmm where vv is the SAP version (17 stands for SAP release 6.40), rr is the API version, pp is the ICLI client's PROTOCOL version, and mm is the internal ICLI client version. If the ICLI client's PROTOCOL version does not match the PROTOCOL version of the ICLI server, you will receive appropriate messages on both sides as soon as you try to establish a connection.

On z/OS, the versions are written to the ICLI.<ID>.msg<n> file in the log directory of the ICLI server.

Example:

```
(0004) ICLS1300I The following ICLI client has successfully connected
       to the DB2 system:
       ICLI client hostname: hostname1
       ICLI client ip: 10.199.137.4
       ICLI client for SAP Version:  6.20
       ICLI client protocol version: 2
       ICLI client internal version: 0
       ICLI client time: Fri Feb 15 09:43:23 2002.
       ICLI client port = 54880 and process ID 1228826
       Workprocess class  = 0 (GENERIC)
       SAPSYSTEM = 14  SAP Workprocess Number = 0
       connected to database attach name
        Plan .
       Time: 1013762603.413263 = (Fri) 2002 Feb 15 09:43:23.413263
```

The ICLI server versions are reported in the server start-up message (ICLS1011I) or on the help text which is displayed when *-HELP* is specified at the command line.

The ICLI connection port

An ICLI server needs to know on which port it is to listen for connection requests from ICLI clients, and an ICLI client must know which port it is to use for connecting to the ICLI server. Both port numbers must be identical. These ports and their port number are called the *ICLI connection port*.

Considerations for selecting an ICLI connection port

The SAP installation tool defines the ICLI connection port for the ICLI client during the installation process. This section describes how to select and define the connection port for the ICLI server and how to change the connection port for an ICLI client if you need to change it after the SAP system has been installed.

The port number you select for the ICLI connection port depends on the communication stack you will use on z/OS. For more information on the Integrated INET stack and the Common INET stack, see "Defining the BPXPRMxx parmlib member" in the *Connectivity Guide.*

Since you are using TCP/IP as the communication protocol, you can run either the Integrated INET stack (recommended) or the Common INET stack.

To select a port number for the ICLI connection port, you should proceed as follows:
- For the Integrated INET stack, which we highly recommend, the following general scheme for using port numbers for TCP applies:
  - Ports between 1 and 1023 are reserved ports.
  - Ports between 1024 and 5000 (ephemeral ports) are generally assigned by the system when an application does a bind() with port 0.
  - Ports above 5000 are available to be assigned by applications.

  You should use a port number *greater than 5000* for the ICLI connection port. If you choose a lower port number, the port may already be in use by some other application.
- For working threads, the ICLI server also issues binds with port 0.

- For the Common INET stack the following applies:

  The first ephemeral port number and the range of ports for z/OS UNIX System Services are specified in the Common INET Physical File System (PFS) parameters *INADDRANYPORT* and *INADDRANYCOUNT*. For the ICLI server, you must specify a connection port number outside this range. We recommend that you use a port number *greater than the last port number* in the above-mentioned range. See the BPXPRMxx parmlib member for the range of excluded port numbers. For more information about *INADDRANYPORT* and *INADDRANYCOUNT*, see *z/OS UNIX System Services Planning*.

Tasks to be performed at the ICLI server

Especially if you are running multiple instances of the ICLI server, proceed as follows to define the ICLI connection port for the ICLI server:

1. Make a service entry either in the HFS file `/etc/services` or, if this file does not exist, in the MVS data sets `<TCPIP>.ETC.SERVICES` or `<SAPSID>ICLI.ETC.SERVICES`. The qualifier `<TCPIP>` stands for the high level qualifier under which the TCP/IP product was installed. `<SAPSID>ICLI` is the user ID under which the ICLI server runs. For the default service name `fome620s` (`fome640s`) the service entry would be specified as follows (example for 6.20):

   `fome620s 5711/tcp`.

   Make sure that the port number (5711 in the example) is not used by a previously defined service. Note that if `/etc/services` exists, then the service must be specified there. Otherwise, the service is found in the MVS data set `<TCPIP>.ETC.SERVICES`, which overrides any entries in `<SAPSID>ICLI.ETC.SERVICES`.

2. If you have made service entries as described above, you can change the ICLI connection port for an ICLI server by specifying a different service name with the option *-SERVICENAME* `<service_name>` on the ICLI server command line (see "ICLI server command line options" on page 101) or in your start procedure (see "Configuring the start procedures for the ICLI server" on page 66).

Instead of looking up service names, the port can also be specified on the ICLI server command line or in the start procedure using the option *-PORT* `<port_number>` (see "ICLI server command line options" on page 101). This explicit specification of the port number will override any port number derived from a service name. Service entry definitions, however, make the administration of the system easier.

> **Service names of ICLI server and client differ:**
> While the port numbers for an ICLI client and the ICLI server it connects to must be identical, the service names used to define the connection port in the services files will usually differ. This results from the fact that the service name on the application server host the ICLI client runs on is predefined by SAP, and that for each SAP system only one such entry (`sapdb2<SAPSID>`) can exist in the services file of each application server. In order to support installations that use more than exactly one ICLI server per SAP system under one z/OS, the ICLI server must use service names different from the default `fome620s` (`fome640s`) with a different connection port.

Tasks to be performed at the ICLI client

The *ICLI connection port* for the ICLI client must be specified in the files

- `/etc/services` (on AIX, Linux, and Solaris)
- `C:\WINNT\system32\drivers\etc\SERVICES` (on Windows)

with the following entry: `sapdb2<SAPSID> <port_number>/tcp`.

If you want to use an ICLI connection port for the SAP transport tool TP that differs from the above service entry, make sure you add an appropriate *R3_PORT* entry to the TP parameter file:

- `/usr/sap/trans/bin/TPPARAM` (on AIX, Linux, and Solaris)
- `\\SAPTRANSHOST\sapmnt\trans\bin\TPPARAM` (on Windows)

Associating the ICLI server with a specific IP address

By default, the ICLI server listens on its ICLI connection port for connection requests from ICLI clients coming in via *any* IP address that belongs to the z/OS on which it is running. This behavior has the disadvantage that you need to use a different ICLI connection port if you want to start a second ICLI server on the same z/OS. This restriction can be overcome by associating the ICLI server with a specific IP address. Then you can run multiple ICLI server instances on a z/OS system and let all of them listen on the same ICLI connection port.

Speaking in generalized terms, this feature allows you to use the same service (remote SAP SQL connection) offered by multiple ICLI server instances on a z/OS system without having to define different connection ports. If you have the database for different SAP systems on one z/OS, for example, this means that you can distinguish and represent each one by a different z/OS host IP address while using the same connection port for all.

When used in conjunction with dynamic virtual IP addresses (VIPAs), the new feature allows you to start an ICLI server and its associated dynamic VIPA on any arbitrary z/OS system in a sysplex. This simplifies VIPA management, because only 'VIPA range' definitions are necessary. Additionally, the decision as to which z/OS an ICLI server is started is then not restricted by TCP/IP definitions. (Before deciding to use dynamic VIPAs, see "Virtual IP Address (VIPA) as a recovery mechanism" in *High Availability for SAP on zSeries Using Autonomic Computing Technologies* to appraise the potential disadvantages for the administration of an SAP system.)

You associate an ICLI server with a specific IP address by setting the environment variable ICLI_BIN_IPADDR to a valid IP address. This is described in "ICLI server environment variables" on page 159.

ICLI server command line options

Except for the parameter *-HELP*, which cannot be used in conjunction with other parameters, you can specify the ICLI server command line options described below in any combination when starting an ICLI server:

- On the command line, when starting an ICLI server manually.
- In a UNIX System Services shell start script.
- In a started task or JCL batch job as you would define any other command line options or within the ICLI server variable *ICLI_PARAMETERS*.

For details on how to define the command line options in your start procedure, see "Configuring the start procedures for the ICLI server" on page 66.

(For 6.40, use the command 'fome640s'.)

```
►►──fome620s─┬───────────────────────────────────────────────┬──►◄
             │  ┌─ -HELP ──────────────────────────────────┐  │
             │  │                                          │  │
             │  ▼ ┌────────────────────────────────────────┐│
             └────┤ ├─ -PORT ──<port_number>──────────────┤┘
                    ├─ -SERVICENAME ──<service_name>───────┤
                    ├─ -PLAN ──<plan_name>─────────────────┤
                    ├─ -LOGDIR ──<log_dir>─────────────────┤
                    ├─ -CHECK ──<db2_subsystem>────────────┤
                    ├─ -MFS ──<filesize>───────────────────┤
                    ├─ -MFN ──<filenumber>─────────────────┤
                    ├─ -MFNAME ──<message_file_name>───────┤
                    └─ -FLUSH ─────────────────────────────┘
```
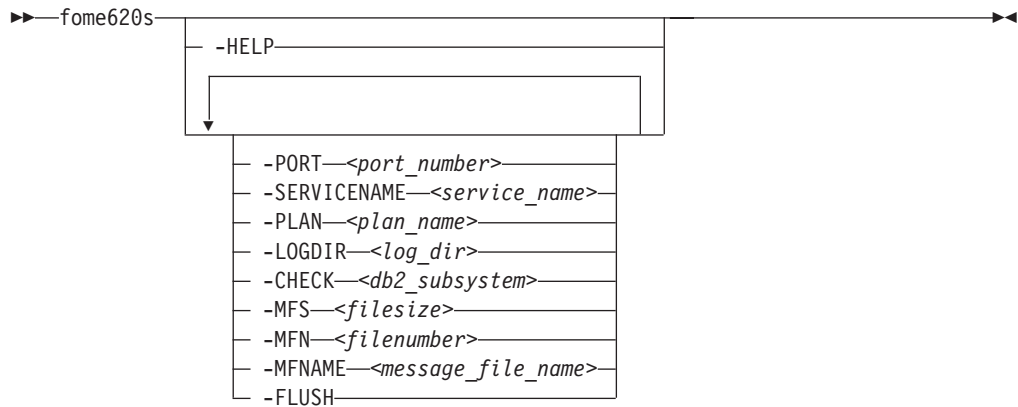
- *-HELP*

  Writes the message ICLS1000I to STDERR. This message gives a short description of the ICLI server command line options.

  Since *-HELP* cannot be used together with any of the other command line options, it cannot be used in regular start scripts or jobs. When you start the server with *-HELP* as required, the ICLI server only starts to write the message ICLIS1000I and stops again immediately.

- *-CHECK* <db2_subsystem>

  Specify this parameter to check a DB2 subsystem for use by the ICLI server. The check will do the following:

  1. Try to connect to a DB2 subsystem using the name <db2_subsystem>.
  2. Open the plan which was specified in the *-PLAN* parameter or the default plan FOMEP620 (FOMEP640).
  3. Try to issue an EXEC SQL SET CURRENT SQLID=<SCHEMA>, where <SCHEMA> is either the value of the environment variable *ICLI_SAP_SCHEMA* or SAPR3 (if the environment variable was not set).
  4. Start the monitor trace if not already started.
  5. Query the DB2 installation parameters using the Instrumentation Facility Interface (IFI).
  6. Fetch the appropriate ASCII-EBCDIC and EBCDIC-ASCII codepage translation tables.
  7. Check that the codepage translation tables are bijective (the mapping is the same in both directions)

  A message is printed to STDERR containing information on whether or not the check was successful. If it was not successful, STDERR will contain an error message.

  Use this option to check the following:

  - The ICLI server is installed and can access its log directory.
  - DB2 modules can be loaded by the ICLI server.
  - The DB2 subsystem is up and accessible.
  - RRS is started.
  - The DB2 plan used by the ICLI server is accessible.

- The ICLI server can set its SQLID to the specified schema.
- ICLI can use the Instrumentation Facility Interface of DB2 to query the DB installation parameters
- The ICLI server can retrieve the codepage translation tables from DB2.
- The codepage translation tables are bijective (the mapping is the same in both directions)

This option allows you to quickly rule out any non-connection related problems without starting an SAP application.

- *-FLUSH*

This option changes the behavior of ICLI tracing. When specified, writing to the trace file is done unbuffered. This means that the trace file will always contain the latest trace record written.

Use this option only when needed, because it adds significantly to the performance impact of tracing.

When this option is not specified, trace output is written buffered to the trace file. Trace records appear in the trace file when the file output buffers are flushed or when a thread terminates normally.

- *-LOGDIR* `<log_dir>`

Specifies the directory for the ICLI server's output files. The directory may be specified with or without a trailing slash ("/"). The default is the current directory of the ICLI server process. The ICLI server's output files are listed in Table 20 on page 130.

- *-MFN* `<filenumber>`

Specifies the maximum number of ICLI message log files. Values greater than zero and the value -1 (for unlimited, except for restrictions imposed by *MAXFILEPROC*; see also "Configuring the ICLI server and SAP application server on z/OS" on page 40) can be specified. If the parameter is omitted, a value of 1 (i.e., only one file) is taken.

For more information on the parameter *-MFN*, see "ICLI server message reporting" on page 117.

- *-MFNAME* *<message_file_name>*

Specifies the file name identifier used for ICLI message log files and ICLI trace files. If specified, the file name of the ICLI message log file has the following appearance:
```
ICLI.<message_file_name>.msg<n>
```

If omitted, the following file naming convention will be used:
```
ICLI.<pid>.msg<n>
```

where `<pid>` is the UNIX System Services process ID of the running ICLI server.

For more information on the parameter *-MFNAME*, see "ICLI server message reporting" on page 117.

- *-MFS* `<filesize>`

Specifies the maximum size (in megabytes) of the ICLI message log file(s). Values greater than zero and the value -1 (for unlimited, except for restrictions imposed by *MAXFILESIZE*; see also "Configuring the ICLI server and SAP application server on z/OS" on page 40) can be specified. If this parameter is omitted, a value of -1 (i.e. "unlimited") is taken.

For more information on the parameter *-MFS*, see "ICLI server message reporting" on page 117.

- *-PLAN* `<plan_name>`

Specifies the plan name used to connect to DB2. If the parameter is omitted, the default plan name `FOMEP620` (`FOMEP640`) will be used when connecting to DB2.

- *-PORT* `<port_number>`

  Specifies the ICLI connection port of the ICLI server. This port number overrides any port number derived from service names (see the description of the parameter *-SERVICENAME* below). The selection and definition of connection ports is described in detail in "The ICLI connection port" on page 99.

- *-SERVICENAME* `<service_name>`

  Specifies a service name that the ICLI server should use to find the ICLI connection port used to listen to connection requests by an SAP program instead of the default service name `fome620s` (`fome640s`). The service name is looked up first in the `/etc/services` HFS file, and if this file does not exist, in the MVS data set `<TCPIP>.ETC.SERVICES` and finally in the data set `<SAPSID>ICLI.ETC.SERVICES` (`<SAPSID>ICLI` is the user ID running the ICLI server, since the high-level qualifier of this data set is always the user ID running the ICLI server).

Example (for 6.20):
```
fome620s -SERVICENAME mysapr3 -PLAN ICLISRV1 -LOGDIR /u/iclirun/log
```

The selection and definition of connection ports is described in detail in "The ICLI connection port" on page 99.

Adding ICLI server instances

To add an additional ICLI server instance to an existing set of instances, perform the following tasks:

1. Choose the connection port number and service name for the new ICLI server instance:
   - If you define a new ICLI server instance on a new z/OS system, you can use the same port number and service name as on your other z/OS systems.
   - If you add an additional ICLI server instance on your existing z/OS system, you need to choose a new port number and a new service name (for details, see "The ICLI connection port" on page 99).
2. If you use a new port and service name, you have to add them as new entries to the TCP/IP services file or data set where you defined the port numbers and service names for the existing ICLI server instances (see "The ICLI connection port" on page 99).
3. Create a new working directory in the z/OS HFS (Hierarchical File System) at the same directory level as the working directories used for the existing ICLI server instances.
4. Decide whether you want to have a new z/OS user ID to run the new ICLI server instance. Reasons to do so might be performance monitoring or work load balancing.
5. Create a new JCL job, UNIX System Services shell script, or started task with a new name, depending on how you have chosen to start the existing ICLI server instances. If you start the ICLI server as a started task, associate the new JCL job with the user ID that runs the new ICLI server instance.

   You need to specify the new port in the invocation of the new ICLI server instance. It is recommend that you do this using the *-SERVICENAME* option of

the ICLI server rather than the *-PORT* option (see also "The ICLI connection port" on page 99 and "ICLI server command line options" on page 101).

6. Decide which application servers should use the new ICLI server instance. On each of them, change the port number in the entry *sapdb2<SAPSID>* in /etc/services (AIX, Linux, and Solaris) and C:\WINNT\system32\drivers\etc\SERVICES (Windows) to point to the new port number.

Starting the ICLI server

There are three ways to start the ICLI server:
- Starting a JCL job or procedure as a started task.
- Submitting a JCL job to batch.
- Executing a UNIX System Services shell script.

A template for each of these start procedures is generated when you use the ICLI server setup program fome620i (fome640i) to set up an ICLI server. For a description of the start procedure templates and for information on how to customize them, see section "Configuring the start procedures for the ICLI server" on page 66.

Stopping the ICLI server

The main reasons for stopping the ICLI server, namely, the *ICLI server address space*, are that you wish to (re)configure your SAP System or perform maintenance work on your hardware or software (including the installation of new ICLI PTFs) on the z/OS image in which the ICLI runs.

If you stop the ICLI server while your SAP system is still running, SAP work processes can no longer access the database via this ICLI server and SAP transactions might be interrupted and manual SAP operator invention may become necessary.

> **Note:**
> Be aware that stopping an ICLI server will potentially disrupt any SAP work in progress. To avoid any disruptions, we recommend moving the ICLI server's workload before initiating a stop. This can be achieved by employing the capability of the SAP application server to proactively redirect work processes, which allows you to switch to a different DB2 host during operations via SAP transaction ST04. For details on the proactive redirection of application servers, see the *SAP Database Administration Guide* and *High Availability for SAP on zSeries Using Autonomic Computing Technologies*.
>
> If you cannot employ proactive redirection for any reason, we recommend using the smooth stopping method, because it is less disruptive on any remaining SAP transactions than the abrupt stopping method. The stopping methods are described below.

If you have to stop the ICLI server, you can choose between two methods:
- The *smooth stopping method* should be employed if you cannot shift the ICLI server's load before initiating the stop, because it has a less disruptive effect on any remaining SAP transactions than the abrupt stopping method. The smooth stopping method comes in two variations: *not forced* and *forced*.

- The *abrupt stopping method* should only be used if you are certain that stopping the ICLI server will not have an impact on ongoing SAP transactions.

The stop methods and the resulting behavior are described in detail in the subsequent sections. The following table gives you an overview of the stop behavior that occurs when a specific stop method is used.

Table 12. ICLI server stop methods and related stop behavior

Method	ICLI accepts new connections	Action on existing ICLI threads		
		Threads which have completed their DB2 logical unit of work	Threads which have not completed their DB2 logical unit of work	
			Before stop timeout	After stop timeout
Smooth + not forced	No	Thread terminates immediately	Thread continues until DB2 COMMIT or DB2 ROLLBACK, then terminates	Thread continues until DB2 COMMIT or DB2 ROLLBACK, then terminates
Smooth + forced	No	Thread terminates immediately	Thread continues until DB2 COMMIT or DB2 ROLLBACK, then terminates	Current statement is interrupted, thread terminates immediately even if it is in DB2.
Abrupt	No	Thread terminates immediately	Current statement is interrupted, thread terminates immediately even if it is in DB2.	n/a

> **Caution:**
> This function does not guarantee that the ICLI server will stop an SAP transaction in a consistent state at the end of an SAP Logical Unit of Work (LUW). This means that—contrary to what the name suggests—even the *smooth* stopping method can have the effect that SAP transactions are interrupted. Candidate transactions are multi-step dialog transactions and long-running batch jobs. Note that interrupted batch jobs generally need manual SAP operator intervention to take appropriate actions afterwards in order to complete the interrupted work.
>
> The reason for this behavior is explained by the following facts:
> - An SAP LUW can span several DB2 LUWs and, therefore, a DB2 point of consistency (DB2 COMMIT or ROLLBACK) within the SAP LUW does not guarantee an SAP point of consistency.
> - The ICLI server is only able to recognize the end of a DB2 LUW but not that of an SAP LUW.

The smooth stopping method

The smooth stopping method allows SAP work processes to continue with their work until a DB2 point of consistency (DB2 COMMIT or DB2 ROLLBACK) is reached. This can be useful, for example, when you want to apply maintenance to your DB2 or z/OS by switching the SAP workload from your default z/OS host to a standby host in a z/OS sysplex. This switch should then be unnoticeable to most of the SAP dialog users. Note, however, that the preferred method for switching an SAP workload away from an ICLI server should be the sysplex failover

mechanism. After such a failover is complete, there should be no remaining ICLI work threads and you can safely stop the ICLI server using the smooth stopping method, for example.

For more information on the sysplex failover mechanism, refer to the SAP online documentation *BC SAP High Availability*, which is available in the SAP Library or at `http://service.sap.com/ha`, and to the *SAP installation guides*.

A smooth stop is characterized by the following behavior:
- The ICLI server will not accept new connections (i.e., new SAP work processes will not be able to connect to this ICLI server).
- All threads which have completed their DB2 LUW (at a DB2 COMMIT or DB2 ROLLBACK point) will be stopped.
- Active threads have a chance to continue working until they reach a DB2 COMMIT or DB2 ROLLBACK point.

Initiating a smooth stop

In order to initiate a smooth stop of the ICLI server, you must first identify the UNIX System Services process ID (PID) or the job name of the ICLI server (see "Identifying SAP work processes serviced by ICLI server threads" on page 110). You can then choose between the following two alternatives, depending upon whether you are working from the UNIX System Services shell or the z/OS console or other equivalent functions:

UNIX System Services shell:
> In the UNIX System Services shell, enter:
> ```
> kill -2 <PID>
> ```
>
> For the smooth stopping method, the parameter `-2` is essential.
>
> The actual stop behavior is determined by the environment variables *ICLI_STOP_TIME* and *ICLI_STOP_MODE*. If you want to exploit the smooth stopping method, you should check if their default values meet your requirements or should be adjusted *before* the ICLI server is started:
> - *ICLI_STOP_TIME* determines for how many seconds the ICLI server will try to stop smoothly. The default value is 300 (seconds).
> - *ICLI_STOP_MODE* determines how the ICLI server behaves when the specified *ICLI_STOP_TIME* has elapsed.
>   *ICLI_STOP_MODE* can be set to the following values:
>   - `-1` means that the remaining threads are stopped. 1 is the default.
>   - `-2` means that the remaining threads are allowed to continue to run.

On the z/OS console or using other equivalent functions:
> On the z/OS console or from other equivalent functions, enter the following command to send a message text to a running job:
> ```
> MODIFY <JOBNAME>,appl=<TEXT>
> ```
>
> The message <TEXT> is a character string for which a special format is expected: It begins with the command (in this case, **STOP**), after which parameters (in this case, *MODE* and *TIMEOUT* and their values) can follow. Possible strings in the <TEXT> character string are presented in the following syntax diagram:

```
►►── STOP ──┬──────────────────────────┬──►◄
            ├─ MODE─(FORCE | NOFORCE)──┤
            └─ TIMEOUT─(<timeout>)─────┘
```
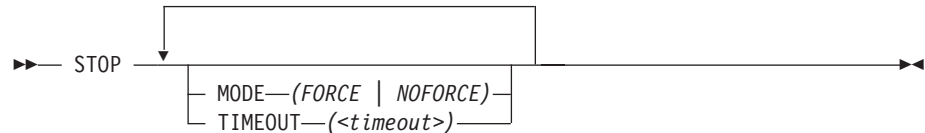
where *(<timeout>)* is a positive number specifying seconds.
For example:

```
MODIFY MYJOB,appl=STOP MODE(FORCE) TIMEOUT(60)
```

If these two options are not specified, their defaults are taken. In the case of *MODE*, the default is <u>FORCE</u>; in the case of *TIMEOUT*, it is <u>300</u> seconds.

If the command (**STOP**), the options (*MODE* and *TIMEOUT*), or their settings are wrong, the **STOP** command will be ignored and the ICLI server will continue. This gives you the chance to try it again with correct values. If *ICLI_MSGLEVEL* is set to I or W, you can find information about what went wrong in the ICLI server message file.

Using the *TIMEOUT* option, it is possible to specify the time in seconds during which ICLI will try to stop smoothly. Using the *MODE* option, is it possible to specify the mode, i.e. the behavior after the specified time has elapsed (*MODE*(FORCE) means that the remaining threads are stopped, while *MODE*(NOFORCE) means that the remaining threads are allowed to continue running).

If you stop the ICLI server from the z/OS console, the environment variables *ICLI_STOP_TIME* and *ICLI_STOP_MODE* will be ignored.

Stop behavior after the defined timeout is reached

When all ICLI server threads reach a DB2 COMMIT or DB2 ROLLBACK point within the timeout period (specified by *ICLI_STOP_TIME* or *TIMEOUT*), a successful smooth stop occurs. In this case, message ICLS1097I is generated and appears in the ICLI message log file.

If any threads are still active when the timeout period expires, message ICLS1098I is generated and the following behavior occurs:

- If the option *MODE*(FORCE) was specified or *ICLI_STOP_MODE* was set to 1, the remaining threads and active DB2 threads are stopped and the ICLI server terminates, which may have disruptive effects upon the SAP work processes connected to these threads.

> **Note:**
> The actual time needed until a stop is complete is not determined by the values assigned to *TIMEOUT* or *ICLI_STOP_TIME* alone but also depends on the cleanup work (for example: rollbacks) that may be performed.

- If the option *MODE*(NOFORCE) was specified or *ICLI_STOP_MODE* was set to 2, the active threads will continue to run (after the timeout) until they reach a DB2 COMMIT or DB2 ROLLBACK point. The ICLI server will not terminate until all working threads have reached one of these DB2 states. If you need to stop the ICLI server while there are still running threads, re-issue the command **MODIFY** with the option *MODE*(FORCE) to stop the ICLI server immediately or use the abrupt stopping method described below.

The abrupt stopping method

If you stop the ICLI server with the commands described below, you should be aware that this can have disruptive effects on ongoing SAP work.

The following steps need to be performed to stop the ICLI server using the abrupt stopping method:

1. Stop the ICLI server using one of the following commands:

    UNIX System Services shell:
    > In the UNIX System Services shell, enter:
    > ```
    > kill <PID>
    > ```

    On the z/OS console or using other equivalent functions:
    > On the z/OS console or from other equivalent functions, enter:
    > ```
    > MODIFY BPXOINIT,TERM=<PID>
    > ```
    >
    > If the ICLI server has been started as a started task, you may also use the **STOP** command:
    > ```
    > STOP <JOBNAME>,A=<ASID>
    > ```
    >
    > Here, the job name <JOBNAME> and the address space identifier <ASID> have to be specified.

2. To identify the UNIX System Services process ID of the ICLI server, see "Identifying SAP work processes serviced by ICLI server threads" on page 110.

3. Verify that the ICLI server has stopped.

Caution if using z/OS V1R3 or earlier:

(**Note:** The following no longer applies as of z/OS V1R4, because **kill -9** no longer causes the ICLI server to hang.)

Regardless of the method you choose to stop the ICLI server, do *not* use the command **kill -9** because it will almost always cause the ICLI server to hang. The reasons for this are as follows. The ICLI server uses the FastPath implementation of TCP/IP. The implementation of the FastPath system call **recv()** does not currently respond to a **kill -9** signal. That is, an ICLI server with a thread waiting in system call **recv()** for the next request from the ICLI client will not stop. If you already attempted to stop an ICLI server with **kill -9**, which will not work under the circumstances mentioned above, you can shut down the server with one of the following procedures:

*   **CANCEL** the ICLI server address space.
*   Explicitly stop every ICLI server thread located in recv() by issuing the command **netstat -D <connection>**. The value of the connection for this can be clearly seen from the screen by issuing the command **netstat -c**.
*   Stop all SAP application servers that communicate via the ICLI server with DB2.

A **MODIFY** (which can be abbreviated with an **F**) with the *FORCE* option has the same effect as the **kill -9** command. You should therefore not use it either.

On the other hand, a **MODIFY** with the *TERM* option *can* be used, and will cause the ICLI server to get control, clean up, and write a message.

Identifying SAP work processes serviced by ICLI server threads

You can get information about all of the ICLI server threads of a currently running ICLI server. This information lets you identify which SAP work processes are served by an ICLI server and on which application server they are running.

First, you need to identify the UNIX System Services process ID (PID) of the ICLI server. This is done as follows:

UNIX System Services shell:
In the UNIX System Services shell, enter the **ps -ef** command.

On the z/OS console or using other equivalent functions
On the z/OS console or from equivalent functions, enter the following command:

DISPLAY OMVS,U=<userid>

or

DISPLAY OMVS,ASID=ALL

Then issue the following command:

DISPLAY OMVS,PID=<pid>

The response might look like this (example for 6.20):

```
0290  D OMVS,PID=35
0090  BPXO040I 13.56.28 DISPLAY OMVS 541
0090  OMVS     000E ACTIVE
0090  USER     JOBNAME  ASID       PID       PPID STATE   START        CT_SECS
0090  VSCH     VSCH7    02F2        35          1 HK    15.30.29        39.857
0090   LATCHWAITPID=       0 CMD=fome620s -PORT 5443 -PLAN FOMEP620    -FLUSH
0090   THREAD_ID      TCB@   PRI_JOB  USERNAME   ACC_TIME SC  STATE
0090   08BDB73800000002 008E0478 OMVS                     .103 IPT  KU
0090   TAG=ICLI server 1.0 for SAP Version 6.20
0090   08BDBFE000000003 008E0068 OMVS                     .144 GNM JY
0090   TAG=ICLI server control thread
0090   08BC90E800000026 008C0060 OMVS                    2.995 PTT JY    V
0090   TAG=ICLI client    10.199.172.72  PID    12410 type  UPDATE
0090   08BCA23800000028 008D6B58 OMVS                     .524 PTT JY    V
0090   TAG=ICLI client    10.199.172.72  PID    40056 type  DIALOG
0090   08BCAAE000000029 008C0E88 OMVS                     .109 PTT JY    V
0090   TAG=ICLI client    10.199.172.72  PID    26498 type ENQUEUE
0090   08BCC4D80000002A 008C0668 OMVS                     .261 PTT JY    V
0090   TAG=ICLI client    10.199.172.72  PID    24452 type   BATCH
0090   08BC99900000002F 008D6108 OMVS                     .818 PTT JY    V
0090   TAG=ICLI client    10.199.172.72  PID    30700 type  DIALOG
```

In this example, the ICLI server's UNIX System Services process ID is 35. For each thread of process 35, two lines are displayed. The first line lists the current values of

' THREAD_ID   TCB@   PRI_JOB USERNAME   ACC_TIME SC STATE'

of a thread and the second line displays a string further detailing that thread. In the example, the strings for the first two threads indicate that the first is the main thread of the ICLI server, while the second is the control thread of the ICLI server.

The detail strings for the third and following threads comprise three parts, for example:

1. ICLI client 10.199.172.72
2. PID 24452
3. type BATCH

The detail strings contain the following information:

ICLI client
> The IP address of the SAP application server running the corresponding SAP work process of the ICLI server work thread.

PID The process ID of the SAP work process on the SAP application server corresponding to this ICLI server work thread.

type The SAP type of the work process on the application server.

This information should suffice to identify the SAP work processes an ICLI server work thread is currently working for, and the SAP application server on which they are running.

Identifying DB2 threads related to SAP work processes

Each individual SAP work process has a one-to-one relationship with a DB2 thread. For better identification of SAP work processes, ICLI server threads, and DB2 threads, additional information is stored in DB2 thread fields. This information allows you to correlate the three different entities in an easy way.

To retrieve information about the DB2 threads, issue the following DB2 command:
```
DISPLAY THREAD (*)
```

A sample output follows:
```
DSNV401I -D514 DISPLAY THREAD REPORT FOLLOWS -
DSNV402I -D514 ACTIVE THREADS - 361
NAME     ST A   REQ ID          AUTHID  PLAN     ASID TOKEN
RRSAF    T  *  8071 010199172133 FUNK    MF620000 0019   85
 V437-WORKSTATION=3 0 007 0000039896, USERID=0A35D4100000000A,
      APPLICATION NAME=merian
DISPLAY ACTIVE REPORT COMPLETE
```

In this example, the DB2 subsystem name is D514. The meaning of each label and the associated column is described in "Monitoring and controlling DB2 and its connections" in the *DB2 for OS/390 and z/OS Administration Guide*. However, some columns are used other than described to map SAP work process-related information.

The first line of the displayed thread in the sample output above contains the current values of:
```
NAME     ST A   REQ ID          AUTHID  PLAN     ASID TOKEN
RRSAF    T  *  8071 010199172133 FUNK    MF620000 0019   85
```

This thread is currently active within DB2 as indicated by the '*' in column A. The ID field contains the IP address of the corresponding application server on which the respective work process is running (10.199.172.133). The dots are omitted because of the character limitation. The name of the plan with which the corresponding ICLI server is running is MF620000.

The second line of the displayed thread contains the following fields:
```
V437-WORKSTATION=3 0 007 0000039896, USERID=0A35D41000000008,
```

The WORKSTATION field is used to store WP-related information. The division of this field into separate subfields is as follows:

Table 13. Format of WORKSTATION field

Bytes	Content
1	WP type
2	Blank
3	Connection handle
4	Blank
5–7	WP number
8	Blank
9–18	WP ID

WP type shows the type of the related work process. The numbers can range from 0 to 6, and refer to the following work process types:

0   GENERIC (not classified)
1   DIALOG
2   BATCH
3   UPDATE
4   ENQUEUE
5   SPOOL
6   UPDATE2

Connection handle represents the connection handle of the database connection. Besides the primary connection, SAP work processes can create up to 5 secondary connections. The connection handle is a single digit that uniquely identifies each connection of a work process. For more information on secondary database connections, see SAP Note 160484.

WP number represents the work process number within the SAP system.

WP ID contains the work process identifier, which is the PID in AIX, Linux, Solaris, or Windows. Numbers in both fields are decimal.

The USERID field contains the corresponding ICLI server thread ID related to this DB2 thread.

The APPLICATION NAME field (third line) contains the host name of the associated application server.

The information displayed via DISPLAY THREAD can also be accessed via the SAP Performance Monitor. For more details, see the section "Thread Activity" in the *SAP Database Administration Guide*.

Stopping individual ICLI server threads

Individual ICLI server threads are usually stopped in order to resolve a problem affecting only a single SAP work process associated with that thread or because of connectivity or resource problems encountered by an individual thread. This operation has no impact on other SAP work processes or other ICLI server threads. It also does not affect the ability of the ICLI server address space to accept new connection requests and create new ICLI server threads. Stopping an individual ICLI server thread not only frees any z/OS resources held by this thread, but will also terminate the associated DB2 thread, thus freeing the DB2 resources it held.

Stopping an ICLI server thread should always be done with care, since it potentially disrupts an ongoing SAP transaction.

The following situations may make it necessary to stop an individual ICLI server thread:

1. It becomes necessary, for SAP-specific reasons, to restart an SAP work process.
2. An SAP work process "hangs" because the database operation initiated by it does not finish within a reasonable amount of time.
3. An ICLI server thread is no longer connected to an SAP work process and is blocking resources.

Individual ICLI server threads can be stopped using either SAP mechanisms or z/OS means. This is described in the following two sections.

Stopping an ICLI server thread using SAP mechanisms

Whenever possible, when stopping an individual ICLI server thread, *SAP mechanisms should be used*; this is because it is least disruptive to the SAP system. Especially in the first two cases above, this method should be employed. SAP transaction SM50 can be used to identify and stop an SAP work process; this will also stop both the associated ICLI thread and its database connection. If the ICLI server thread was connected to DB2 when it was stopped, message ICLS1065I will be reported in the ICLI message log files. This indicates that the associated DB2 thread was stopped using DB2 CANCEL THREAD.

Stopping an ICLI server thread using z/OS means

It should be only very rarely necessary to stop an individual ICLI server thread using z/OS means, for example when there is some prior error in the ICLI server's thread termination routine. Should it nevertheless become necessary to stop an ICLI server thread in this fashion, you should be aware that you are risking the disruption of an ongoing SAP transaction which this thread is currently processing. In the event of a disruption, DB2 will nevertheless perform a ROLLBACK.

The following steps must be performed to stop an individual ICLI server thread from the z/OS console or equivalent functions:

1. Identify the ICLI server work thread you want to stop (see "Identifying SAP work processes serviced by ICLI server threads" on page 110).
2. Issue the following command

   ```
   MODIFY BPXOINIT,TERM=<pid>.<tid>
   ```

   where `<tid>` is the THREAD_ID as reported by the command

   ```
   DISPLAY OMVS,PID=<pid>
   ```
3. Verify that the ICLI server work thread has terminated.

If the ICLI server thread was in the RCV2 state, **MODIFY** with the *TERM* option will not stop it. You have to use the **MODIFY** command above with the *FORCE* option instead. Note that the **MODIFY** command with the *FORCE* option does not give control to the ICLI thread termination routine.

Stopping the database server without stopping the application servers

If you have to stop the SAP database server for maintenance purposes (such as z/OS, DB2, or ICLI server maintenance), you should try to minimize the impact this has on your SAP system.

In a data sharing environment, this can best be achieved by using the capability to proactively redirect work processes to switch SAP work away from the z/OS system on which you plan to apply maintenance. Once you have completed the maintenance work and the z/OS system and all its necessary components (for example, DB2 and ICLI) are back up, you should also switch back your SAP system. (For details on the proactive redirection of SAP work processes, see the *SAP Database Administration Guide*. For a complete description of proactive redirection, see the *SAP Database Administration Guide* and the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*.

If you cannot proactively redirect the SAP work processes to another DB2 member for any reason, there are still ways of stopping the database server without having to stop the application servers, which will save you the operational work of stopping them and has the advantage that they will not lose their SAP buffers. The following two sections describe how this can be achieved.

Stopping the database server in a non-data-sharing environment

In a non-data-sharing environment, stopping the database server has the effect that *no work is possible* on this SAP system *until the database server is started again*.

To stop the database server, proceed as follows:

1. Choose the right point in time for this operation, since the SAP system will not be available when the database server is stopped.
2. Stop all ICLI server instances by following the steps described in "Stopping the ICLI server" on page 105.
   Use the smooth stopping method, because it minimizes the impact that stopping the ICLI server has on SAP transactions. The work processes on the applications server will go into a reconnect state. This means that they will periodically attempt to reconnect to the same ICLI server instance until the ICLI server becomes available again.
3. Apply maintenance (this may include, for example, IPL of z/OS and stopping DB2).

To restart the stopped database server, proceed as follows:

1. Restart everything except for the ICLI server on the database server (DB2 subsystem, RRS, etc.).
2. Restart the ICLI server instance using the same port number as before.

Stopping the database server in a data sharing environment

In a data sharing environment, stopping one of the database servers has the effect that the capacity of this SAP system is reduced until that database server is started again, but it is still possible to work with this SAP system.

You should use SAP transaction ST04 to switch the application servers, currently connected to the database server that is to be stopped, to one of the remaining database servers. When the switch is complete and no more ICLI server work threads are connected to DB2, you can safely apply maintenance to the database server. Depending on the kind of z/OS maintenance work, you can even keep DB2 and the ICLI server up and running.

If you cannot proactively redirect the SAP work processes to another DB2 member for any reason, you should:

1. Choose the right point in time for this operation, since certain application servers in your SAP system will not be available when the database server is stopped.
2. Stop all ICLI server instances on that database server by following the steps described in "Stopping the ICLI server" on page 105. Use the smooth stopping method, which minimizes the impact that stopping the ICLI server has on SAP transactions. The work processes on the application servers will go into the reconnect state. This means that they will periodically attempt to reconnect to the same ICLI server instance until the database server becomes available again.
3. Apply maintenance (this may include, for example, IPL of z/OS and stopping DB2).

To restart the stopped database server:
1. Restart everything except for the ICLI server on the database server (DB2 subsystem, RRS, etc.)
2. Restart the ICLI server instance using the same port number as before.

Changing the database server of an application server running on UNIX, Linux, or Windows

If you increase DB2 capacity by adding another DB2 data sharing member, you may want to redirect some of the existing application servers on UNIX, Linux, or Windows to the new database server.

To redirect an application server to a different database server:
1. Make sure that the application server can access the new database server via TCP/IP.
2. Make the necessary changes in your SAP profiles and SAP environment such that the application server will access the new database server host.
3. If the ICLI server instance on the new database server uses a different ICLI connection port, you have to change the following entries:

   UNIX/Linux:
   The entry `sapdb2<SAPSID>` in `/etc/services`

   Windows:
   The entry `sapdb2<SAPSID>` in `C:\WINNT\system32\drivers\etc\SERVICES`

ICLI server registration with Automatic Restart Management

In a sysplex environment, a program can enhance its recovery potential by registering as an element of Automatic Restart Management (ARM). ARM can reduce the impact of an unexpected error to an element because MVS can restart it automatically, without operator intervention. For details on how to use ARM, see *z/OS MVS Programming: Sysplex Services Guide*. For details on how to set up ARM policies, see *z/OS MVS Setting Up a Sysplex*.

ICLI supports this restart mechanism by registering as an element with ARM. The default element name used by the ICLI server for registration is `FOME620S` for 6.20 (`FOME640S` for 6.40). If you plan to run multiple ICLI servers, you need to use unique element names for each of the ICLI servers that you start. The environment variable *ICLI_ARM_ELEMNAME* can be used for this purpose. In choosing the element name, make sure that you follow the ARM element naming rules as described in *z/OS MVS Sysplex Services Reference*:

• Valid characters are:
– Uppercase alphabetic characters.
– The numbers 0 through 9.
– $, #, @, and underscore (_).
• The first character must not be a number.
• The name must be 16 characters long, padded on the right with blanks.
• The name must be unique across the sysplex.

> **Note:**
>
> ARM can be used only for started tasks or batch jobs. If you start the ICLI
> from the UNIX System Services shell, you will get the following ICLI
> warning message:
>
> ```
> ICLS1106W
> Registration with Automatic Restart Manager (ARM)
> for element name <<sss>> failed:
> return code = <nnn>, reason code = <nnn> (HEX: <xxx>)
> You will not be able to exploit ARM mechanisms to restart the ICLI server.
> ```
>
> If you do not plan to use ARM, you can safely ignore the warning message
> ICLS1106W, which is issued when the ICLI server is started. ICLI will run
> without further restrictions.

Message reporting capabilities

Both the ICLI client and the ICLI server generate messages which are categorized
as follows:

• **I**: informational message
• **W**: warning
• **E**: error
• **S**: severe error

You can control the message level on the ICLI server side:

• by setting the environment variable *ICLI_MSGLEVEL* before starting the ICLI
  server.
• via SAP transaction ST04 at runtime.

You can choose between the following values:

Table 14. Possible values of the ICLI server environment variable ICLI_MSGLEVEL

Value	These messages will be generated
W	Warnings, errors, and severe errors (default)
E	Errors and severe errors
I	All

In general, the default level of **W** is the recommended choice, but if you need to
restrict the number of messages that are reported, you can achieve this by setting
the message level to **E**. In a test environment or if you need more diagnostic
details (for example, about connect/disconnect attempts on the ICLI server), you
should use the message level **I**.

You will find the complete list of ICLI messages in Appendix D, "ICLI messages and codes," on page 165.

ICLI client message reporting

The ICLI client messages are written to the SAP developer trace files. This mechanism depends on the setting of SAP trace options (parameter *rdisp/TRACE*).

When errors cause the termination of a work process, the ICLI client messages are also written into the SAP system log. Should the pipe from the ICLI client to the SAP developer trace files and the SAP system log be broken, you will find the ICLI client messages which could not be transferred to the SAP side in the ICLI client trace files (see also Table 21 on page 131).

ICLI server message reporting

ICLI server messages issued by one ICLI server process are written into one or more ICLI message log files and, optionally, into the z/OS System Log.

The clocks on the SAP application servers and z/OS database server should be synchronized to simplify the validation and comparison of the timestamp information attached to each message on both sides. The environment variable *TZ* should be set to your z/OS time zone; this will ensure that the ICLI message log timestamps report the correct local time. See also "Configuring the JCL started task" on page 67.

Messages that are written by the ICLI server during startup before the message log file is open for writing are directed to STDERR. If the ICLI server encounters problems when writing into its message log file (for example because of a 'file system full' condition), all further messages will be redirected to STDERR. It is therefore important that you specify a STDERR DD statement in your started task. Otherwise, you will not be able to see such messages. For this reason, we also recommend that you direct STDERR output to a file system which is different from the one into which the ICLI message log is written. See also "ICLI server messages" on page 192.

The names of these files have the form
ICLI.<id>.msg<n>

where <id> is either the value <message_file_name> specified using the command line option *-MFNAME* <message_file_name> or the UNIX System Services process ID of the ICLI server process (the default), and <n> is an ascending number starting with 1 (if you are using the command line option *-MFN* <filenumber> to create more than one ICLI message log file).

The message log files are written to the directory indicated with the *-LOGDIR* option or, if this option is not used, into the current working directory. For started tasks, the current working directory defaults to the home directory of the user associated with the started task.

If you are running multiple ICLI server instances serving different SAP systems, then you can direct the ICLI logs to different file systems; this will avoid filling up the file system when tracing with only one ICLI server.

If the environment variable *ICLI_WRITE_TO_SYSLOG* is set to 1, all ICLI messages are passed additionally to the z/OS System Log. The layout of the message is as follows:

```
BPXF024I (<user-id>) May  3 11:14:21 <thid>:<SSID> <pid>  : <message_id>
<message text>
```

where
- `<user-id>` is the user ID of e.g., the started task (for example: `<SAPSID>ICLI`),
- `<thid>` is the z/OS thid of the corresponding ICLI thread (for example: `0004`) or "NONE" if the message is issued by an ICLI control or main thread,
- `<SSID>` is the database attach name (for example: `SG94`),
- `<pid>` is the ICLI server process ID (for example: `1459617827`),
- `<message_id>` is the ICLI server message ID (for example: `ICLS1420I`) , and
- `<message text>` is the text of the ICLI message.

Thus, an actual message might have the following appearance:
```
BPXF024I (E1EICLI) May 3 11:14:21 0004:SG94 1459617827 : ICLS1420I
 Client sent DISCONNECT, this thread will end.
```

Housekeeping for ICLI message log files

In order to keep the size of the ICLI message log files small and to allow for regular archiving of the message logs, the command line options *-MFN* `<filenumber>` and *-MFS* `<filesize>` should be used.

-MFN `<filenumber>` controls the number of message log files that a single ICLI server process can write during its "lifetime".

-MFS `<filesize>` controls the size of these files.

Note that the values given for *-MFN* `<filenumber>` and *-MFS* `<filesize>` may be further restricted by the UNIX System Services BPXPRMxx parameters *MAXFILESIZE* and *MAXFILEPROC*.

When using the *-MFN* `<filenumber>` and *-MFS* `<filesize>` options, the ICLI will write its messages into the ICLI message log file `ICLI.<id>.msg1` until this file reaches the size limit `<filesize>`. It will then open the next message log file `ICLI.<id>.msg2` and so on until `<filenumber>` is reached. Each ICLI message log file will contain an initial message ICLS1011I indicating how many message log files will be written as a maximum and the current message log file number.

The last message that was written will always be followed by the line
```
----- EOF -----
```

When the last such ICLI message log file reaches its size limit `<filesize>`, wrapping occurs. This means that the first ICLI message log file is overwritten.

The following method should be used to find the ICLI messages in chronological order:
- Search all `ICLI.<id>.msg<n>` files for the occurrence of the EOF line; preceding that line, you have now found the last message that was written by the ICLI server.
- You can now go backward in that file to find older messages.
- When you hit the top of the file, go backwards in file numbers, i.e., to find messages before the ones in `ICLI.<id>.msg3`, go to `ICLI.<id>.msg2`, then to `ICLI.<id>.msg1`, and then wrap to `ICLI.<id>.msg20` (if 20 was your `<filenumber>`).

- Archiving of ICLI message log files can now be done, e.g., by creating regular backups of the inactive ICLI message log files (these are the ones *not* containing the string `----- EOF -----`).
- The command line option *-MFNAME* can also be used to reduce the total size of the ICLI server logs, because then the restart of an ICLI server does not result in a new set of files. The old log files will be overwritten instead. If you want to use this option, be sure to save the old set of log files before restarting the server, if you need them for problem analysis.

Trace capabilities

This section describes the available ICLI server and client trace capabilities.

ICLI server trace

The ICLI server supports the trace levels listed in Table 15 on page 120. All of these trace levels are controlled by the UNIX System Services environment variable *ICLI_TRACE_LEVEL*, which can be either not set or set to one of the values shown. The ICLI server reads this environment variable during its initialization and activates its tracing accordingly. The default trace level (if no environment variable was specified) is 0, which means ″no tracing. At runtime, the trace level can be modified by SAP transaction ST04.

Traces are written to the file `ICLItrace.<ID>.<UNIQUE_ID>` in the log directory of the ICLI server process, where `<id>` is either the value `<message_file_name>` specified using the command line option *-MFNAME* `<message_file_name>` or the UNIX System Services process ID of the ICLI server process in decimal representation and `<UNIQUE_ID>` represents a unique identifier (last four bytes of the UNIX System Services thread ID converted to decimal representation). The log directory of the ICLI server is specified with the command line option *-LOGDIR* *<log_dir>*.

Table 15. ICLI server trace levels

Trace level	Description
0	Tracing completely disabled (default) Should be used for optimal performance.
1	Developer trace: SQLCODE traceback Should be used when doing test runs where you expect errors. For every non-zero SQLCODE, it will produce an SQLCODE trace, which shows the SQL statement producing the error and some previous statements. Performance impact is small since traces are written into memory and dumped into the trace files only when an error occurs. Traceback information can be valuable when tracking down error situations.
2	SQL trace of statements and parameter markers Should be used if you want a list of all SQL statements that are executed and to find errors in input data to the SAP database. Can also be used to get some timing values because a timestamp is written for each SQL statement. For the format of this SQL trace, see "Format of the SQL trace" on page 121.
3	Same as trace level 2 plus fetched data In addition to the output of trace level 2, fetched data is also traced. This can help in analyzing errors which occur while retrieving data from the database. The amount of trace data is significantly higher than with trace level 2. For the format of this SQL trace, see "Format of the SQL trace" on page 121.
4, 5, 6	These are internal ICLI trace levels. They will have a severe impact on the performance of the ICLI server. **Use them only if instructed to do so by IBM/SAP service personnel.**

See Table 20 on page 130 for the output files used for the ICLI server trace.

ICLI client trace

The ICLI client supports the trace levels listed in Table 16 on page 121. They are all controlled by the environment variable *ICLI_TRACE*, which can be either not set or set to one of the values shown. The ICLI client reads this environment variable during its initialization and activates its tracing accordingly. The default trace level (if no environment variable was specified) is 0, which means "no tracing". At runtime, the trace level can be modified by using SAP transaction ST04.

Traces are written to the client trace files described in Table 21 on page 131).

> **Note:**
> All client traces have a significant impact on performance and can generate huge trace files.

Table 16. ICLI client trace levels

Trace level	Description
0	Tracing is disabled. Trace files are only written in case of severe errors which cannot be piped to the SAP database layer.
1	CLI layer trace. Logs all CLI calls and their return codes.
2	Same as trace level 1 plus some additional status information.
3	Same as trace level 2 plus ICLI performance monitor trace.
4	Same as trace level 3 plus memory management trace.
5	Same as trace level 4 plus communication trace.

ICLI client and server communication trace

When you set the environment variable *ICLI_COM_TRACE* to 1 (see Table 26 on page 159) for the ICLI server or client, they write a detailed communication trace to the output files described in Table 20 on page 130 and Table 21 on page 131.

This trace feature is used for diagnostic purposes. There is a substantial performance degradation when this trace is active.

At runtime, the ICLI server communication trace can be started and stopped from the SAP side by SAP transaction ST04.

Format of the SQL trace

The SQL trace for *ICLI_TRACE_LEVEL* set to 3 is written in the following format:

```
ICLI Server SQL trace
OS/390 Version:        3.12
ICLI server for SAP Release:  6.20
ICLI server protocol version: 2
ICLI server internal version: 0
ICLI Comm. Protocol:       TCPIP
started with port 30507.
Compile Timestamp: Feb 19 2002 09:32:45 Build level:
Trace started at Mon Mar 18 10:40:29 2002
Thread id is: 1639C300 00000005
Workprocess-ID = 0000671786, Workprocess_class  = 0, name = GENERIC
--timestamp
|            --stmt id
|            |  --function
|            |  |                   --sqlcode
|            |  |                   |   --elapsed DB2 time(seconds)
|            |  |                   |    |   --data
|            |  |                   |    |   |
|            |  |                   |    |   |
V            V  V                   V    V   V
10:40:29.583    CONNECT             0   0.002588
10:40:29.595    EXECUTE_IMM         0   0.001119 SET CURRENT SQLID='SAPR3'
10:40:29.596    COMMIT              0   0.000235
10:40:29.598    EXECUTE_IMM         0   0.000668 SET CURRENT PRECISION = 'DEC15'
10:40:29.600    EXECUTE_IMM         0   0.000449 SET CURRENT DEGREE = '1'
10:40:29.608 000 PREPARE            0   0.000116 SELECT TRANSTAB FROM SYSIBM.SYSSTRINGS WHERE
                INCCSID=500 AND OUTCCSID= 819 FOR FETCH ONLY WITH UR
10:40:29.609 000 OPEN_CURSOR        0   0.000043   NO HOLD
10:40:29.609 ----fetched row -------->   0.000266 (V256)'0100000102039C09867F978D8E0B0C0D0E0F
                101112139D8508871819928F1C1D1E1F80818283840A171B88898A8B8C050607909116939495
                960498999A9B14159E1A20A0E2E4E0E1E3E5E7F15B2E3C',
10:40:29.612 000 FETCH              0   0.002697      1/1
10:40:29.617 000 CLOSE_CURSOR       0   0.000041   NO HOLD
10:40:29.617 001 PREPARE            0   0.000089 SELECT TRANSTAB FROM SYSIBM.SYSSTRINGS WHERE
                INCCSID=819 AND OUTCCSID= 500 FOR FETCH ONLY WITH UR
10:40:29.618 001 OPEN_CURSOR        0   0.000031   NO HOLD
10:40:29.618 ----fetched row -------->   0.000089 (V256)'010000010203372D2E2F1605250B0C0D0E0F1
                01112133C3D322618193F271C1D1E1F404F7F7B5B6C507D4D5D5C4E6B604B61F0F1F2F3F4F5F6
                F7F8F97A5E4C7E6E6F7CC1C2C3C4C5C6C7C8C9D1D2D3',
10:40:29.621 001 FETCH              0   0.002513      1/1
10:40:29.623 001 CLOSE_CURSOR       0   0.000044   NO HOLD
10:40:29.623 002 PREPARE            0   0.000088 SELECT DUMMY FROM "#LOB" FOR FETCH ONLY WITH UR
10:40:29.624 002 OPEN_CURSOR        0   0.000040   NO HOLD
10:40:29.624 ----fetched row -------->   0.000154 (V1/819)'5',
10:40:29.625 002 FETCH              0   0.000389      1/906
10:40:29.626 002 CLOSE_CURSOR       0   0.000043   NO HOLD
10:40:29.628    COMMIT              0   0.000258
10:40:29.859 003 PREPARE            0   0.004430 SELECT VERSION FROM SVERS  FOR FETCH ONLY WITH UR
10:40:29.860 003 OPEN_CURSOR        0   0.000045   NO HOLD
10:40:30.035 ----fetched row -------->   0.001889 (V72/819)'610                           ',
10:40:30.036 003 FETCH              0   0.002205      1/1
10:40:30.039 003 CLOSE_CURSOR       0   0.000070   NO HOLD
10:40:30.041    ROLLBACK            0   0.000103
10:40:30.042    DISCONNECT          0   0.000000
```

The header and the columns of the trace file contain the following information:

Information in the header of the trace file

The header contains the version number of the ICLI server and the start time (and its equivalent processor timestamp in milliseconds). It also tells you the UNIX System Services thread ID of the ICLI work thread, the SAP work process ID, and its type. This allows you to relate ICLI server trace files to SAP work processes and to UNIX System Services threads.

timestamp

The current timestamp when the DB2 SQL statement was executed.

stmt id

In the case of the function 'fetch result set' (FETCH_RES_SET), this column will contain the result set number being fetched. In the case of all other functions, this column contains the statement ID, which is a decimal number uniquely identifying the SQL statement in the DBIF statement cache of each work process. The statement IDs are unique only within a work process, not across work processes. This cache currently contains 316 entries, so the statement IDs currently range from 000 to 315. Since the ICLI exploits DB2's Dynamic Statement Cache, you must sometimes go backward beyond the last COMMIT in the trace to find the

PREPARE function that belongs for example to an OPEN_CURSOR function. Take the statement ID of OPEN_CURSOR and search backward in the trace for the previous PREPARE statement having the same ID.

function

The second column contains the ICLI server function that is executed. In most cases, this corresponds directly to a DB2 SQL statement. For example: OPEN_CURSOR corresponds directly to an EXEC SQL OPEN CURSOR ... statement. In some cases, this function corresponds to other functions of the ICLI server (CONNECT, DISCONNECT).

The following Table 17 shows the SQL trace format for all ICLI server functions.

Table 17. ICLI SQL trace format

ICLI server function	Output in data field
CLOSE_CURSOR	WITH HOLD or NO HOLD indicating if the cursor was defined with or without hold
COMMIT	- none -
CONNECT	- none -
DESCRIBE	The SQL statement being described
DISCONNECT	- none -
EXECUTE	Rows = number of rows affected by the operation (INSERT, UPDATE, and DELETE) If trace level was 3, each individual EXECUTE is traced with its corresponding input data and SQLCODE.
EXECUTE_IMM	Statement text as it is passed to DB2
FETCH	Number of rows fetched / number of rows requested, and fetched data. For *ICLI_TRACE_LEVEL*=3, fetched data precedes the FETCH line and starts with ------------------>".
FETCH_RES_SET	Number of rows fetched / number of rows requested for this result set of the stored procedure call and of the fetched data. For *ICLI_TRACE_LEVEL*=3, fetched data precedes the FETCH line and starts with ------------------>
LOB_GET_LENGTH	`lob_length is <value>`
LOB_FREE_LOCATOR	- none -
OPEN_CURSOR	WITH HOLD or NO HOLD plus the input data
PREPARE	Statement text as it is passed to DB2
ROLLBACK	- none -
STORED_PROC	input data and rc of STORED_PROC

sqlcode

In this column, you will find the SQLCODE of the DB2 statement executed during the ICLI server function.

The case of the PREPARE function is special because it always triggers the DB2 statements PREPARE and DECLARE CURSOR. This means that a negative SQLCODE in the PREPARE function can refer to either a failing PREPARE statement or a failing DECLARE CURSOR statement.

A FETCH function will usually result in more than one FETCH DB2 statement. A non-zero SQLCODE in FETCH means that the last of these FETCH statements returned that SQLCODE.

A 'fetch result set' (FETCH_RES_SET) function will usually result in more than one FETCH DB2 statement. A non-zero SQLCODE in FETCH_RES_SET means that the last of these FETCH_RES_SET statements returned that SQLCODE.

elapsed DB2 time (seconds)

This column gives an elapsed time for the SQL function being executed. For array INSERTs or FETCHes, the accumulated elapsed time is reported. The value allows rough estimates for the runtime of DB2 SQL requests as perceived by the ICLI server code. It is not a replacement for exact DB2 performance measurements using a DB2 performance monitor.

data

Contains varying data depending on the ICLI server function. Table 17 on page 123 lists the functions and their data.

data input and output format

Values of parameter markers and values of fetched columns are written to the trace file in readable format.

Variable length strings (DB2 types VARCHAR or LONG VARCHAR) are prefixed by V<nnn> where <nnn> is the length of the column. The V<nnn> prefix can be followed by /<NNN> where <NNN> is the ASCII Coded Character Set Identifier (CCSID) used to access character data.

Character data which is declared FOR BIT data is written in hexadecimal format. A maximum of 80 hex values are displayed.

Character data which is not FOR BIT data is displayed as text. To make ASCII text readable on z/OS, an ASCII to EBCDIC (ISO8859-1 to IBM-1047) translation is applied before the strings are written to the trace. A maximum of 256 characters are displayed.

Character and binary LOB locators (DB2 types CLOB_LOCATOR and BLOB_LOCATOR) are 4 byte pointers and appear as follows in the trace:

```
CLOBLOC(EAD18B9B) or
BLOBLOC(EAC17B90)
```

Character large objects (DB2 type CLOB) or binary large objects (DB2 type BLOB) can be found in the trace as:

```
(CLOB(43/43))'0000002B000....' or
(BLOB(43/43))'0000002B00....'
```

for example.

Numbers (integers and floats) are written out in readable format, while for packed decimals, a hexadecimal format is used. Time and timestamp values are written as EBCDIC strings.

Troubleshooting

The following section helps in classifying and solving problems which may arise when the SAP system is running on DB2 for z/OS.

Start-up and connection problems

Table 18 lists some common errors which may occur when an SAP work process connects to the ICLI server.

Table 18. ICLI server errors upon connection

Symptom	Cause	Resolution
ICLS1502E	DB2 subsystem not started	Make sure that the DB2 subsystem you plan to use is running. See also the message description in Appendix D, "ICLI messages and codes," on page 165.
ICLS1503E	DB2 subsystem not found	Make sure that the DB2 subsystem has a subsystem ID or a group attach name identical to <SAPSID> and that the DB2 subsystem exists on your system. See also the message description in Appendix D, "ICLI messages and codes," on page 165.
ICLS1900E	Either the message catalog that is used to retrieve a message text did not contain the message, or the specified message set was not found in the message catalog.	See the message description in Appendix D, "ICLI messages and codes," on page 165.
SQLCODE -553	DB2 authorization error: The ICLI server was running under a user ID which does not have appropriate DB2 authorization to change the current SQLID to <SCHEMA>.	Stop the ICLI server and restart it such that it runs under a user ID that has a secondary DB2 authorization ID of <SCHEMA>.
SQLCODE -818	Bind timestamp error: ICLI server was using a plan whose bind timestamp does not match the timestamp in the executable.	Make sure you specify the correct DBRMs in the FOME620B (FOME640B) job and re-run that job. Make sure there is no other ICLI server in the search path for APF modules.
SQLCODE -805	DBRM or package could not be found in plan error: ICLI server was using a plan which does not contain the specific package.	Make sure you specify the correct DBRMs in the FOME620B (FOME640B) job and re-run that job. Make sure there is no other ICLI server in the search path for APF modules.
SQLCODE -922	1. The specified plan does not exist or 2. the user does not have the appropriate access rights.	1. Create the plan or 2. correct the user's plan access rights.
SQLCODE -923	You have attempted to execute a non-existent plan. The reason code is used by the automatic BIND function to indicate that no record exists in SYSIBM.SYSPLAN for the plan ID you are attempting to run. As a consequence, plan allocation fails.	Create the plan.

Testing the database connection

Before you start up an SAP system, you might want to test the connection. A short test to determine whether a connection from the application server to the ICLI server and the database can be established is the following:

1. Log on as <sapsid>adm.
2. Test the network connections as described in chapter "Testing the database connection" in the *Connectivity Guide*.
3. On the application server, execute the command

   R3trans -x

If it completes with a return code of 0, the connection has been established.
Otherwise, look for error messages in the file `trans.log` on the application
server side and in the ICLI server log. If this does not help, you can set
ICLI_MSGLEVEL to I and *ICLI_TRACE_LEVEL* to 1 for the ICLI server and
rerun the **R3trans -x** command. Check the environment settings listed and look
for other error messages to identify the cause of the connection problem. If you
receive error messages, refer to the explanations provided in Appendix D, "ICLI
messages and codes," on page 165 and try the recommended actions.

> **Note:**
> If the connection problems cannot be resolved, try to collect diagnostic
> information (see "Diagnostic information to collect" on page 166).

Checking the status of ICLI threads

As of release 6.20, it is possible to generate two different types of status reports
that make it easier to analyze communication problems and identify hanging
threads:

- *Basic status reports* are generated for all the threads of one ICLI server. The
  generation of these reports has no impact on the ICLI server and its threads.
- An *extended status report* is generated for a specific ICLI server thread and
  contains detailed information on this thread.

  Note: When such a status report is generated, the ICLI server thread is stopped.

When you contact IBM support to help you determine and correct the cause of a
problem, you may be asked to provide us with thread status reports in addition to
the ICLI server message file and the ICLI client trace file normally requested.

The status reports are described in the following sections.

Generating basic status reports for all threads of an ICLI server

Basic status reports include the information at which time a thread last received
data. Consequently, this type of thread status report is especially useful when you
suspect that one or more threads are hanging. However, be aware that a thread
that has not received data for a longer period of time may not be hanging but just
idle, i.e., waiting for new data.

To generate basic status reports for all ICLI server threads, proceed as follows:

1. Determine the UNIX System Services process ID (PID) or the job name of the
   ICLI server (see "Identifying SAP work processes serviced by ICLI server
   threads" on page 110 for more information).
2. Issue the following command, e.g., on the z/OS console:

   `MODIFY <MYJOB>,appl=STATUS`

   where `<MYJOB>` is the job name of the ICLI server

When you issue this command, the status information for all ICLI server threads is
dumped into the ICLI server message file. You will receive one status report
(informational message ICLS1705I) per ICLI server thread. In the message, the time
appears in human-readable format, all other information is in hexadecimal
notation. The running system is not impacted, the ICLI server and all server
threads continue processing.

The following is a sample basic status report as you will receive it for all threads (in the sample, different typefaces are used to distinguish between the information units that are relevant for analysis; these are explained in Table 19):

```
(0002) ICLS1705I Diagnostic information: An ICLI server dump buffer is
       produced, processing continues.
       ICLI server health check:
18561450 17DADFD0 0000000A 17CE5938 17CC58D4 ...}...........M
18561460 00000051 00000051 00000000 00000555 ................
18561470 3C734832 00000010 17CC5640 00000000 ........... ....
18561480 E6858440 C6858240 F2F040F0 F77AF5F4 Wed Feb 20 07:54
18561490 7AF4F240 F2F0F0F2 1500           :42 2002..

       Time: 1014194367.941289 = (Wed) 2002 Feb 20 09:39:27.941289
```

Such a basic thread status report contains the following information:

Table 19. Thread-specific information in basic status reports

Thread-related information	Type face used in the sample status report
Thread ID	bold
The time at which the thread received the last data	underlined
The number of bytes the server *expects to receive*	underlined bold italics
The number of bytes the server *received in the last step*	italics
The *total number* of bytes (modulo 2**64) the server thread has *received* since it was created	bold italics

Generating an extended status report for a specific ICLI thread

> **Note:**
> When you generate an *extended status report* for a specific ICLI server thread, the thread will terminate. We recommend that you use this function only when IBM support requests such a report for problem analysis.

When IBM support asks you to generate an extended status report, proceed as follows:

1. Determine the thread ID of the ICLI server thread that is hanging (for example, by generating basic status reports for all ICLI server threads as described above).

2. Proceed as described in "Identifying SAP work processes serviced by ICLI server threads" on page 110 to obtain the necessary client information, i.e., the IP address of the SAP application server and the process ID of the SAP work process that corresponds to the ICLI server work thread.

3. Issue one of the following commands, e.g., on the z/OS console:

   ```
   MODIFY <MYJOB>,appl=THREADSTATUS <THREAD_ID>

   MODIFY <MYJOB>,appl=TS <THREAD_ID>
   ```

   where <MYJOB> is the job name of the ICLI server and <THREAD_ID> is the ID of the ICLI server thread for which the report is to be generated.

If you use an incorrect command, e.g., if you enter an invalid thread ID, the command will be ignored. If the environment variable ICLI_MSGLEVEL is set to

W, you can use the information in the message file to determine what went wrong. For example, ICLS1102W will be issued when you enter an invalid thread ID.

When you invoke the command using a valid thread ID, the following happens:

- An extended status report for the thread is written to the ICLI server message file (informational message ICLS1705I). In addition to the information provided in a basic status report, this message also includes the beginning of the last send and receive buffers.

- The ICLI server continues processing, but the thread terminates. If you are interested only in the health buffer on the server side and do not want to terminate the thread, you can use the following command:

  ```
  Modify <MYJOB>,appl=TS NOSTOP <THREAD_ID>
  ```

- On the client side, the corresponding dev_w* file will contain message ICLC2447I with the first 64 bytes of the transmit and receive buffers. This can be used for a first correlation. To capture both buffers in their entirety, see the following paragraph.

- If the ICLI client environment variable ICLI_TRACE was set to 1 or higher before the ICLI client (work process) was started, the termination of the server thread will induce the ICLI client to write the contents of its buffers into its trace file.

The buffer information in the ICLI server message file and the ICLI client trace file can then be used to correlate the server and client buffers for the thread, which can facilitate the analysis of a communication problem.

Note: If the environment variable ICLI_TRACE was not set before the ICLI client (work process) was started, you need to set it and try to reproduce the problem to obtain the client buffers in the client trace file.

The following is an example of an extended status report for one ICLI server thread as it will appear in the ICLI server message file:

```
(0002) ICLS1705I Diagnostic information: An ICLI server dump buffer is
       produced, processing continues.
       ICLI thread health check (this thread will be canceled):
185B3C68 17DC1630 00000013 17CE5938 17CC58D4 ...............M
185B3C78 0000006B 0000006B 00000000 00101D67 ...,...,........
185B3C88 3C735314 00000014 17CC5640 00000000 ........... ....
185B3C98 E6858440 C6858240 F2F040F0 F87AF4F1 Wed Feb 20 08:41
185B3CA8 7AF0F840 F2F0F0F2 15000000 00000000 :08 2002........
185B3CB8 00000067 00000014 00000000 38000000 ................
185B3CC8 04000000 00170000 00000000 00000000 ................
185B3CD8 00000000 00000000 00000300 0303C100 ..............A.
185B3CE8 0401F100 0401F100 04E25CFB 9B000000 ..1...1..S*.....
185B3CF8 00000010 00000000 00000000 00000004 ................
185B3D08 00000000 00000064 00000000 00000001 ................
185B3D18 00000000 01000000 00FA0061 61616161 .........../////
185B3D28 61616161 61616161 61616161 61616161 ////////////////
```

Runtime errors

Expected SQLCODEs

Some non-zero SQLCODEs are expected or even forced in the sense that the SAP application uses SQL, which leads to SQLCODEs that are again processed by the SAP system. An example for such an SQLCODE is -803 (duplicate row). It is possible that the application logic expects this in certain situations and reacts accordingly.

An SQLCODE of +466 reports that a stored procedure completed normally and specifies how many SQL query result sets were returned by the procedure.

An SQLCODE of +610 indicates an object was set into PENDING state after some CREATE or ALTER statements.

Some other SQLCODEs result from ICLI's exploitation of DB2's Dynamic Statement Cache. The SQLCODEs that fall into this category are -514, -516, and -518.

The default behavior of ICLI for the expected SQLCODES (-514, -516, -518, -803, 466, 610) is that no messages are written to the ICLI server message log.

See the description of the ICLI server environment variable *ICLI_IGNORE_SQLCODES* in Table 26 on page 159. The SQLCODES are still passed back to the ICLI client such that the DBIF of the SAP application can react accordingly.

Unexpected SQLCODEs

All other SQLCODEs are by default reported to the ICLI server message log file as ICLS1418E messages to help error determination. You can use the ICLI server environment variable *ICLI_IGNORE_SQLCODES* (see Table 26 on page 159) to prevent this reporting for certain DB2 SQLCODEs.

ICLI .<ID>.msg<n> message log file

The ICLI message log file `ICLI.<ID>.msg<n>` contains the messages which are reported by the ICLI server. For details, refer to "Message reporting capabilities" on page 116 and Appendix D, "ICLI messages and codes," on page 165.

Other errors

Other errors may arise from connection problems or resource problems.

Resource problems are most likely when the ICLI server creates a thread to handle a new connect. When the limit of thread numbers is reached (the maximum possible number of threads depends on your *REGION* size, see "Maximum number of threads" on page 47), no more threads can be created. However, the threads that are already up and running will remain alive.

In this situation, a second ICLI server should be started listening on a different connection port to handle any further ICLI client connect requests.

Other program failures (e.g., protection exceptions) are handled in the code by a signal handler which will write a CEEDUMP into the current working directory, which can be used to analyze the error.

DB2-related errors

The number of DB2 threads depends on the parameters *IDBACK* and *CTHREAD* . For a detailed description of the DB2 system parameters, refer to the *SAP installation guides*. Whenever no further DB2 connections are allowed, SAP seems to hang while the connection threads are initiated and waiting for an existing DB2 thread to end. These parameters have to be considered during the installation of DB2.

Output files

Table 20 on page 130 and Table 21 on page 131 list the output files used by the ICLI server, the ICLI client, and the DBIF.

Table 20. Output files written by the ICLI server

ICLI server message file	Comments
ICLI.<ID>.msg<n>	ICLI server message file located in the ICLI server log directory `<log_dir>`. `<id>` is either the value `<message_file_name>` specified using the command line option *-MFNAME* `<message_file_name>` or the UNIX System Services process ID of the ICLI server process (the default), and `<n>` is an ascending number starting with 1 (if you are using the command line option *-MFN* `<filenumber>` to create more than one ICLI message log file). The content of this file depends on the value of the environment variable *ICLI_MSGLEVEL*. For details, refer to Table 25 on page 157.
ICLItrace.<ID>.<UNIQUE_ID>	Trace output file located in ICLI server log directory `<log_dir>`. Identifier `<UNIQUE_ID>` is the decimal representation of the last four bytes of the UNIX System Services thread ID of the ICLI working thread. Output is generated only if the variable *ICLI_TRACE_LEVEL* is set. Refer to "ICLI server trace" on page 119.
ICLI.trc	Communication trace file located in the ICLI server log directory `<log_dir>`. Output depends on setting of the environment variable *ICLI_COM_TRACE*. Refer to "ICLI client and server communication trace" on page 121.
STDOUT	Used for special messages of the ICLI server.
STDERR	Used by z/OS Language Environment and for special messages of the ICLI server.

Table 21. Output files written by the ICLI client and the DBIF

Program	File	Comments
ICLI client	`STDOUT`	ICLI messages of command line tools (for example, R3trans). Refer to "ICLI client message reporting" on page 117.
	`dev_w<x>`	ICLI messages of work process. `<x>` is a current number for the work process. The file is located in /usr/sap/<SAPSID>/<Instance_Name>/work (for AIX, Solaris, and Linux) or \\<SAPLOCALHOST>\saploc\<SAPSID>\<Instance_Name>\work (for Windows).
	`ibmiclic.trc`	Communication trace file if environment variable *ICLI_COM_TRACE* is set; file is located in the current directory of the process using the ICLI client shared library. For the work process, this is the directory /usr/sap/<SAPSID>/<Instance_Name>/work (for AIX, Solaris, and Linux,) or \\<SAPLOCALHOST>\saploc\<SAPSID>\<Instance_Name>\work (for Windows).
	`icli_YYYYMMDDhh_pid.trc`	If the ICLI client environment variable *ICLI_TRACE* is not set or set to 0 (the default), the ICLI client creates these trace files only when unrecoverable errors occur that cannot be passed to the SAP DBIF layer. If *ICLI_TRACE* is set to 1 or greater, one trace file is created for each work process. The following naming conventions are used for the trace files:
		`icli_YYYYMMDDhh_pid.trc`
		where `YYYY` is the year, `MM` the month, `DD` the day, `hh` the hour of file generation, and `pid` the process ID of the work process.
		The files are created in the current working directory; for standalone tools, this is the directory from which the tool was invoked. SAP work process trace files can be found in the same directory as the `dev_w<x>` files. If errors occur early, the trace files may be written to the directory from which the SAP System was started.
		Depending on the setting of *ICLI_TRACE*, a high number of large trace files may be created. Consequently, the directories should be checked regularly and old trace files should be removed. Since creation date and hour are represented in the file name, old files can be identified easily.
		Contact IBM if the trace files are created although the environment variable *ICLI_TRACE* is not set or set to 0.
DBIF (client)	`STDERR`	DBIF error messages of command line tools (for example, R3trans)
	`STDOUT`	DBIF trace output of command line tools. Output is generated only if the environment variable *TRACE* is set.
	`dev_w<x>`	DBIF error messages and trace output of work process. `<x>` is a current number for the work process. The file is located in /usr/sap/<SAPSID>/<Instance_Name>/work (for AIX, Linux, and Solaris) or \\<SAPLOCALHOST>\saploc\<SAPSID>\<Instance_Name>\work (for Windows).

Chapter 8. Applying service to ICLI and the SAP application server on z/OS

> **Note**
>
> When using the 6.40 (downward compatible) kernel, ICLI component names contain the string '640' instead of '6.20'. The following discussion indicates the corresponding 6.40 names in parentheses. File content examples are shown for 6.20.

Some tasks that are performed automatically when the SAP system is installed must be performed manually in certain cases, for example:

- when a plan name other than the default is used
- when a new SAP system patch level for the application server on z/OS is installed
- when an IBM ICLI PTF is installed on z/OS
- when you change the user ID of an ICLI server
- when you install the first non-z/OS SAP application server after installing an SAP application on z/OS, because then the SAP installation tool will not have submitted the jobs for the ICLI server.

The tasks that have to be performed manually for the ICLI server and client components, and the SAP application server on z/OS, are described in this chapter:

- If you are using a plan name other than the default, you have to perform the tasks described in:
  - "DB2 authorization for the ICLI server and SAP application server on z/OS" on page 134
  - "DB2 binding for the ICLI server" on page 135
- If you have installed a new patch level, follow the instructions in "DB2 binding for the SAP application server on z/OS" on page 137.
- If you have installed a new ICLI PTF, the following sections apply:
  - "DB2 binding for the ICLI server" on page 135
  - "ICLI client download" on page 138
- If you have changed the user ID of an ICLI server, the following sections apply:
  - "DB2 authorization for the ICLI server and SAP application server on z/OS" on page 134
  - "DB2 binding for the ICLI server" on page 135
- When you have installed the first SAP application server on a non-z/OS host, you have to perform the tasks described in the following sections:
  - "DB2 authorization for the ICLI server and SAP application server on z/OS" on page 134
  - "DB2 binding for the ICLI server" on page 135

DB2 authorization for the ICLI server and SAP application server on z/OS

The SAP installation tool will grant the required DB2 authorizations for a certain plan name (see *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* and *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*).

You will only need to complete the procedures described in this chapter in certain cases. The following are the most common ones:

- You want to use other plan names for the ICLI server or the SAP application server.
- You have changed the user ID of the ICLI server.

For the **ICLI server**, you have to edit and submit a copy of the grant job template FOME620G (FOME640G), which is located in the directory <SYS1>..SAMPLIB. Do not change the original job in <SYS1>.SAMPLIB, because the SAP installation tool needs the original version!

The job template FOME620G (FOME640G) is used for the ICLI server. The corresponding SAP template DB2GRANT.tpl is only used for the SAP application server on z/OS. This SAP template and the currently submitted job DB2GRANT.jcl are located in the SAP system installation and upgrade directories.

As long as you use the same plan names, the job has to be submitted only once, that is, you do not have to resubmit the job if an ICLI PTF or SAP patch level is installed and the plan names remain unchanged.

The following is the JCL grant job template FOME620G (FOME640G):

```
//DBGRANT  EXEC PGM=IKJEFT01
//STEPLIB DD DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSIN    DD  *
    GRANT EXECUTE ON PLAN <PLANNAME> TO <ICLIRUN>;
    GRANT EXECUTE ON PACKAGE <PACKNAME>.* TO <SCHEMA>;
    GRANT EXECUTE ON PACKAGE DSNUTILS.* TO <SCHEMA>;
    GRANT EXECUTE ON PACKAGE DSNACC.* TO <SCHEMA>;
    GRANT EXECUTE ON PACKAGE DSNAOCLI.* TO <SCHEMA>;
    GRANT PACKADM ON COLLECTION <PACKNAME> TO <SCHEMA>;
    GRANT SYSCTRL TO <SCHEMA>;
    GRANT ALL ON TABLE SYSIBM.SYSPRINT TO <SCHEMA>;
    GRANT ALL ON TABLE DSNACC.TSTEMP_TBL TO <SCHEMA>;
    GRANT ALL ON TABLE DSNACC.IXTEMP_TBL TO <SCHEMA>;
    GRANT ALL ON TABLE DSNACC.DBSTATUS_TBL TO <SCHEMA>;
    GRANT ALL ON TABLE DSNACC.CMDMSG_TBL TO <SCHEMA>;
/*
//SYSTSIN  DD  *
 DSN SYSTEM(<SUBSYSTEM_NAME>)
   RUN  PROGRAM(DSNTIAD) PLAN(<DB2_DSNTIAD_PLAN>) -
        LIB('<DB2_RUNLIB_LOAD>')
 END
/*
```

To use this job template to grant the required privileges and authorizations for an ICLI server or SAP application server, complete the following tasks:

1. Make a copy of the default JCL job FOME620G (FOME640G). You will find the job in <SYS1>.SAMPLIB.

2. In the copy, replace the following placeholders with the appropriate values:

- `<DB2_DSNTIAD_PLAN>` with the name of the plan that was used to bind the DB2 utility DSNTIAD
- `<DB2_SDNSLOAD>` with the name of the DB2 load library
- `<DB2_RUNLIB_LOAD>` with the name of the library that contains the DB2 program DSNTIAD. DSNTIAD must be bound and the plan name must be adjusted (if the plan name is different from DSNTIA61)
- `<ICLIRUN>`
  - with the user ID <SAPSID>ICLI if you are using the grant job for the ICLI server
  - `<sapsid>adm` if you are using the grant job for the SAP application server
- `<PACKNAME>` with the correct package name
- `<PLANNAME>` with the ICLI server-specific or SAP application server-specific plan name. For details, see the *SAP installation guides*.
- `<SCHEMA>` with the appropriate schema name
- `<SUBSYSTEM_NAME>` with the appropriate DB2 subsystem name.

3. Use the user ID with DB2 SYSADM authority (see "User ID for selected SAP system installation tasks" on page 29) to submit the GRANT job. For subsequent grants with the same job, a user ID with DB2 SYSADM authority is not needed, because the minimum set of DB2 privileges needed has been included in the JCL job.

You do not need to resubmit the grant job if the ICLI server or the SAP application server is replaced and you choose the same plan name.

DB2 binding for ICLI and the SAP application server on z/OS

DB2 binding for the ICLI server

During the installation of the SAP system, the installation tool binds a certain plan (see *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*) for the ICLI server.

Manual binding is only necessary:
- when you use a plan name other than the above
- when a new ICLI PTF is installed on z/OS after the SAP system has been installed.

In these cases, you have to perform the binding manually for each DB2 subsystem the ICLI server connects to. In both cases, you can use existing bind jobs, which you either resubmit as is or tailor to your specific requirements before submitting them. Both options are described in the following sections.

Note: Although using the default plan name is quite sufficient in most cases, it can be useful to have different plan names for different ICLI servers that connect to the same DB2, because it improves the readability of DB2 performance data and DB2 threads and makes it easier to correlate them to a specific ICLI server.

DB2 binding by resubmitting a previously used bind job

If you have to rebind a plan for ICLI, for instance, after PTF installation, and you are using the same plan name, you can simply resubmit the currently submitted

job. If the previous binding was performed by the SAP installation tool, you will find the bind job in the installation subdirectory. To rebind it, transfer it to z/OS and resubmit it.

DB2 binding by changing a copy of the bind job template

If you are using a plan name other than the default, use a copy of the JCL bind job template FOME620B (FOME640B) and change it as needed.

The JCL job template is located in <SYS1>.SAMPLIB (shown here for 6.20):

```
//*BEGIN OF BIND PACKAGES
//DBPACK    EXEC PGM=IKJEFT01
//DBRMLIB  DD  DISP=SHR,DSN=<SYS1>.SAMPLIB
//STEPLIB  DD  DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(<SUBSYSTEM_NAME>)
    BIND PACKAGE(<PACKNAME>) OWNER(<SCHEMA>) MEMBER(FOME6201) -
         ACTION(REPLACE) ISOLATION(UR) DYNAMICRULES(RUN) -
         RELEASE(COMMIT) -
         CURRENTDATA(YES) KEEPDYNAMIC(YES)
    BIND PACKAGE(<PACKNAME>) OWNER(<SCHEMA>) MEMBER(FOME6202) -
         ACTION(REPLACE) ISOLATION(UR) DYNAMICRULES(RUN) -
         RELEASE(COMMIT) -
         CURRENTDATA(YES) KEEPDYNAMIC(YES)
  END
/*
//*END OF BIND PACKAGES
//*BEGIN OF BIND PLAN
//DBBIND    EXEC PGM=IKJEFT01
//STEPLIB  DD  DISP=SHR,DSN=<DB2_SDSNLOAD>
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(<SUBSYSTEM_NAME>)
    BIND PLAN(<PLANNAME>) OWNER(<SCHEMA>) -
         PKLIST(<PACKNAME>.*,DSNUTILS.*,DSNACC.*) -
         ACTION(REPLACE) RETAIN ISOLATION(UR) DYNAMICRULES(RUN) -
         ACQUIRE(USE) RELEASE(COMMIT) -
         CURRENTDATA(YES) KEEPDYNAMIC(YES)
  END
/*
//*END OF BIND PLAN
```

To use this job template for DB2 binding, proceed as follows:

1. Make a copy of the default JCL job FOME620B (FOME640B). You will find the job in <SYS1>.SAMPLIB.
2. In the copy, replace the following placeholders with the appropriate values:
   - <DB2_SDSNLOAD> with the name of the DB2 load library.
   - <PACKNAME> with a package name of your choice.
   - <PLANNAME> with the ICLI-specific plan name. For details, see *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*.
   - <SCHEMA> with the appropriate schema name.
   - <SUBSYSTEM_NAME> with the appropriate DB2 subsystem name.
   - <SYS1>.SAMPLIB with the database request module library that contains the ICLI DBRMs FOME6201 (FOME6401) and FOME6202 (FOME6402).
3. Submit your copy of the JCL job after the DB2 subsystem has been created.

DB2 binding for the SAP application server on z/OS

When installing a dialog instance on z/OS, the SAP installation tool binds the plan <PLANNAME> for the application server. When upgrading the SAP system, the SAP upgrade tool **R3up** binds the new plan for the application server.

Manual DB2 binding for the SAP application server is necessary, however, when the SAP DBSL shared library is replaced. This is always the case, for example, when a new SAP patch level is installed. Then, the corresponding bind job must be resubmitted.

These are the names of the SAP DBSL shared libraries on the different platforms:

AIX   dbdb2slib.o

Linux and Solaris
       dbdb2slib.so

Windows
       dbdb2slib.dll

After SAP system installation with the SAP installation tool, the JCL job template DB2BIND.tpl and the currently submitted job DB2BIND.jcl are located in the installation directory. The DBRM is located in the directory /usr/sap/<SAPSID>/exe/run. After an update with the SAP upgrade tool, the JCL job template and the currently submitted job are located in the upgrade directory.

The following is the JCL job template DB2BIND.tpl:

```
//*BEGINN OF DBRM COPY
//OCOPY   EXEC PGM=IKJEFT01,REGION=128M
//INHFS    DD PATH='@DBRMPATH@/DBRM.dbdb2099',
//         PATHOPTS=(ORDONLY)
//OUTMVS   DD  DSN=&&DBRMDATA(DBDB2099),UNIT=SYSDA,DISP=(NEW,PASS),
//         RECFM=FB,LRECL=80,BLKSIZE=3120,SPACE=(3120,(10,10,5))
//SYSTSPRT DD SYSOUT=*
//SYSTSIN  DD *
OCOPY INDD(INHFS) OUTDD(OUTMVS) BINARY PATHOPTS(USE)
/*
//*END OF DBRM COPY
//*BEGIN OF BIND PACKAGES
//DBPACK   EXEC PGM=IKJEFT01
//DBRMLIB  DD  DISP=SHR,DSN=&&DBRMDATA
//STEPLIB  DD  DISP=SHR,DSN=@LOADLIB@
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(@SUBSYSTEM@)
    BIND PACKAGE(@PACKNAME@) OWNER(@SCHEMA@) MEMBER(DBDB2099) -
         ACTION(REPLACE) CURRENTDATA(NO) -
         DYNAMICRULES(RUN) ISOLATION(UR) -
         CURRENTDATA(YES) RELEASE(COMMIT) KEEPDYNAMIC(YES)
  END
/*
//*END OF BIND PACKAGES
//*BEGIN OF BIND PLAN
//DBBIND   EXEC PGM=IKJEFT01
//STEPLIB  DD  DISP=SHR,DSN=@LOADLIB@
//SYSPRINT DD  SYSOUT=*
//SYSTSPRT DD  SYSOUT=*
//SYSTSIN  DD  *
  DSN SYSTEM(@SUBSYSTEM@)
    BIND PLAN(@PLANNAME@) OWNER(@SCHEMA@) -
         PKLIST(@PACKNAME@.*,DSNUTILS.*, -
               DSNACC.*,DSNAOCLI.*) -
         ACTION(REPLACE) RETAIN ISOLATION(UR) DYNAMICRULES(RUN) -
         ACQUIRE(USE) RELEASE(COMMIT) -
         CURRENTDATA(YES) KEEPDYNAMIC(YES)
  END
/*
//*END OF BIND PLAN
```

ICLI client download

During the installation of the SAP system, the ICLI client is transferred from its installation library on z/OS to the SAP execution directory on the application server. A manual transfer becomes necessary whenever an ICLI PTF is installed on z/OS after the installation of the SAP system.

The following steps describe how to manually transfer the ICLI client software to an application server.

> **Note**
>
> When using the 6.40 kernel, replace '620' with '640' in the following commands.

1. Change directory:
   - AIX:
     ```
     cd <sapmnt>/<SAPSID>/exe/run
     rm ibmiclic.o
     ```
   - Linux:

```
              cd <sapmnt>/<SAPSID>/exe/run
              rm ibmiclic.so
```
    - Windows:
```
              cd \\<SAPGLOBALHOST>\SAPMNT\<SAPSID>\SYS\exe\run
              del ibmiclic.dll
```
    - Solaris:
```
              cd <sapmnt>/<SAPSID>/exe/run
              rm ibmiclic.so
```

2. FTP connect to z/OS and issue the following two commands:
```
   cd 'SYS1.SFOMDATA'
   binary
```

3. Transfer the ICLI client software (for 6.40, replace '620' with '640'):
    - AIX:
```
         get FOME6205 ibmiclic.o
```
      Perform FTP disconnect and issue:
```
         chmod 755 ibmiclic.o
```
      To remove any unused references to the old library which may exist in AIX
      memory, issue the command:
```
         /usr/sbin/slibclean
```
    - Linux:
```
         get FOME620X ibmiclic.so
```
      Perform FTP disconnect and issue:
```
         chmod 755 ibmiclic.so
```
    - Windows:
```
         get FOME620N ibmiclic.dll
```
    - Solaris:
```
         get FOME620U ibmiclic.so
```
      Perform FTP disconnect and issue:
```
         chmod 755 ibmiclic.so
```

Multiple ICLI versions on one system

The term *multiple versions* in this context means different APAR levels of ICLI for
the same SAP kernel release. The ICLI modules for different SAP kernels do not
interfere with each other.

In general, it is not recommended to have multiple versions of ICLI installed on
one system. The preferred way of keeping different ICLI versions (such as for
pre-production tests) is to install them on a separate test LPAR and run those
versions in an SAP test system. If this is not feasible and you need to install
multiple versions of ICLI code for the same SAP kernel release, you should follow
these guidelines:

- Use SMP/E mechanisms to keep the different levels separate. Make sure that
  you install them to different target data sets and different paths in HFS.
- For a 6.20 ICLI, make sure the new SYS1.LINKLIB dataset containing FOME620S
  has the same characteristics (in particular, APF authorization) as your main
  SYS1.LINKLIB. For 6.40, see "ICLI client/server internals" on page 95).

In order to use the newly installed ICLI you need to:

- do the DB2 bind for ICLI as described earlier in this chapter

- if necessary, download the ICLI client as described above
- modify the NLSPATH environment variable such that it points to the correct ICLI message catalog fome620m.cat (fome640m.cat) in HFS.
- modify your ICLI start procedure to point to the new modules (insert the new SYS1.LINKLIB in your STEPLIB for 6.20 ICLI)

Chapter 9. Using z/OS printing facilities to print SAP documents

This chapter describes how to use z/OS printing facilities to print SAP documents.

General remarks

There are several printer configurations which you can use to print SAP documents. They are illustrated in Figure 20.
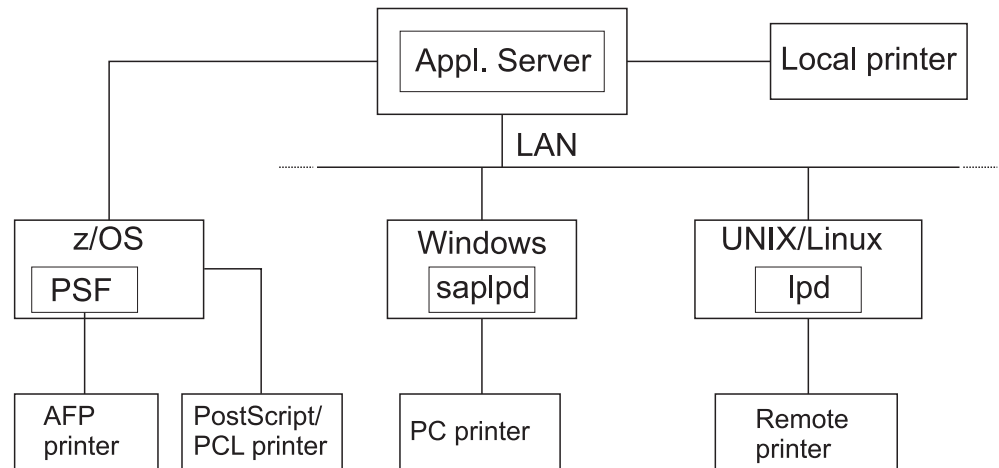
Figure 20. Printer configuration

You can set up printers in the following ways:

* Create a local printer attached to the system on which the SAP system is running.
* Use a UNIX or Linux system as a remote print server. For this purpose, the standard **lpd** daemon (line printer daemon) or other print manager software can be used.
* Use a PC as a remote print server. Use the **saplpd** utility, because Windows does not support the standard **lpd** daemon.
* Use Advanced Function Presentation (AFP) printers on the z/OS system. The AFP server function can be on any z/OS system in the sysplex environment.

In the SAP system, each printer is assigned to an application server instance running one or more spool work processes. It is therefore possible to assign host printers to an application server instance running on z/OS while network printers are handled by an application server instance on Windows. The first three configurations are no different for SAP on DB2 than they are for any other SAP platform. The AFP configuration is of special interest to z/OS customers and is therefore discussed here in greater detail.

Even if no application server instance is configured on z/OS, the z/OS Infoprint Server can be used for host printing, including AFP printing. In this case, the SAP system must be configured to recognize z/OS as a remote printer server.

Infoprint Server for z/OS

The following printing functions let you print SAP documents on local and remote z/OS printers, including IBM Advanced Function Presentation (AFP) printers:

Infoprint Server
> *Infoprint Server*, a priced feature of z/OS, lets SAP system users print on local and remote z/OS printers from an SAP application server. If the target printer is an IBM AFP printer, Infoprint Server can automatically invoke data transforms provided by Infoprint Server Transforms to convert SAP data to the AFP format.

Infoprint Server for z/OS Transforms
> *Infoprint Server for z/OS Transforms*, a product available at no charge to Infoprint Server customers, provides a set of data stream transforms that run on z/OS. The SAP to AFP transform converts SAP OTF and ABAP documents to the AFP format.

> **Note:**
> These z/OS printing functions mean that you no longer need to convert SAP data to the AFP format on an AIX system, using either PSF for AIX or Infoprint Manager for AIX, and upload the data to z/OS for printing.

Configurations

Two configurations are possible. The SAP application server can run on either

1. the *same* z/OS system as Infoprint Server. In this case, print requests are handled by the Infoprint Server SAP Output Management System (OMS). The OMS also lets users cancel jobs and obtain job status. It also contains a Callback daemon that lets users receive immediate notification when their documents have completed (successfully or unsuccessfully). Figure 21 on page 143 shows this configuration.

2. a *different* system (UNIX/Linux or Windows). In this case, print requests are handled by the Infoprint Server LPD. Users cannot receive notification about their print requests or cancel jobs. Users can receive job status; however, the job status is not as accurate as when the SAP application server runs on the z/OS system and the Infoprint Server OMS is used. Figure 22 on page 144 shows this configuration.

Both configurations let users print to any local or remote printer defined in the Infoprint Server Printer Inventory. In both configurations, if the printer is an IBM AFP printer, the data can be transformed from the SAP format to the AFP format on the z/OS system before printing.

Software requirements

- z/OS:
- The SAP application server for z/OS or, in the case of a different system, an SAP application server.
- To print on an IBM AFP printer:
  - Infoprint Server Transforms (5697-F51) Release 1.0 or later
  - PSF for OS/390 (5655-B17) Version 3 Release 1.0, or later or a comparable product.

Overview

Figure 21 and Figure 22 on page 144 show how SAP application server, Infoprint Server, Infoprint Server Transforms, and PSF for OS/390 work together to let you print any SAP documents on z/OS printers. These figures also show the possible printer configurations you can set up in your installation.

Figure 21: When your SAP application server runs on a z/OS system, the Infoprint Server OMS receives print and status requests and the Infoprint Server SAP Callback daemon provides notification of job events.

Figure 22 on page 144: When your SAP application server runs on a non-z/OS system, the Infoprint Server LPD — and not the Infoprint Server OMS — receives print and status requests, and the notification of job events does not occur.
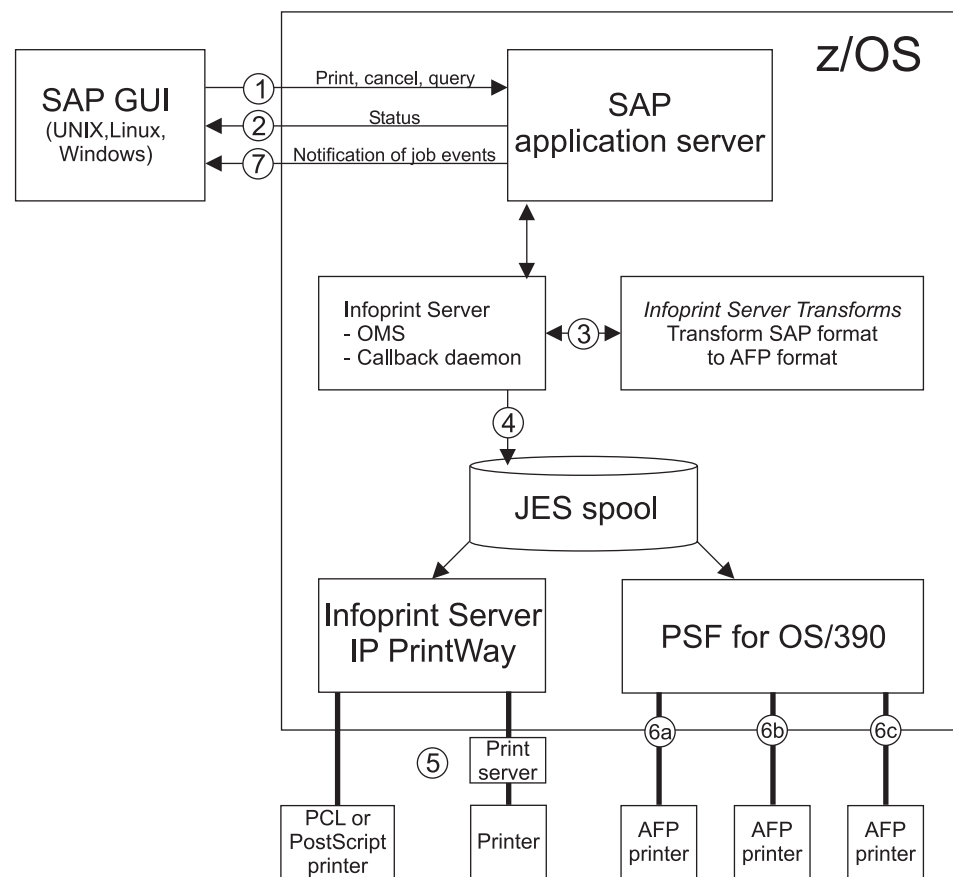
Figure 21. Printing SAP documents with the SAP application server on z/OS

UNIX, Linux, Windows

SAP GUI
(UNIX, Linux, Windows)

① Print, cancel, query →

② Status

SAP application server

TCP/IP network connection

Infoprint Server - LPD

③

Infoprint Server Transforms
Transform SAP format to AFP format

④

JES spool

z/OS

Infoprint Server IP PrintWay

PSF for OS/390

⑤

Print server

6a   6b   6c

PCL or PostScript printer

Printer

AFP printer

AFP printer

AFP printer
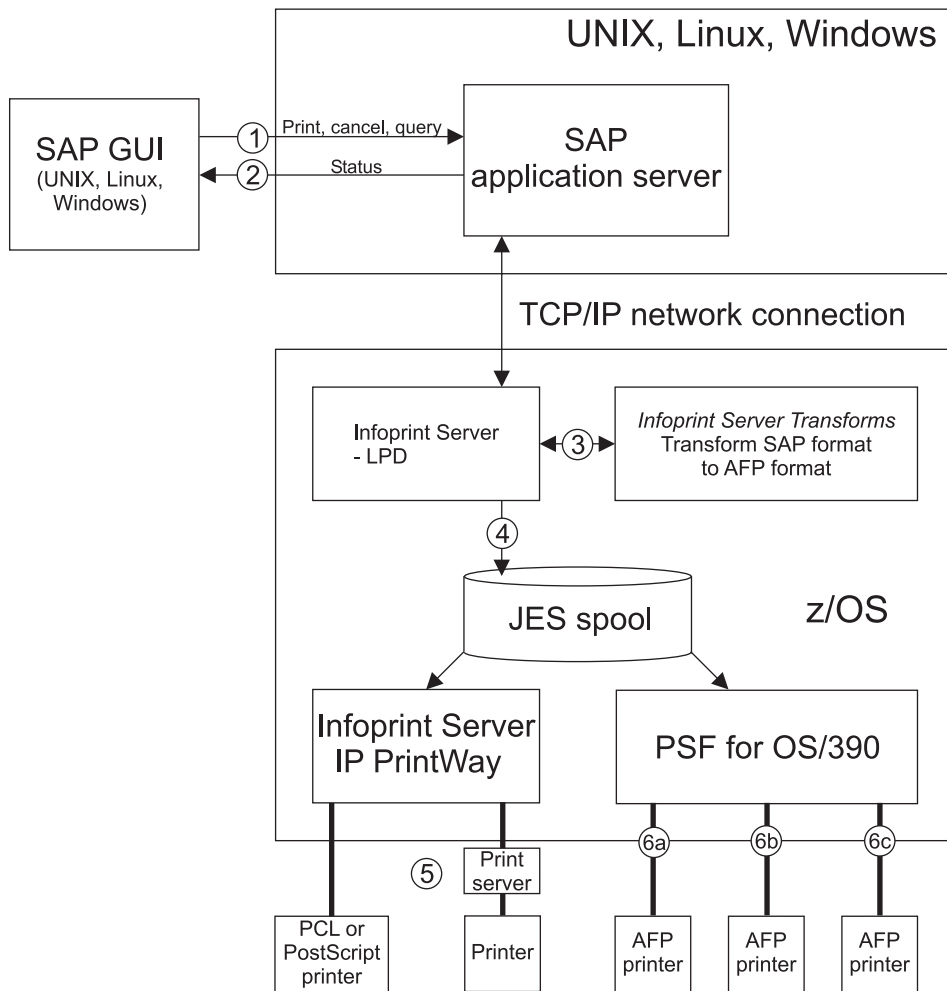
Figure 22. Printing SAP documents with the SAP application server on non-z/OS systems

The processing steps, including possible printer configurations, are basically the same in both figures:

1. From an SAP GUI, users make print, cancel, job query, and device query requests to the SAP application server on z/OS. These print requests specify the name or a printer definition that the administrator has created in the Infoprint Server Printer Inventory.

2. For a status request, Infoprint Server returns the status of a print job or a list of print jobs that the printer is processing.

3. For a print request, Infoprint Server first detects the data format of the input document. For SAP OTF or ABAP data to be printed on an AFP printer, *Infoprint Server Transforms* transforms the data to the AFP format. The SAP to AFP transform uses transform options that the administrator has specified for the printer. For PCL or PostScript data, the data is not modified.

4. Infoprint Server creates a print job on the z/OS JES spool. From the JES spool, the IP PrintWay component of Infoprint Server, PSF for OS/390, or JES can print the document.

5. IP PrintWay transmits documents selected from the JES spool to remote printers or print servers in your TCP/IP network. The remote printer or print server must

   • contain an LPD,

- contain an Internet Printing Protocol (IPP) server, or
- support TCP/IP direct sockets printing.

6. PSF for OS/390 prints documents selected from the JES spool to IBM AFP printers. The printers can be

   a. local,

   b. TCP/IP-attached, or

   c. SNA-attached.

7. As print jobs complete (successfully or unsuccessfully), the Infoprint Server SAP Callback daemon sends notifications back to the SAP system users.

Installation and customization requirements

The following steps summarize the installation and customization requirements for the administrator. For more detailed instructions, see "Where to find more information" on page 146.

1. Install Infoprint Server for z/OS. To print to AFP printers, also install Infoprint Server Transforms and PSF for OS/390.

2. Proceed as follows:

   a. If the SAP application server is on a z/OS system, customize the Infoprint Server SAP Callback daemon, and customize and start Infoprint Server.

   b. If the SAP application server is on a non-z/OS system, customize and start the Infoprint Server LPD.

3. Create a printer definition for each target printer in the Infoprint Server Printer Inventory.

   - For AFP printers, configure the printer definition so that Infoprint Server automatically performs the SAP to AFP transform for SAP OTF and SAP ABAP documents. Also, specify transform options such as the resolution of the printer.

   - For non-AFP network printers, you can select the standard IP PrintWay formatting option, which is the default option.

4. Customize the SAP system to define a Real Output Management System (ROMS) and a Logical Output Management System (LOMS) that interface with Infoprint Server. Infoprint Server provides a sample file that you can import into SAP to define one ROMS and two LOMS. If the SAP application server is on a non-z/OS system, you do not need to perform this step.

5. If the SAP application server is on the z/OS system:

   a. Select **E** (External output management system) as the host spool access method.

   b. If the printer is an AFP printer, select **SAPGOF** as the device type. For other printers, select a type supported by the printer.

   c. Select an Infoprint Server LOMS to handle print requests.

   d. Name the Infoprint Server printer definition as the host printer.

   If the SAP application server is on a non-z/OS system:

   a. Select **U** (Print using Berkeley protocol) as the host spool access method.

   b. If the printer is an AFP printer, select **SAPGOF** as the device type. For other printers, select a type supported by the printer.

   c. Name the Infoprint Server printer definition as the host printer.

   d. Specify the address of the z/OS system as the destination host.

e. For the port number, specify the port where the Infoprint Server LPD will listen.

Where to find more information

For further information on the Infoprint product family, visit the home page of the IBM printing division at `http://www.printers.ibm.com`.

Infoprint Server for z/OS

For detailed information about the customization and administration of the Infoprint Server and the SAP system, refer to these publications:

- *z/OS Infoprint Server Customization* for customizing Infoprint Server and Infoprint Server Transforms.
- *z/OS Infoprint Server User's Guide* for customizing SAP and defining printers to SAP.

Infoprint Manager for AIX

The AIX Infoprint Manager is the related product for AIX that supports the SAP OMS interface. For more information on the Infoprint Manager setup for SAP, see the documentation *SAP R/3 Support: Configuration and Printing Tasks*, which is available at ftp://ftp.software.ibm.com/printers/specs/sapr332.pdf.

Chapter 10. Planning the transition to DB2 Connect

The SAP database connectivity through ICLI will be replaced by the IBM product DB2 Connect. This chapter describes some planning issues that you should consider prior to the transition from ICLI to DB2 Connect.

SAP Note 728743 contains general information about if and when DB2 V8 will be supported for SAP components. As far as SAP releases up to SAP 4.6D are concerned, such support will be available using only ICLI connectivity, if at all.

On the other hand, DB2 Connect is the only supported database connectivity for SAP in the following scenarios:

1. When you plan to run any new SAP systems based on the upcoming SAP Web Applications Server 6.40 within the SAP NetWeaver '04 framework.
2. When you are running an SAP system based on the SAP Web Application Server 6.20 and you want to migrate your database to DB2 for z/OS V8.

SAP Web Application Server 6.40 / SAP NetWeaver '04

SAP NetWeaver '04 requires z/OS 1.4 and DB2 for z/OS V8 running in new-function mode. The only supported database connectivity option will be via the IBM DB2 Connect product. There will be no ICLI that supports DB2 for z/OS V8 running in new-function mode. The specific hardware and software prerequisites to run SAP NetWeaver '04 are listed in SAP Note 660528. For more information on SAP Web Application Server 6.40, refer to the SAP documentation for this release.

As of 6.40, this *Planning Guide* will be available as a new SAP document entitled *Planning Guide: z/OS Configuration for SAP on IBM DB2 Universal Database for z/OS*.

SAP Web Application Server 6.20 on DB2 for z/OS V8

Migrating an SAP 6.20 database from DB2 V7 to DB2 V8 will be possible only using the downward compatible SAP 6.40 kernel. This kernel will be dual-mode in the sense that it uses ICLI to connect to a DB2 V7 database and DB2 Connect to connect to a DB2 V8 database. To allow for a smoother migration path, the ICLI running with the 6.40 (downward compatible) kernel can be run against DB2 V8 while it is still running in compatibility mode. As soon as DB2 is switched to enabling-new-function mode, SAP needs to connect to the database via DB2 Connect. Thus, the process of migrating to DB2 V8 with your existing V7 database will involve the installation of the DB2 Connect product on all SAP application servers and the adjustment of SAP parameters. The exact required level of the DB2 Connect product is expected to be documented in SAP Note 81737. For the customization of the DB2 installation and the necessary profile changes in SAP, see the description in SAP Note 731937. All potential scenarios involving DB2 V8 with SAP are depicted in the following figure:

Introducing DB2 V8 and DB2 Connect at SAP
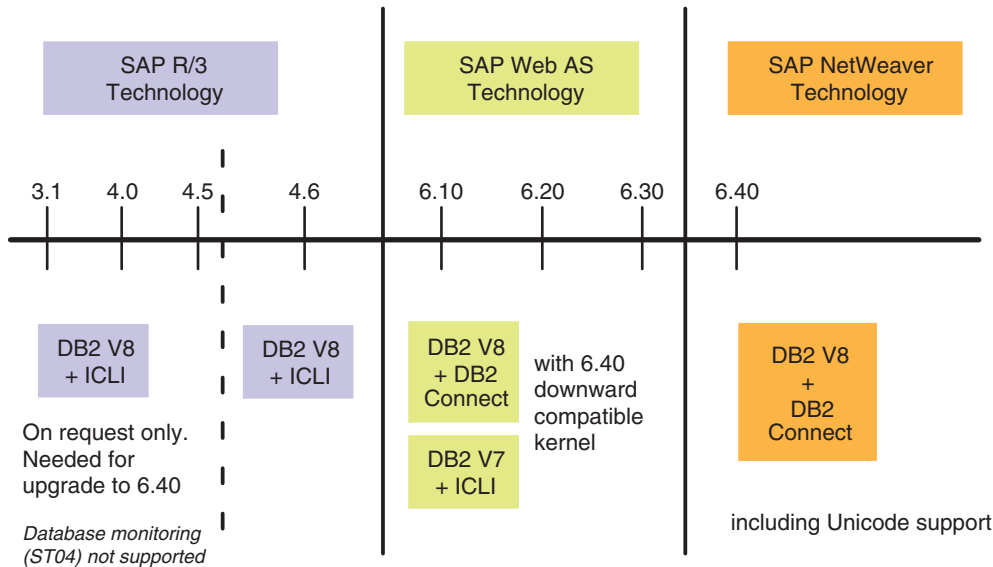
Figure 23. Introduction of DB2 V8 and DB2 Connect at SAP

For further and more current information about possible upgrade paths involving DB2 V8, see the SAP document *SAP on IBM DB2 UDB for OS/390 and z/OS: Best Practice for Installing or Migrating to DB2 V8*, available at

`http://service.sap.com/solutionmanagerbp`

WLM exploitation

Currently, WLM exploitation is done on the z/OS application server side and by ICLI. With DB2 Connect, DBSL uses the sqleseti() call to supply new WLM qualifiers to DDF. For more information, refer to the SAP *Planning Guide* for release 6.40.

Even with the newly available qualifiers and the new mechanism for specifying them, velocity goals are still recommended. WLM response time goals cannot be used to influence the response times of SAP transactions.

WLM transition from DB2 V7 to DB2 V8

For non-z/OS application servers, there is no smooth transition. The old WLM definitions for the SAP subsystem in WLM apply only to ICLI. Customers need to establish new classification rules for the DDF subsystem using the new qualifiers. DBSL supplies the necessary qualifiers to DDF via the sqleseti() call at a commit/rollback boundary.

> **Caution**
>
> Customers need to specify classification rules with an asterisk, especially for the CI qualifier, to avoid having, for example, the E1E001 instance classified incorrectly because 'E1E' was specified in the rule and not 'E1E*'.

Part 4. Appendixes

151

Appendix A. Checklists guiding through this document

The following checklists will guide you through those chapters of this document that provide you with important information on how you should set up your system to get it up and running. The emphasis is on the tasks described in Part 2, "Preparation," on page 21, which must be performed before the SAP System can be installed.

It is recommended that you make a copy of the checklist for each SAP system you are going to set up, and that you fill it out while working through the document.

Planning: Required reading

Before you start on the actual configuration of the system to prepare it for SAP system installation, you must have completed the planning phase. The documents listed in the following table will help you with successfully planning your system.

Table 22. Planning: Required reading

Done	Topic/Document
	Overview of the configuration structure
	See Chapter 1, "Configuration structure," on page 3 in this document.
	Network- and connectivity-related information
	See the *Connectivity Guide* and *High Availability for SAP on zSeries Using Autonomic Computing Technologies*.
	Hardware and software requirements
	Refer to the current installation documentation.
	High availability and data sharing
	See the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.
	Planning for backup and recovery
	See the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.
	See also the whitepaper *SAP R/3 Storage Management for OS/390: an Examination of IBM's RAMAC Virtual Array Turbo 8-Path Storage Control*, which is available to customers and partners via SAPNet, and the *SAP Database Administration Guide*.

Preparation: Mandatory tasks

The following table lists the tasks you have to perform on z/OS before the SAP system can be installed. If not stated otherwise, all the tasks listed are described in detail in Part 2, "Preparation," on page 21 of this document. The checklist ends with the installation of the SAP system.

Table 23. Preparation: Mandatory tasks

Done	Task
	Ensure that the latest PTFs are installed. See SAP Note 81737.
	Choose the SAP system ID. See the *SAP installation guides*.
	Create the z/OS user IDs. See "z/OS group IDs and user IDs" on page 24.

	Configure z/OS.
	
	Once you have completed the following tasks, an IPL of the system will most probably be required.
	
	The configuration tasks are described in Chapter 3, "Configuring z/OS":
	• **Set up RRS and SMS for DB2.**
	  See "Setting up Resource Recovery Services (RRS)" on page 34 and "Setting up SMS" on page 33.
	• **Configure the initial WLM setup.**
	  See "WLM considerations for DB2 subsystems" on page 35, "Workload Management (WLM)" on page 36, and "Workload Management (WLM) — Classification of SAP workload" on page 82.
	• **Select the UNIX System Services parameters for the ICLI server and the SAP application server on z/OS.**
	  See "Selecting the UNIX System Services parameters" on page 41.
	• **Set up FTP.**
	  Set up the FTP server on z/OS (see "User ID for selected SAP system installation tasks" on page 29). For more information on setting up an FTP server, refer to *z/OS Communications Server: IP Configuration Guide*.
	• **For the SAP application server on z/OS, complete the following tasks:**
	  – **Set up TCP/IP and define the host names.**
	    See "TCP/IP setup" on page 47 and "The hostname command for application servers on z/OS" on page 49.
	  – **Set up the file system.**
	    See "File system setup" on page 49.
	  – **Create links for tcsh and the Korn shell.**
	    See "The C-shell, tcsh, and Korn shell" on page 63.
	  – **Check whether the correct version of Java is installed.**
	    See "Java on z/OS" on page 63.
	• **If necessary, IPL the system to activate the configuration changes.**

	Set up DB2. Refer to the *SAP installation guides*.
	Set up ICLI server instances. See Chapter 4, "Setting up instances of the ICLI server," on page 65 and "Configuring ICLI PassTicket signon" on page 29.
	Run the SAP installation tool. See the *SAP installation guides*.

Appendix B. Where to find further information on important tasks

The IBM and SAP documentation listed in Table 24 contains further information on important tasks and concepts discussed in this document.

Table 24. Important tasks

Activity	Documentation
Binding an ICLI server	Chapter 8, "Applying service to ICLI and the SAP application server on z/OS," on page 133
Changing an ICLI connection port	"The ICLI connection port" on page 99
Database layout (Structure, Changes)	*SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*, section "Database Layout"
DB2	In the *SAP installation guides* and the IBM documentation *DB2 for OS/390 and z/OS Installation Guide*, section "DB2 Setup"
Environment variables	Appendix C, "ICLI environment variables," on page 157
High availability	See *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206 (available from the IBM SAP on DB2 Web site), and the *SAP Database Administration Guide*.
Installing an ICLI client	• *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*, sections "Installing the Central Instance" and "Installing Additional Dialog Instance" • *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*, section "Installing an SAP Instance"
Operation (backup, recovery, reorganization)	• *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*, sections "Basic Operations" and "Database Management" • *DB2 for OS/390 and z/OS Administration Guide*, section "Operation and Recovery" • *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206
z/OS setup	Chapter 3, "Configuring z/OS," on page 33
Performance monitoring	• *DB2 for OS/390 and z/OS Administration Guide*, section "Performance Monitoring and Tuning" • *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server*, section "Database Management"
Permissions (user IDs), authorizations, RACF definitions	Chapter 2, "z/OS security, group IDs and user IDs," on page 23 and the *SAP installation guides*.
Printing	• Chapter 9, "Using z/OS printing facilities to print SAP documents," on page 141 • *AFP Printing for SAP Using R/3 and R/2*
PTFs required	SAP provides an easy-to-use tool to check the PTF status. Refer to SAP Note 183311 for details. See also the *SAP Database Administration Guide* section entitled "Automated PTF Check".

Table 24. Important tasks (continued)

Activity	Documentation
SAP (kernel, SAP data)	• *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* • *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*
SAP system	• *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS* • *SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS*, section "Completing and Checking the Installation"
Security	See *Permissions*.
Setting up the network between ICLI client and server	*SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition* contains SAP-specific information for an optimized network setup. See also *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.
Space management	• *SAP Database Administration Guide*, section "Database Management"
Starting/stopping DB2	*DB2 for OS/390 and z/OS Administration Guide*
Starting/stopping an ICLI server	See Chapter 7, "ICLI client and server," on page 95
Starting/stopping the complete system	*SAP Database Administration Guide*: "Start and Restart Operations"
TCP/IP support	See "TCP/IP setup" on page 47
Tuning	• *DB2 for OS/390 and z/OS Administration Guide*, section "Performance Monitoring and Tuning" • *SAP Database Administration Guide*, section "Database Management" • *z/OS UNIX System Services Planning*, section "Tuning UNIX System Services Performance" • *Porting Applications to the OpenEdition MVS Platform*
User IDs	Chapter 2, "z/OS security, group IDs and user IDs," on page 23
Utilities	• *SAP Database Administration Guide*, section "Database Management" • *DB2 for OS/390 and z/OS Utility Guide and Reference*
WLM:	
• WLM setup for the ICLI server and the SAP application server	• "Workload Management (WLM)" on page 36 • "Workload Management (WLM) — Classification of SAP workload" on page 82
• Configuring a WLM service definition	• "Workload Management (WLM) — Classification of SAP workload" on page 82 • *z/OS MVS Planning: Workload Management*
• Switching on/off WLM management for the SAP application server	• *SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS*, section "SAP Profile Parameters and Environment Variables"

Appendix C. ICLI environment variables

This appendix lists the environment variables supported by the ICLI server and client. For details about the timeout behavior influenced by environment variables, see the *Connectivity Guide*.

ICLI client environment variables

The environment variables listed in Table 25 should be changed by modifying the files `.dbenv_<hostname>.sh` and `.dbenv_<hostname>.csh` in the home directory of the `<sapsid>adm` user ID and restarting SAP. Since there are multiple places that define environment variables, you should not try to determine their values by looking at the various files but instead by looking at the actual environment when being logged on as user ID `<sapsid>adm`.

The SAP profile parameters should be changed by modifying the SAP instance profile `<SAPSID>_<Instance_Name>_<hostname>` located in the `/usr/sap/<SAPSID>/SYS/profile` directory on AIX, Linux, Solaris, and in the `\\SAPGLOBALHOST\sapmnt\<SAPSID>\SYS\profile` directory on Windows and afterwards restarting SAP.

Table 25. Environment variables supported by the ICLI client

Variable	Possible Values[1]	Comments
ICLI_NET_STATS_INTERVAL	0, nnn	Defines the number of ICLI request/response pairs after which network performance statistics information is externalized and reset. The variable must be set before the application server is started. Interdependence of *ICLI_NET_STATS_SECONDS* and *ICLI_NET_STATS_INTERVAL*: • If both are not set or set to 0 before the application server is started, network performance statistics are not generated. • If both are set before the application server is started, *ICLI_NET_STATS_SECONDS* takes precedence over *ICLI_NET_STATS_INTERVAL*. For additional information on network performance statistics, see "Generating and analyzing network performance statistics" on page 78.

Table 25. Environment variables supported by the ICLI client (continued)

Variable	Possible Values[1]	Comments
ICLI_NET_STATS_SECONDS	0, nnn	Defines the number of seconds after which the ICLI client externalizes and resets network performance statistics. The information is not written for idle work processes until the next network request occurs. Use this variable if you want to gather information about the network throughput of each SAP work process. The variable must be set before the application server is started. Interdependence of *ICLI_NET_STATS_SECONDS* and *ICLI_NET_STATS_INTERVAL*: • If both are not set or set to 0 before the application server is started, network performance statistics are not generated. • If both are set before the application server is started, *ICLI_NET_STATS_SECONDS* takes precedence over *ICLI_NET_STATS_INTERVAL*. For additional information on network performance statistics, see "Generating and analyzing network performance statistics" on page 78.
ICLI_SIGNON_FILE	File name	Specifies the name of the signon file. The name must contain the full path and the file name. This value is needed if you want to use more than one ICLI server per instance. The file specified by *ICLI_SIGNON_FILE* overrides an existing default file, see "Configuring PassTicket signon for ICLI clients" on page 30.
ICLI_TRUSTED_CONNECTIONS	0, 1	This variable allows disabling the PassTicket signon function. If not set, the ICLI client assumes that the ICLI server requires PassTicket signon and behaves accordingly. If set (to 1), the ICLI client assumes that the ICLI server trusts the ICLI client by not requiring PassTicket signon and behaves accordingly. The variable is set for the ICLI client and the ICLI server; the settings for both must be identical. See also ICLI server variable *ICLI_TRUSTED_CONNECTIONS*.
ICLI_TRACE	0, 1, 2, 3, 4, 5	If not set or set to 0, the ICLI client creates trace files only when unrecoverable errors occur that cannot be passed to the SAP DBIF layer. If set to 1 or greater, one trace file is created for each work process. The following naming conventions are used for the trace files: `icli_YYYYMMDDhh_pid.trc` where `YYYY` is the year, `MM` the month, `DD` the day, `hh` the hour of file generation, and `pid` the process ID of the work process. Since setting the variable to 1 or greater has a dramatic impact on performance and huge trace files may be written, do so only upon instruction by IBM or SAP.
ICLI_COM_TRACE	0, 1	If set to 1, the ICLI client generates a communication trace. This trace is written into the file `ibmiclic.trc` (in the current directory of the process using the ICLI client shared library). If not set, the communication trace is not generated.
[1]		Default values are <u>underlined</u>. They are used when the variable is not defined.

ICLI server environment variables

Depending on the start method you are using, you change the values of the environment variables for an ICLI server by modifying either the UNIX System services shell script, the batch job, or the environment file `iclienv` for the started task and restarting the ICLI server.

> **Note**
>
> If you want to define several blank-separated values for an environment variable, for example, for *ICLI_IGNORE_SQLCODES*, you must enclose the values in double quotes (″ ″) if and only if you are setting the environment variable via a UNIX System Services shell script. In the examples below, this is assumed. In other words, if you are using a started task or batch job to start the ICLI server, you *must not* enclose blank-separated values in double quotes.
>
> Example of an entry in a UNIX Services Shell script:
>
> ```
> export ICLI_IGNORE_SQLCODES="-603 -514 -516 -518 -803 466"
> ```
>
> Example of an entry in a JCL job or `iclienv` file of a started task:
>
> ```
> ICLI_IGNORE_SQLCODES=-603 -514 -516 -518 -803 466
> ```

The values of the following three variables can also be changed by using SAP transaction ST04, which has the advantage that you do not have to restart the ICLI server afterwards:

- *ICLI_TRACE_LEVEL*
- *ICLI_MSGLEVEL*
- *ICLI_COM_TRACE*

Table 26. ICLI server-specific environment variables

Variable	Possible Values[1]	Comments
ICLI_ARM_ELEMNAME	String containing the ARM element name	Element name used by ICLI to register with Automatic Restart Management (ARM). See "ICLI server registration with Automatic Restart Management" on page 115 for naming rules.
ICLI_BIND_IPADDR	Character string representing a TCP/IP host address expressed in standard dotted decimal notation. (for example: ″10.0.0.1″)	When this environment variable is set to a valid z/OS host IP address, the main communication port of the ICLI server will be bound to this IP address. In this case, the ICLI server will only receive connection requests on its ICLI connection port that are made by ICLI clients using this IP address. The IP address is reported in message ICLS1011. If the variable is not set, the ICLI server will listen on its communication port for connection requests by ICLI clients received over *any* IP address. In this case, message ICLS1011 will contain the string INADDR_ANY instead of an actual IP address. Note that with IP address ″0.0.0.0″, the server behaves as if the variable were not set but message ICLS1011I will contain ″0.0.0.0″ instead of the string INADDR_ANY. For more information, see "Associating the ICLI server with a specific IP address" on page 101.

Table 26. ICLI server-specific environment variables (continued)

Variable	Possible Values[1]	Comments
ICLI_COM_TRACE	0, 1	If set (to 1), this variable causes a communication trace to be generated. The trace is written into the file ICLI.trc in the log directory (see *-LOGDIR* parameter on page 103) of the ICLI server. If not set, a communication trace is not generated.
ICLI_MESSAGE_SYNC	Any	If this environment variable is not set (default), the last messages are buffered and are not written to the ICLI message log immediately. As a result, these last messages are lost in the event of an ICLI server abend. If it is set to any value, it ensures that each message has been written to the error message file. This behavior needs considerably more time; it is therefore recommended that you check to determine if you really need it.
ICLI_IGNORE_SQLCODES	″-514 -516 -518 -803 466 610″, a list of DB2 SQLCODES separated by blanks, or the value ALL	Use this variable to restrict the DB2 SQLCODES which will be reported in the ICLI server message log as messages. As a default (i.e. when you do not explicitly set the environment variable to any value), the DB2 SQLCODES which can result from the use of the Dynamic Statement Cache ″-514 -516 -518″, SQLCODE ″-803″ (which indicates duplicate records in a DB2 INSERT operation), SQLCODE ″466″ (which documents that a stored procedure completed normally and specifies how many SQL query result sets were returned by the procedure), and ″610″ (which indicates an object in PENDING state after some CREATE or ALTER) are not reported. You can explicitly set this variable to a list of SQLCODES which you do not want to be reported in the ICLI server log, thereby avoiding cluttering up this file. Another possible value for the variable is ALL, which indicates that no DB2 SQLCODE will be reported. The variable does not influence the reporting of the SQLCODES to the ICLI client and the passing of these SQLCODES back to SAP applications. See also the note on page 159.
ICLI_MSGLEVEL	W, I, E	Message level on the ICLI server. This value controls the minimum severity of messages written to the message file ICLI.<ID>.msg<n> in the log directory of the ICLI server. Supported values are: **W** — The default. Display warning, error, and severe error messages. **I** — Display all messages: informational, warning, error, and severe error messages. **E** — Display error and severe error messages.

Table 26. ICLI server-specific environment variables (continued)

Variable	Possible Values[1]	Comments
ICLI_PARAMETERS	A list of ICLI server parameters starting with a minus sign ('-') and separated by blanks	This variable is used to specify additional ICLI server parameters usually provided as command line arguments. The reason for using this environment variable is the limit of 100 characters when specifying parameters in a started task or batch job. Parameters specified by this environment variable are overridden by parameters specified on the command line. Parameters specified by this environment variable have the same syntax and use the minus sign ('-') as parameters specified on the command line. Example of a UNIX System Services shell script entry: `ICLI_PARAMETERS="-CHECK MYDB -PLAN FOMEP620"` For more information on ICLI server command line parameters, see "ICLI server command line options" on page 101. See also the note on page 159.
ICLI_SAP_SCHEMA	String containing a DB2 schema	The value of the variable is only used as input for the ICLI server command line option *-CHECK*. Only set this variable to your DB2 schema name if you are not using the default DB2 schema SAPR3 and you want to use *-CHECK*. The check function will then verify whether the ICLI server can set its SQLID to the specified schema. If the variable is not set, *-CHECK* will use the default DB2 schema SAPR3 to perform its check. (See option *-CHECK* in "ICLI server command line options" on page 101).
ICLI_SET_STMT_ID	0, 1	If set to 1, this variable causes all PREPARE statements to be done without a SET_ID call (DB2 Version 7 only). In this case you lose some problem determination possibilities within SAP (such as statement identification of deadlock victims). Therefore, set this variable only if instructed by SAP or IBM service personnel.
ICLI_STOP_MODE	1, 2	This environment variable is used to specify the mode, i.e., to influence the behavior of the ICLI server, after the specified ICLI_STOP_TIME has elapsed: • 1 (which is the default) means that the remaining threads are stopped • 2 means that the remaining threads are allowed to continue running, and you will find an information message in your message file.
ICLI_STOP_TIME	n>0, (300)	This variable is used to specify the time period (in seconds) for a smooth stop. If this variable is either not set or set to an invalid value, a default of 300 seconds is taken.
ICLI_TCP_KEEPALIVE	0 - 2147460	The value 0 disables TCP Keep Alive for ICLI server sockets. Values of 1 to 2147460 set an ICLI server socket-specific TCP Keep Alive value. See the *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition* for a detailed description.
ICLI_TRACE_LEVEL	0, 1, 2, 3, 4, 5, 6	See "ICLI server trace" on page 119.

Table 26. ICLI server-specific environment variables (continued)

Variable	Possible Values[1]	Comments
ICLI_TRUSTED_CONNECTIONS	0, 1	This variable permits the PassTicket signon function to be disabled. If set (to 1), the ICLI server trusts the ICLI client and does not require the ICLI client to use PassTicket signon. If not set, the ICLI server requires the ICLI client to use PassTicket signon. The variable is set for both the ICLI client and the ICLI server; and the settings for both must be identical. See also ICLI client variable *ICLI_TRUSTED_CONNECTIONS*.
ICLI_WRITE_TO_SYSLOG	0, 1	This variable allows messages to be written not only to the message file but to the z/OS System Log, too. If this environment variable is set to 1, all ICLI messages are written to the z/OS System Log according to the setting of *ICLI_MSGLEVEL*. If not set or if set to 0 (which is the default), no messages will be passed to the System Log.
[1] Default values are underlined. They are used when the variable is not defined.		

The environment variables presented in the following table are not ICLI server-specific, but may nevertheless need to be set for the ICLI server.

Table 27. Other z/OS environment variables which are relevant for the ICLI server

Variable	Possible Values	Comments
STEPLIB	DB2 loadlib	Some DB2 setups require that the DB2 load library <DB2_SDSNLOAD> be put into *STEPLIB*. If your particular DB2 setup does require this and you use a UNIX shell script to start the ICLI server, set *STEPLIB* as you would any other environment variable. When you use a JCL batch job or started task to start the ICLI server, use a *STEPLIB DD* statement in your job to specify the *STEPLIB*. In this case, if you additionally define a *STEPLIB* as environment variable in the environment file of the started task, the *STEPLIB* from the environment file will override the *STEPLIB* setting in your started task.

Table 27. Other z/OS environment variables which are relevant for the ICLI server (continued)

Variable	Possible Values	Comments
TZ		The environment variable defines the time zone in which the ICLI server runs and thus determines the timestamp used for ICLI server messages. To make it easier to map ICLI server messages to ICLI client messages, client and server should run in the same time zone. If the variable is not set when the server is started, GMT is used. If you start the server with the started task or JCL batch job template created by the ICLI server setup tool, the variable is set to the correct time zone automatically (see "Running the ICLI server setup program" on page 65 for details). If you define your own start procedure, you should set the value of this environment variable to your local time zone if: • you start the ICLI server with the started task or the JCL batch job • and GMT is not your local time. You do not have to set the environment variable if the ICLI server is started with a UNIX shell script, because then the correct value is taken from the `profile` file. For information on how to set *TZ*, see *z/OS UNIX System Services Command Reference*. Some additional information on time settings is also provided in section "Choosing consistent time settings" on page 64.

Appendix D. ICLI messages and codes

This appendix describes the messages issued by the ICLI client and server, as well as information on what diagnostic information to save when problems are reported to SAP/IBM. For detailed information on the ICLI client and server output files, refer to "Output files" on page 129.

Format of ICLI messages

ICLI messages have the following format:

`(<UNIQUE_ID>) <MESSAGE_PREFIX> <MESSAGE_NUMBER><SEVERITY_ID>`

(<UNIQUE_ID>)
>   The four numbers (in parentheses) represent the *<UNIQUE_ID>* of the thread which wrote the message. The *<UNIQUE_ID>* is a decimal representation of the last four bytes of the UNIX System Services thread ID.

<MESSAGE_PREFIX>
>   The textual message prefix can be *ICLC*, which represents ICLI client messages, *ICLS*, which represents ICLI server messages, or *ICLI*, which represents instance setup program messages.

<MESSAGE_NUMBER>
>   A four-digit decimal number.

<SEVERITY_ID>
>   One of the following letters:
>   **I**   Informational message
>   **W**   Warning
>   **E**   Error
>   **S**   Severe error

The last line of a message is a timestamp that indicates when the message was written.

Some ICLI messages contain values for 'errno' and 'errno2'. 'errno' is reported in both decimal and hexadecimal notation, 'errno2' is reported in hexadecimal notation only. The values represent Return Codes and Reason Codes of z/OS UNIX System Services. For an explanation of the values, see *z/OS UNIX System Services Messages and Codes* and *z/OS MVS Programming: Workload Management Services*. The explanations can give you valuable information as to why a specific ICLI error occurred.

Here is an example of an ICLI server message:

```
(0003) ICLS1029E Error creating thread, R/C = -1
       errno = 112 (0070), errno2 = 0B51025D
       Message: EDC5112I Resource temporarily unavailable..
       Time: 980168837.969214 = (Mon) 2001 Jan 22 14:07:17.969214
```

Diagnostic information to collect

Whenever you receive an error which you should report to SAP/IBM, you should try to save the diagnostic information listed in the following sections. The more complete the information, the better the problem can be analyzed. Some of the information may not always be available (for example, trace files if tracing was switched off).

For a complete problem analysis, it is also important that both client and server information are collected. Some errors on the client might appear to be "local errors" but are in fact caused by prior errors on the server side and vice versa.

Basic diagnostic information to collect

To collect diagnostic information perform the following steps:

1. Save all the output files mentioned in Table 20 on page 130 and Table 21 on page 131. Saving the files is important because some files are overwritten when SAP is restarted (for example the dev_.. files on the application server).

2. Find any CEEDUMPs that have been produced and save them. CEEDUMPs are usually written to the current working directory of the user that started the ICLI server. This is not necessarily the directory you specified in the command line parameter LOGDIR. Look for CEEDUMPs in the following places:

   - The directory found in _CEE_DMPTARG, if found.
   - The current working directory, if this is not the root directory (/), and the directory is writeable.
   - The directory found in the environment variable TMPDIR (if the temporary directory is not /TMP).
   - The directory /TMP

3. Save any z/OS system log/console messages that were written around the time the error occurred.

Diagnostic information to collect in case of communication problems

Perform the following steps:

1. Set the environment variable *ICLI_MSGLEVEL* to I on the ICLI server side.
2. Issue the following UNIX System Services commands on the ICLI server side:

   ```
   netstat -A -E <ICLI server job name or started task name>
   ```

   Additionally, many helpful options are available for the **netstat** command. In order to get a list and short description of these options, issue

   ```
   netstat -?
   ```

3. Issue the following UNIX System Services command on the ICLI client side:

   ```
   netstat -s > prev_net_s
   ```

4. Set the environment variable *ICLI_MSGLEVEL* to "I" on the ICLI client side.
5. Recreate the problem.
6. Issue the following command on the ICLI client side:

   ```
   netstat -s > post_net_s
   ```

   Compare the `prev_net_s` and the `post_net_s` file for unexpected changes.

7. Issue the following UNIX System Services command on the ICLI server side:

   ```
   netstat -A -E <ICLI server job name or started task name>
   ```

Additionally, many helpful options are available for the **netstat** command. In order to get a list and short description of these options, issue

```
netstat -?
```

Compare the `prev` and `post` files and look for unexpected changes.

8. Save all files as explained under "Basic diagnostic information to collect" on page 166 and the generated `post` and `prev` files.

List of ICLI messages

The format in which the ICLI messages are listed below contains certain placeholders for variables that are substituted at runtime. The list of such placeholders is:

Table 28. Placeholders in ICLI messages

Placeholder	Substitutes for
nnn	decimal value
c	single character
sss	character string
xxx	hexadecimal value
ppp	hexadecimal address value
<TCPIP>	high-level qualifier under which the TCP/IP product was installed
<ICLIRUN>	user ID under which the ICLI server runs
ssid	database attach name (i.e. DB2 subsystem name or DB2 group attachment name)

ICLI client messages

> **Notes:**
>
> 1. The message text of some ICLI client messages lists diagnostic data (for example 'Datagram', 'Local Socket', 'Remote Socket', 'Buffer Length') which is not explained in the respective explanation section. This data serves only as input for your IBM Service Center.
>
> 2. Some ICLC messages are transformed into SQLCODEs and appear as such in the dev_w* files. For example, all ICLC25xx messages appear as SQLCODE -925xx in the dev_w* files. Furthermore, some artificial SQLCODEs (for example: -95099) are generated on the server side.

ICLC2112W **Environment variable ICLI_COM_TRACE has an unknown value '*sss*'.**
Default value 'unset' will be used.
Processing continues,
but no communications trace will be generated.

Explanation: The value that was specified for the ICLI_COM_TRACE environment variable is invalid.

System Action: Processing continues with the default value, that is, no communications trace will be generated.

User Response: Check your environment settings and specify a valid value for this variable. For details, see Appendix C, "ICLI environment variables," on page 157.

ICLC2120W **ICLI passticket verification is disabled.**
Processing continues, trusted connections are assumed, that is, no security information is passed to the ICLI server.

Explanation: The environment variable ICLI_TRUSTED_CONNECTIONS was set to 1 to disable passticket security checking. This will only be successful if you set the ICLI_TRUSTED_CONNECTIONS variable to 1 on the ICLI server as well.

System Action: Processing continues. No security information is passed to the ICLI server.

User Response: None.

ICLC2121W **The environment variable ICLI_TRUSTED_CONNECTIONS was set to the invalid value '*sss*'.**
Passticket Security checking will be performed.

Explanation: The environment variable ICLI_TRUSTED_CONNECTIONS was set to an invalid

value. Passticket security checking will be performed.

System Action: Processing continues with security checking enabled.

User Response: Check your environment settings and specify valid values for this variable. For details see Appendix C, "ICLI environment variables," on page 157.

ICLC2122E **System error in line *nnn* and file *sss*.**
Insufficient memory!
The ICLI client process immediately returns an error to the calling SAP process.
Details: Request for *nnn* bytes failed.

Explanation: The ICLI client could not obtain enough memory. The requested memory size is given in "Details".

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Make more memory available. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLC2125I **Network Statistics summary output:**
ssssss

Explanation: This informational message is written if the ICLI client environment variable ICLI_NET_STATS_INTERVAL is set to a value greater than 0. It contains summary output generated by the network statistics counter.

System Action: Processing continues.

User Response: None.

ICLC2126I **Network Statistics about sent packets (requests): *sss***

Explanation: This informational message is written if the ICLI client environment variable ICLI_NET_STATS_INTERVAL is set to a value greater

than 0. It contains a statistics record about the size of requests.

System Action: Processing continues.

User Response: None.

ICLC2127I **Network Statistics about received packets (responses):** *sss*

Explanation: This informational message is written if the ICLI client environment variable ICLI_NET_STATS_INTERVAL is set to a value greater than 0. It contains a statistics record about the size of responses.

System Action: Processing continues.

User Response: None.

ICLC2128I **Network Statistics about the time consumption of request/response pairs:** *sss*

Explanation: This informational message is written if the ICLI client environment variable ICLI_NET_STATS_INTERVAL is set to a value greater than 0. It contains a statistics record about the time consumed by request/response pairs.

System Action: Processing continues.

User Response: None.

ICLC2129E **No service entry was found for service name '*sss*', protocol TCP.**
The connection port could not be determined.
The ICLI client process immediately returns an error to the calling SAP process.
Please check your entries in the services file.

Explanation: The service name <sss> with protocol TCP was not found in the `services` file, but this entry is needed to determine the connection port.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: See the SAP Installation documentation for details on how to add the expected service name to the `services` file.

ICLC2184E **System error in line *nnn* and file *sss*.**
Processing continues.
Diagnostic information:
Data transmission for the performance monitor function of SAP transaction ST04 failed
with return code = *nnn*.
The cause of the error should be

described in a previous error message.

Explanation: Within the performance monitor function of SAP transaction ST04, some data could not be sent or retrieved. There should be a preceding message giving more details on this error.

System Action: Processing continues.

User Response: Repeat your steps in transaction ST04. If the error occurs again, save all available diagnostic information and report the error to SAP/IBM.

ICLC2251W **System error: The attempt to open the communications trace file ibmiclic.trc failed.**
Processing continues, but no communications trace will be generated.
Diagnostic information:
open() RC = *nnn*, errno = *nnn* (0x*xxx*)
Message: *sss*

Explanation: The communications trace file `ibmiclic.trc` could not be opened in the current working directory. Check system message <sss> for the reason.

System Action: The ICLI client continues processing with communications tracing disabled.

User Response: See the system message <sss> for the reason. Resolve the problem by, e.g., increasing the file system size and checking the permissions.

ICLC2281W **System error: The attempt to send data to the ICLI server failed.**
The ICLI client will retry to send the data after 2 seconds.
Diagnostic information:
Temporary TCP buffer space shortage (errno=ENOBUFS).

Explanation: The ICLI client is trying to send a packet to the ICLI server. The send() call returned with an error because there is not enough remaining TCP send buffer space.

System Action: The ICLI client waits for 2 seconds before it retries to send the data.

User Response: Check if your send buffer space is defined large enough (system wide TCP parameter) and if the MTU sizes of client and server are configured correctly.

ICLC2282W **System error: Data from the ICLI server could not be received.**
The ICLI client will retry to receive the data after 2 seconds.
Diagnostic information:
Temporary TCP buffer space shortage (errno=ENOBUFS).

Explanation: The ICLI client is trying to receive a packet from the ICLI server. The recv() call returned with an error, because not enough TCP receive buffer space was available.

System Action: The ICLI client waits for 2 seconds before it retries to receive the data.

User Response: Check if your receive buffer space is defined large enough (system wide TCP parameter) and if the MTU sizes of client and server are configured correctly.

ICLC2300W **System restriction: The installed hardware does not support high resolution performance counters. Network Performance Statistics service is not available.**

Explanation: This error message is issued when the installed hardware does not support high resolution performance counters. This counter is needed for time measurements in microseconds.

System Action: Processing continues, but no Network Performance Statistics are generated.

User Response: If you want to utilize Network Performance Statistics, install hardware that supports high resolution performance counters.

ICLC2446I **Diagnostic information: Sending of data failed in the transmission phase in line** *nnn* **and file** *sss***. Number of bytes sent:** *nnn* **/ number of bytes to be sent:** *nnn* **Refer to the preceding messages for further information.**

Explanation: The ICLI client could not send all the data to the ICLI server. This message is preceded by other messages which provide more information about the cause of the problem. If the value of bytes sent is negative, no data was delivered at all. The reason may be a broken line. If the number of bytes sent is lower than the number bytes to be sent, only part of the data was delivered. The reason may be heavy network traffic (the media is at its limit).

System Action: The ICLI client returns an error to the calling SAP process.

User Response: See the explanations of the preceding messages.

ICLC2447I **Diagnostic information:** *sss* **of data failed in line** *nnn* **and file** *sss* **with rc =** *nnn***.** *sss***.**

Explanation: The ICLI client could receive or send data from or to the ICLI server. For additional information dump the first 64 Bytes of the current transmit and receive buffer. You can uses this buffer

together with the health buffer on server side to analyze the problem.

System Action: The ICLI client will continue. The next message will follow.

User Response:

ICLC2448I **Diagnostic information: Receiving of data failed in stream transmission phase. Refer to the preceding messages for further information.**

Explanation: The ICLI client could not receive all the data from the ICLI server. This message is preceded by other messages which give more information about the cause of the problem.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: See the explanation sections of the preceding messages for further information.

ICLC2449I **Diagnostic information: Receiving of data failed normal in transmission phase. Refer to the preceding messages for further information.**

Explanation: The ICLI client could not receive all the data from the ICLI server. This message is preceded by other messages which give more information about the reason.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: See the explanation sections of the preceding messages for further information.

ICLC2454I **Diagnostic information: The attempt to connect to the ICLI server failed, because the version information could not be sent. The ICLI client returns an error to the calling SAP process. Refer to the preceding messages for further information.**

Explanation: The connect to the ICLI server failed.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: See the explanation sections of the preceding messages for further information.

ICLC2500E **Error during the initialization phase of the ICLI interface:**
An invalid database host '*sss*' was specified, the connection could not be established.
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information: line *nnn* and file *sss*

Explanation: The name of the host where the ICLI server resides was not specified by the SAP process, or the name specified cannot be resolved by TCP/IP.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: The database host must be specified in all of the following three places: (1) environment variable 'SAPDBHOST', (2) parameter 'SAPDBHOST' in the SAP profile, (3) parameter <SAPSID>/dbhost in file TPPARAM. For details, see "Changing the database server of an application server running on UNIX, Linux, or Windows" on page 115. Check the settings and test with nslookup if TCP can resolve the specified name. If the name resolution is successful but this message still appears, save all diagnostic information and report the error to SAP/IBM. Note: These database host settings are normally done during the SAP installation.

ICLC2503E **The attempt to connect to the ICLI server was rejected:**
The authorization check during the connection phase failed,
because the timestamp request could not be transmitted to the ICLI server.
The ICLI client process immediately returns an error to the calling SAP process.
Refer to the preceding messages for further information.

Explanation: The transmission of the timestamp request to the ICLI server failed. This timestamp is needed for PassTicket authentication for Secured Signon. This message is preceded by other messages which give the reason for the unsuccessful transfer.

System Action: The ICLI client returns an error to the calling SAP process because the attempt to connect to the ICLI server was refused.

User Response: Refer to the explanation sections of the preceding messages to determine the cause of the problem.

ICLC2504E **ICLI server error: The attempt to connect to the ICLI server was rejected:**
The authorization check during the connection phase failed
because the timestamp generation on the ICLI server failed .
The ICLI client process immediately returns an error to the calling SAP process.
See the messages in the ICLI server message file on the database host for further information.

Explanation: The ICLI server could not generate a timestamp value that is needed for PassTicket authentication for Secured Signon. The ICLI server will reject any further requests from the calling SAP process.

System Action: The ICLI client returns an error to the calling SAP process because the attempt to connect to the ICLI server was rejected.

User Response: See the messages in the ICLI server message file on the database host for further information.

ICLC2505E **The attempt to connect to the ICLI server was rejected:**
The ICLI version check during the connection phase failed,
because ICLI client and server versions are not compatible.
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI client version is incompatible with the ICLI server version. Normally, such an incompatibility should be reported by message ICLC2555E, listing the different version identifiers.

System Action: The ICLI client returns an error to the calling SAP process because the attempt to connect to the ICLI server was rejected.

User Response: Look for messages in the ICLI server message file on the database host. Check your installation and upgrade the part with the lower version number. If this does not resolve your problem, save all diagnostic information and report the error to SAP/IBM.

ICLC2508E **System Error in line *nnn* and file *sss*.**
The ICLI client could not send data to the ICLI server.
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:
send() RC=*nnn*, errno=*nnn*, message: *sss*

Explanation: The ICLI client tried to send a packet to

the ICLI server. The send() call returned with an error. The message <sss> gives detailed information about the problem.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem indicated by the errno and the corresponding message. Also take a look at the ICLI server message file to determine the cause of the problem, e.g., the corresponding server thread might have abended. If the error was not caused by connection problems, save all available diagnostic information and report the error to SAP/IBM.

ICLC2509E **System error: The attempt to send data to the ICLI server failed because more than** *nnn* **signal interruptions occurred.**
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The attempt to send data to the ICLI server was interrupted too often (<nnn> times) by a signal.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to determine the cause of the interrupts and correct it. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLC2510E **System error in line** *nnn* **and file** *sss*.
The ICLI client could not receive data because
more than *nnn* **signal interruptions occurred.**
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI client was interrupted too often (<nnn> times) by a signal while waiting for data from the ICLI server.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to determine the cause of the interrupts and correct it. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLC2513E **System error: The attempt to create a temporary socket failed.**
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:

socket() RC=*nnn*, **errno=***nnn*, **message:** *sss*

Explanation: During connect processing, the ICLI client could not create a required temporary socket. The string <sss> provides detail information about the problem.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the TCP/IP problem indicated by the errno and its corresponding message. If you cannot resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

ICLC2516E **Authorization check failed because the ICLI Signon File '***sss***' could not be accessed.**
System message: *sss*
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI Signon File <sss> could not be accessed. The reason is described in the system message <sss>.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem indicated by system message <sss>. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLC2517E **Error in line** *nnn* **and file** *sss*:
Authorization check failed because of a syntax error
in the ICLI Signon File in line *nnn*:
sss
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI Signon File has a syntax error in line <nnn>. The string <sss> specifies the error.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Correct the cause of the error. For details see "Configuring ICLI PassTicket signon" on page 29.

ICLC2530E *sss*

Explanation: The ICLI server thread to which this ICLI client was connected encountered an abend. This abend may be caused either by a problem in the ICLI server thread itself or by an abend in a component the ICLI server is using. A common reason for such abends are DB2 abends (abend code 04E), which indicate either

some kind of DB2 problem or, e.g., the cancelling of the DB2 thread. This problem will also occur when an SAP work process that is active in DB2 is restarted. The error message reported by the ICLI server should give an indication of where the abend happened. In any case, the ICLI server thread is terminated and no further database requests may be passed to this thread. Depending on the kind of abend, it may be possible to perform a successful reconnect, e.g., when a DB2 was cancelled manually on the database host. In most other cases, the reconnect will only be successful if the cause of the abend is corrected on the database host.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: The database host operators need to resolve the cause of the error. Check the ICLI server message file for more information.

| ICLC2535E | Internal error in line *nnn* and file *sss*: ICLI client thread cannot continue. Details: Control Block buffer pointer is NULL. The ICLI client process returns an error to the calling SAP process. |

Explanation: This is an internal ICLI error.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Sometimes preceding messages may help to determine the cause of the problem. If not, save all available diagnostic information and report the error to SAP/IBM.

| ICLC2540E | ICLI server error during the initialization phase of the ICLI interface: A connection to the ICLI server could not be established because the ICLI server did not provide a valid port number for the new work thread. The ICLI client process immediately returns an error to the calling SAP process. Check the ICLI server message file for error messages. |

Explanation: During the initialization of a new work thread, the ICLI client must receive the port number it is to use for further communication with the server work thread. This error indicates that the server could not provide such a port number.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Check the ICLI server message file for error messages. Correct the server problem. If you are using the Common INET stack, the problem might be that the specified port range is exhausted.

| ICLC2541E | Interface error during the initialization phase of the ICLI client: A connection to the ICLI server could not be established because the ICLI client could not determine a valid ICLI connection port number. The ICLI client process immediately returns an error to the calling SAP process. |

Explanation: The connect to the ICLI server failed because the port to use was not specified correctly.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check if the connection port for the TCP protocol is specified in the corresponding line of the services file. Check if the environment variable R3_PORT is set to a non-supported value. If the error was not caused by any of these conditions, save all available diagnostic information and report the error to SAP/IBM.

| ICLC2542E | System error during the initialization phase of the ICLI interface: The attempt to connect to the new ICLI server work thread on host '*sss*' using port *nnn* failed because the connection attempt was rejected by the server. The ICLI client process immediately returns an error to the calling SAP process. |

Explanation: The connect to the ICLI server work thread failed within the TCP/IP layer.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are two situations which might lead to this message:
1. The processing of the connection request failed on the ICLI server.
Action: Check the ICLI server message file and correct the error situation on server.
2. The network went down.
Action: Bring up your network and check it using ping and/or traceroute.

If the error was not caused by any of these situations, save all available diagnostic information and report the error to SAP/IBM.

ICLC2543E **System error during the initialization phase of the ICLI interface:**
The attempt to connect to the new ICLI server work thread on host '*sss*' using port *nnn* failed because the connection timed out. The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The connect to the ICLI server failed.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are three situations which might lead to this message:
1. The processing of the connection request failed on the ICLI server.
Action: Check the ICLI server message file and correct the error situation on server.
2. The network went down.
Action: Bring up your network and check it using ping and/or traceroute.
3. There is no route defined from the client to the server host. Action: Check your network setup and test for a possible connection using ping and/or traceroute. The TCP/IP commands ping and traceroute should work in both directions.
If the error was not caused by any of the situations listed above, save all available diagnostic information and report the error to SAP/IBM.

ICLC2544E **System error during the initialization phase of the ICLI interface:**
The attempt to connect to the new ICLI server work thread on host '*sss*' using port *nnn* failed. The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:
connect() RC=*nnn*, errno=*nnn*, message: *sss*

Explanation: The connect to the ICLI server failed. The string <sss> gives detail information about the problem.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are four situations which might lead to this message:
1. The processing of the connection request failed on

the ICLI server.
Action: Check the ICLI server message file and correct the error situation on server.
2. The network went down.
Action: Bring up your network and check it using ping and/or traceroute.
3. There is no route defined from the client to the server host.
Action: Check your network setup and test for a possible connection using ping and/or traceroute. The TCP/IP commands ping and traceroute should work in both directions.
4. Other reason as described in <sss>.
Action: Try to resolve this problem using TCP/IP documentation.
If the problem cannot be resolved, save all available diagnostic information and report the error to SAP/IBM.

ICLC2551E **System error in line *nnn* and file *sss*. The ICLI client could not receive data from the ICLI server. The ICLI client process immediately returns an error to the calling SAP process. Diagnostic information: recv() RC=*nnn*, errno=*nnn*, message: *sss***

Explanation: The ICLI client waits for a packet from the ICLI server which is the answer to a previous request. The recv() call returned with an error. The cause of this error can be either connection problems or an internal error. The string <sss> gives detailed information about the problem.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: If the error was not caused by connection problems, save all available diagnostic information and report the error to SAP/IBM.

ICLC2555E **The attempt to connect to the ICLI server failed due to a version mismatch:**
ICLI server/client SAP version: *nnn* (*sss*) / *nnn* (*sss*)
ICLI server/client protocol version: *nnn* / *nnn* and internal version: *nnn* / *nnn*
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The connect to the ICLI server fails because this client is incompatible with the contacted ICLI server.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check if you have specified the correct database host and the port of the ICLI server

you want to use. Check if the correct ICLI client was downloaded to the application server execution directory.

ICLC2556E System error: A connection to the ICLI server could not be established because the connection attempt was rejected.
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:
Using Host '*sss*', IP address '*sss*', ICLI connection port *nnn*,
and communication protocol TCP/IP.

Explanation: The connect to the ICLI server failed because no server can be reached on the given host and port.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are several situations which might lead to this message:
1. No ICLI server was running and listening on the ICLI connection port used by the client.
Action: Start the ICLI server on the required port.
2. The ICLI client used the wrong ICLI connection port.
Action: Correct the connection port used by the ICLI client.
3. The processing of the connection request failed on the ICLI server.
Action: Check the ICLI server message file and correct the error situation on the server.
4. The network went down.
Action: Bring up your network and check it using ping and/or traceroute.
5. Other SAP setup or communication problems.
Action: Check SAP setup or communication line.
If the error was not caused by any of the situations listed above, save all available diagnostic information and report the error to SAP/IBM.

ICLC2557E System error: The attempt to connect to the ICLI server failed because the connection timed out.
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:
Using Host '*sss*', IP address '*sss*', ICLI connection port *nnn*,
and communications protocol TCP/IP

Explanation: The connect to the ICLI server failed because the TCP connection request timed out.

System Action: The ICLI client returns an error to the

calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are several situations which might lead to this message:
1. The processing of the connection request failed on the ICLI server.
Action: Check the ICLI server message file and correct the error situation on server.
2. The network went down or was not set up correctly.
Action: Bring up your network and check it using ping and/or traceroute.
3. There is no route defined from the client to the server host.
Action: Check your network setup and test for a possible connection using ping and/or traceroute. The TCP/IP commands ping and traceroute should work in both directions.
4. Other SAP setup or communication problems.
Action: Check SAP setup or communication line.
If the error was not caused by any of the situations listed above, save all available diagnostic information and report the error to SAP/IBM.

ICLC2558E System error: The attempt to connect to the ICLI server failed.
The ICLI client process immediately returns an error to the calling SAP process.
Diagnostic information:
connect() RC=*nnn*, errno=*nnn*, system message: *sss*
Using Host '*sss*', IP address '*sss*', ICLI connection port *nnn*,
and communications protocol TCP/IP

Explanation: The connect to the ICLI server failed. The system message <sss> gives detail information about the problem.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check which of the situations listed below produces the problem. Correct the situation as described under Action. There are several situations which might lead to this message:
1. The processing of the connection request failed on the ICLI server.
Action: Check the ICLI server message file and correct the error situation on server.
2. The network went down.
Action: Bring up your network and check it using ping and/or traceroute.
3. Other reason as described in <sss>.
Action: Try to resolve this problem using TCP/IP documentation.
If the error was not caused by any of the situations listed above, save all available diagnostic information

and report the error to SAP/IBM.

ICLC2559E   **ICLI server error: The attempt to connect to ICLI server failed because the ICLI server reports that the creation of a new ICLI server thread failed.**
Server message: *sss*
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The connect to the ICLI server failed. The server message <sss> gives detail information about the problem.

System Action: The ICLI client returns an error to the calling SAP process because it is not possible to establish a connection.

User Response: Check the ICLI server message file for the error message pointing to the problem and resolve the problem according to that message and its description.

ICLC2564E   **Internal Error: An unexpected eyecatcher '*sss*' was received from the ICLI server.**
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI client received an unexpected eyecatcher. The reason may be a corrupted data buffer or an ICLI protocol error.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Check the ICLI server message file for related error messages. Also check that you are using the ICLI client which was packaged with the ICLI server in use. If the client is older than the server, a client upgrade might resolve the problem. Otherwise, save all available diagnostic information and report the error to SAP/IBM.

ICLC2565E   **ICLI server error: The ICLI server received an unexpected eyecatcher '*sss*' from the ICLI client.**
The ICLI client process immediately returns an error to the calling SAP process.

Explanation: The ICLI server reports that the client has sent an invalid eyecatcher within the transmission buffer. The reason may be a corrupted data buffer or an ICLI protocol error.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Check for preceding ICLI client messages. Check the ICLI server message file for

related error messages. Also check that you are using the ICLI client which was packaged with the ICLI server in use. If the client is older than the server, a client upgrade might resolve the problem. Otherwise, save all available diagnostic information and report the error to SAP/IBM.

ICLC2571E   **System error in line *nnn* and file *sss*: The ICLI client process cannot continue. Diagnostic information: Function setsockopt(SO_KEEPALIVE) failed with return code = *nnn* errno = *nnn* (0x*xxx*), message:** *sss*

Explanation: The system function setsockopt() failed with return code <nnn>. Message <sss> gives more details.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem reported by errno <nnn> and message <sss> using operating system and TCP/IP documentation. If you cannot resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

ICLC2572E   **System error in line *nnn* and file *sss*: The ICLI client process cannot continue. Diagnostic information: Function setsockopt(SO_SNDBUF) failed with return code = *nnn* errno = *nnn* (0x*xxx*), message:** *sss*

Explanation: The system function setsockopt() failed with return code <nnn>. Message <sss> gives more details.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem reported by errno <nnn> and message <sss> using operating system and TCP/IP documentation. If you cannot resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

ICLC2573E   **System error in line *nnn* and file *sss*: The ICLI client process cannot continue. Diagnostic information: Function setsockopt(SO_RCVBUF) failed with return code = *nnn* errno = *nnn* (0x*xxx*), message:** *sss*

Explanation: The system function setsockopt() failed with return code <nnn>. Message <sss> gives more details.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem reported by errno <nnn> and message <sss> using operating system and TCP/IP documentation. If you cannot

resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

| ICLC2574E | System error in line *nnn* and file *sss*: An error occurred during the initialization phase of the ICLI interface. The socket for communication could not be created. The ICLI client process immediately returns an error to the calling SAP process. Diagnostic information: socket() RC=*nnn*, errno=*nnn*, system message: *sss* |

Explanation: The system function socket() failed with errno <nnn>. Message <sss> gives more details.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Try to resolve the problem reported by errno <nnn> and message <sss> using operating system and TCP/IP documentation. If you cannot resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

| ICLC2575E | Interface error in line *nnn* and file *sss*: The ICLI client process cannot continue. The ICLI client returns an error to the calling SAP process. Details: Service name '*sss*' does not have the expected length of 9 characters, the actual length is *nnn*. |

Explanation: This is an interface error. The ICLI expects to receive a service name that is exactly 9 characters long.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Check the `services` file for the corrupted entry. If you cannot resolve the problem, save all available diagnostic information and report the error to SAP/IBM.

| ICLC2577E | Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. The ICLI client returns an error to the calling SAP process. Details: Protocol error. The ICLI client received *nnn* bytes, but *nnn* bytes were expected. Save all available diagnostic information and report the error to SAP/IBM. |

Explanation: This is an internal ICLI error.

System Action: The ICLI client returns an error to the calling SAP process.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

| ICLC2579E | Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. Details: Function ID *nnn* is not enabled for stream mode transmit. Save all available diagnostic information and report the error to SAP/IBM. |

Explanation: This is an internal ICLI error.

System Action: The ICLI client returns an error to the calling process.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

| ICLC2580E | Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. The ICLI client returns an error to the calling process. Details: Attempt to bundle LOB request error. Save all available diagnostic information and report the error to SAP/IBM. |

Explanation: This is an internal ICLI error which occurred while a database request involving DB2 LOB data was being processed.

System Action: The ICLI client returns an error to the calling process.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

| ICLC2581E | Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. The ICLI client returns an error to the calling process. Details: Function *sss* wanted to add *nnn* bytes to the send buffer, but it is not big enough. Save all available diagnostic information and report the error to SAP/IBM. |

Explanation: The SAP database layer called an ICLI client function with invalid parameters. The maximum possible overall length of the input SQL variables might be exceeded. This error points to a problem within the DBSL or some layer above.

System Action: The requested ICLI function cannot execute. The calling function will get an error return code.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLC5000E Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. The SQL operation cannot be completed. Details: An invalid call sequence was encountered in function *sss*.

Explanation: The ICLI client received a FETCH statement which was not preceded by an OPEN CURSOR statement. This request could not be executed.

System Action: The ICLI client returns with an error. The SQL operation cannot be completed.

User Response: Check your SAP trace files for additional error messages. Save all diagnostics and report the error to SAP/IBM.

ICLC5001E Error in line *nnn* and file *sss*: The ICLI client process cannot continue. Details: Function GetCcsids() received an invalid parameter connection handle (*nnn*).

Explanation: The ICLI client received a request to retrieve the host CCSIDs. This request could not be executed because the connection handle passed by the caller (SAP DBSL) was invalid.

System Action: The ICLI client cannot deliver the requested CCSIDs to the DBSL and the connection to the database cannot be established.

User Response: Check your SAP trace files for additional error messages. Save all diagnostics and report the error to SAP/IBM.

ICLC5002E Internal error in line *nnn* and file *sss*: The ICLI client process cannot continue. Details: Function GetCcsids() invalid ICLI control block pointer *ppp* (hcon=*nnn*)

Explanation: The ICLI client received a request to retrieve the host CCSIDs. This request could not be executed because no valid control block pointer was associated with the connection handle passed by the caller (SAP DBSL).

System Action: The ICLI client cannot deliver the requested CCSIDs to the DBSL and the connection to the database cannot be established.

User Response: Check your SAP trace files for additional error messages. If the error cannot be resolved, save all diagnostics and report the error to SAP/IBM.

ICLC5004E Error in line *nnn* and file *sss*. Details: Function GetCcsids() received an invalid *sss* CCSID pointer *ppp*.

Explanation: The ICLI client received a request to retrieve the host CCSIDs. The ASCII or EBCDIC CCSID cannot be delivered because the caller (SAP DBSL) did not provide a valid pointer for the result.

System Action: The ICLI client cannot deliver the ASCII or EBCDIC CCSID to the DBSL. Normally, the DBSL cannot establish a connection to the database in this case.

User Response: Check your SAP trace files for additional error messages. If the database connection cannot be established and the error cannot be resolved, save all diagnostics and report the error to SAP/IBM.

ICLC5013E ICLC5013E The following ICLI client messages for handle type *nnn* and handle *nnn* were not consumed:

Explanation: The ICLI client detected a condition where error messages it had generated were never retrieved by the calling SAP DBSL layer. To ensure that such messages are not lost, the messages are recorded in the ICLI trace file.

System Action: Processing continues.

User Response: Check the recorded messages for error conditions which might impact your system and act on these messages accordingly. If this error occurs repeatedly, report the error to SAP/IBM.

ICLC5014W ICLC5014W ICLI client message buffer overflow. The following message is lost:

Explanation: The ICLI client detected a condition where an error message it generated would overflow the available message buffer. To ensure that the message is not lost, the message is recorded in the ICLI trace file. The message points to an internal message reporting problem in ICLI.

System Action: Processing continues.

User Response: If you see this error repeatedly, report the error to SAP/IBM.

ICLC5015W *nnn* additional ICLI client messages were lost due to message buffer overflow.

Explanation: The ICLI client detected a condition where error messages it generated would overflow the available message buffer. To avoid that the messages are lost, the messages are recorded in the ICLI trace file. The message points to a internal message reporting problem in ICLI.

System Action: Processing continues.

User Response: If you see this error repeatedly, report the error to SAP/IBM.

ICLC5016E | **ICLC5016E An SQL error message could not be retrieved from the database server.**
The SQL error message cannot be passed to the SAP DBSL layer.

Explanation: The ICLI client tried to retrieve an SQL error message from the ICLI server. This message could not be retrieved and can therefore not be passed to the SAP DBSL layer.

System Action: Processing continues. Message reporting of an SQL error in SAP not possible.

User Response: Check for other messages which might indicate the reason for this behavior. One reason might be that a TCP/IP connection to the ICLI server was dropped. If you cannot resolve the problem, report the error to SAP/IBM.

ICLC5017E | **Error in line *nnn* and file *sss*.**
Number of parameters received (*nnn*) is less than 1 or greater than the DB2 maximum (*nnn*).
The database request cannot be processed.
Check your SAP application to determine the cause of the problem.

Explanation: The ICLI client received a database request from the SAP DBSL which contained more parameters than DB2 can handle.

System Action: The database request cannot be processed.

User Response: Check your SAP application to determine the cause of the problem. If you cannot resolve the error within your SAP application, report the error to SAP/IBM.

ICLC5018E | **Error in line *nnn* and file *sss*:**
The received statement handle (*nnn*) is not within the valid range (0) .. (*nnn*).
The SQL request cannot be processed.
Check for preceding messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from the SAP DBSL that contained an invalid statement handle.

System Action: The SQL request cannot be processed.

User Response: Check for preceding messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5019E | **Error in line *nnn* and file *sss*.**
The sqlIcliSetStmtAttr request contains an invalid statement handle.
Length of statement attribute (*nnn*) is negative: (*nnn*).
The SQL request cannot be processed.
Check for preceding messages which might indicate the cause of the problem.

Explanation: The ICLI client received a sqlIcliSetStmtAttr request from the SAP DBSL which contained an invalid statement handle. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for preceding messages which might indicate the cause of the problem. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5020E | **Error in line *nnn* and file *sss*.**
The sqlIcliSetStmtAttr request contains an invalid statement attribute.
Length of statement attribute (*nnn*), type (*nnn*) is not valid: (*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received a sqlIcliSetStmtAttr request from the SAP DBSL which contained an invalid statement attribute. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5021E | **Error in line *nnn* and file *sss*.**
The sqlIcliSetStmtAttr request contains an invalid statement attribute.
Statement attribute (*nnn*) has an invalid type (*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received a sqlIcliSetStmtAttr request from the SAP DBSL which contained an invalid statement attribute. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and

report the error to SAP/IBM.

ICLC5022E **Error in line** *nnn* **and file** *sss***: The ICLI client received an invalid pointer. The request terminates with an error. Invalid pointer: (***ppp***). Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an invalid pointer. The reason is probably an internal ICLI problem.

System Action: The request terminates with an error.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5023E **Error in line** *nnn* **and file** *sss***. Function** *sss***: connection handle:** *nnn* **is not allocated. The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid connection handle. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5024E **Internal error in line** *nnn* **and file** *sss***: The SQL request cannot be processed. Function** *sss***. Check for preceding messages which might indicate the cause of the problem.**

Explanation: The ICLI client detected an internal error.

System Action: The SQL request cannot be processed.

User Response: Check for preceding messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5026E **Error in line** *nnn* **and file** *sss***. Function** *sss***: cannot use statement handle:** *nnn***, statement had not been freed before. (Old statement handle was** *nnn***). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid statement handle. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5027E **Error in line** *nnn* **and file** *sss***. Function** *sss***: received an invalid DB2 locator type** *nnn* **for locator** *xxx***. The LOB request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an request involving a DB2 LOB locator of an invalid type. The reason is probably an internal SAP DBSL problem.

System Action: The LOB request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5028E **Error in line** *nnn* **and file** *sss***. Function** *sss***: statement handle:** *nnn* **is not in use. The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL that contained a statement handle which had not been allocated before. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5029E **Error in line** *nnn* **and file** *sss***. Function** *sss***: handle** *nnn* **of type** *nnn* **is outside of the allowed range (***nnn* **..** *nnn***). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid handle. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which

might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5030E   Error in line *nnn* and file *sss*.
Function *sss*: handletype *nnn* is invalid.
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid handle. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5035E   Error in line *nnn* and file *sss*.
Function *sss*: returned sqlda_len (*nnn*) is greater than the
maximum allowed size of an SQLDA (*nnn*).
The reason is probably an internal ICLI problem.
The SQL request cannot be processed.
Check for ICLI server messages which might indicate the cause of the problem.

Explanation: The ICLI client received a response from the ICLI server which contained an invalid length field. The reason is probably an internal ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for ICLI server messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5036E   Error in line *nnn* and file *sss*.
Function *sss* was called with an column number (*nnn*) which is
not within the valid range of (*nnn*..*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from SAP DBSL for a column number which is invalid. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5037W   Warning in line *nnn* and file *sss*.
Function *sss* was called for column (*nnn*) with a maximal column
name length of (*nnn*), which is not sufficient (*nnn*).
Processing continues, but the column name shown in SAP is truncated.

Explanation: The ICLI client received an SQL request from SAP DBSL for a column number with an invalid length The reason is probably an internal SAP DBSL problem.

System Action: Processing continues, but the column name shown in SAP is truncated.

User Response: If the warning occurs repeatedly, report it to SAP/IBM.

ICLC5038W   Warning in line *nnn* and file *sss*.
Function *sss* was called for column (*nnn*). The column specification contains an
unsupported DB2 data type (*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from SAP DBSL with an unsupported data type specification. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5039E   Error in line *nnn* and file *sss*.
Allocation of *nnn* bytes failed in function *sss*.
The SQL request cannot be processed.
Check for memory shortage conditions on your application server.

Explanation: The ICLI client encountered an out-of-memory condition when it tried to dynamically allocate memory.

System Action: The SQL request cannot be processed.

User Response: Check for memory shortage conditions on your application server.

ICLC5040E   Error in line *nnn* and file *sss*.
Function *sss*() is not supported in this ICLI driver.
The SQL request cannot be processed.
Check your ICLI and SAP kernel maintenance levels.

Explanation: The ICLI client received an SQL request

from SAP DBSL which is not supported The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check your ICLI and SAP kernel maintenance levels. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5041E   **Internal error in line** *nnn* **and file** *sss*.
Function *sss* **handle type (***nnn***) statement handle (***nnn***): diag info invalid.**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client encountered an internal problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5042E   **Internal error in line** *nnn* **and file** *sss*.
Function *sss* **invalid control block pointer (***ppp***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client encountered an internal problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5043E   **Internal error in line** *nnn* **and file** *sss*.
Function *sss* **invalid sqlda pointer (***ppp***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client encountered an internal problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5044E   **Internal error in line** *nnn* **and file** *sss*.
Function *sss* **invalid sqlcode pointer (***ppp***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client encountered an internal problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5045E   **Error in line** *nnn* **and file** *sss*.
Function *sss* **was called with an unsupported DB2 SQLTYPE (***nnn***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from SAP DBSL for an unsupported SQLTYPE. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5046E   **Error in line** *nnn* **and file** *sss*.
Function *sss* **was called with an unsupported input/output type (***nnn***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL request from SAP DBSL for an unsupported input/output type. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5047E   **Error in line** *nnn* **and file** *sss*.
Function *sss* **was called with an unsupported DB2 locator of SQLCTYPE (***nnn***).**
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the problem.

Explanation: The ICLI client received an SQL LOB request from SAP DBSL with an unsupported locator

type. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5048E   **Error in line** *nnn* **and file** *sss*. **Function** *sss* **was called with an unsupported DB2 TargetCType (*nnn*). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL LOB request from SAP DBSL with an unsupported TargetCType. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5049E   **Error in line** *nnn* **and file** *sss*. **The received connection handle (*nnn*) is not within in the valid range of (1) .. (*nnn*). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid connection handle. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5050E   **Error in SQLConnect on Connection** *nnn*: **Required parameter** *sss* **is not set. The SQLConnect request cannot be processed. Report the error to SAP/IBM.**

Explanation: The ICLI client received a connection request which could not be executed because a required parameter is missing. The reason is probably an internal DBSL problem.

System Action: The SQLConnect request cannot be processed.

User Response: Report the error to SAP/IBM.

ICLC5051E   **Error in line** *nnn* **and file** *sss*. **Invalid indicator value was received for output value (*nnn*). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid indicator value. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5052E   **Error in line** *nnn* **and file** *sss*. **Invalid length (*nnn*) was received for output value (*nnn*). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid output length field. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5053E   **Error in line** *nnn* **and file** *sss*. **Invalid indicator value (*nnn*) length (*nnn*) received for output value (*nnn*). The SQL request cannot be processed. Check for other messages which might indicate the cause of the problem.**

Explanation: The ICLI client received an SQL request from the SAP DBSL which contained an invalid indicator value. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5054E   **Error in line** *nnn* **and file** *sss*. **Output column (*nnn*): Column length (*nnn*) is less than received length (*nnn*), The SQL request cannot be processed. Check for other messages which might**

indicate the cause of the problem.

Explanation: The ICLI client received data from the ICLI server containing data which does not match the length expected by SAP DBSL. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5055E **Error in line** *nnn* **and file** *sss***.**
Input value (*nnn***): column length (***nnn***)**
is less
than received length (*nnn***).**
The SQL request cannot be processed.
Check for other messages which might
indicate the cause of the problem.

Explanation: The ICLI client received data from the ICLI server containing data that does not match the length expected by SAP DBSL. The reason is probably an internal DBSL/ICLI problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5056E **Error in line** *nnn* **and file** *sss***.**
Invalid input buffer received in
function *sss*
while retrieving CCSIDs from the ICLI
server.
The SQL request cannot be processed.
Check for other messages which might
indicate the cause of the problem.

Explanation: The ICLI client encountered an internal error when retrieving CCSIDs from the ICLI server.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5057E **Error in line** *nnn* **and file** *sss***.**
The ICLI client encountered a problem
while retrieving CCSIDs from the ICLI
server.
The data buffer could not be used in
function *sss***, rc is (***nnn***).**
The request cannot be processed.
Check for other messages which might
indicate the cause of the problem.

Explanation: The ICLI client encountered an internal

error when retrieving CCSIDs from the ICLI server.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5058E **Error in line** *nnn* **and file** *sss***.**
Function *sss* **did not receive any data**
from the ICLI server.
The request cannot be processed.
Check for ICLI server messages or
communications errors
which might indicate the cause of the
problem.

Explanation: The ICLI client encountered an internal error when retrieving CCSIDs from the ICLI server.

System Action: The request cannot be processed.

User Response: Check for ICLI server messages or communications errors which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5059E **Error in line** *nnn* **and file** *sss***.**
Function *sss* **encountered an**
out-of-memory condition on the ICLI
server.
The request cannot be processed.
Check for ICLI server messages or
communications errors which might
indicate the cause of the error.

Explanation: The ICLI client encountered an internal error when retrieving data from the ICLI server.

System Action: The request cannot be processed.

User Response: Check for ICLI server messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5060E **Interface error in line** *nnn* **and file** *sss***.**
Invalid maximal length *nnn* **for**
SQLCTYPE *nnn***.**
The request cannot be processed.
Check for other messages which might
indicate the cause of the error.

Explanation: The ICLI client encountered an interface error when processing the max length filed of the specified SQLTYPE.

System Action: The request cannot be processed.

User Response: Check for additional messages which might indicate the reason for this message. If the error

persists, save all available diagnostics and report the error to SAP/IBM.

ICLC5061E   **Error in line *nnn* and file *sss*.**
Function *sss* was called with an unsupported SQLCTYPE *nnn*.
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received an SQL request from SAP DBSL for an unsupported SQLCTYPE The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5070E   **Error in function *sss* line *nnn* and file *sss*.**
Value *nnn* exceeds maximum of *nnn*.
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client encountered an error when processing a field. Value exceeds allowed maximum length.

System Action: The request cannot be processed.

User Response: Check for additional messages which might indicate the reason for this message. If the error persists, save all available diagnostics and report the error to SAP/IBM.

ICLC5071E   **Interface error in line *nnn* and file *sss*.**
Function *sss* was called with an unsupported sqllen (*nnn*) for that SQLTYPE (*nnn*).
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client encountered an interface error when processing the length filed of a (short/long integer) SQLDATA type.

System Action: The request cannot be processed.

User Response: Check for additional messages which might indicate the reason for this message. If the error persists, save all available diagnostics and report the error to SAP/IBM.

ICLC5072E   **Error in line *nnn* and file *sss*.**
Function *sss*, use count of environment handle *nnn* is null.
The environment handle is either not allocated or has not been freed.

The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received a request from the SAP DBSL for an environment handle which is either not allocated or has not been freed. The reason is probably an SAP DBSL internal problem.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5073E   **Error in line *nnn* and file *sss*.**
Function *sss*, unsupported arguments received, attribute: *nnn* value: *nnn*.
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received a request from the SAP DBSL with unsupported arguments. The reason is probably an SAP DBSL internal problem.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5074E   **Error in line *nnn* and file *sss*.**
Function *sss*, no valid database connection exists for HDBC handle *nnn*.
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received a request from the SAP DBSL with a database connection handle for which no connection to DB2 exists. The reason is probably an SAP DBSL internal problem.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5075E   **Error in line *nnn* and file *sss*.**
Function *sss*, no more free connection handles are available.
The request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received a request from the SAP DBSL with a database connection handle for which no connection to DB2 exists. The reason is probably an internal SAP DBSL problem.

System Action: The request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5076W Error in line *nnn* and file *sss*.
Function *sss*, error code *nnn* occurred while disconnecting
from the database using connection handle *nnn*.
Possibly, the ICLI server thread had already terminated.
The request cannot be processed.

Explanation: The ICLI client sent a disconnect request to the ICLI server. This request could not be delivered. The most probable reason is that the ICLI server thread had already terminated.

System Action: The request cannot be processed.

User Response: If the message is written in a situation where the ICLI server thread has already terminated, this message can be ignored. If this is not the case, then the error may have been caused by a communications failure (TCP/IP problem) between application server and database host.

ICLC5077E Error in line *nnn* and file *sss*.
Function *sss* was called with the unsupported option (*nnn*),
parameter value (*nnn*), parameter length (*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received a request with unsupported argument values. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5078I Data was truncated in function *sss*, because the result buffer was not large enough.
Buffer size is *nnn*, result length is *nnn*.
As much data as possible is stored in the buffer.

Explanation: The buffer provided by the SAP DBSL was not large enough to store the complete result.

System Action: As much data as possible was stored in the buffer.

User Response: If you are confronted with useless data due to truncation, report the problem to SAP.

ICLC5079E Error in line *nnn* and file *sss*.
Function *sss* was called with an unsupported attribute type (*nnn*).
The SQL request cannot be processed.
Check for other messages which might indicate the cause of the error.

Explanation: The ICLI client received an SQL request from SAP DBSL for an unsupported attribute. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5080S ICLI_ASSERT failed in file *sss* line *nnn*.
The current function immediately returns an error to the calling layer.
Nothing is processed.
Save all available diagnostics and report the error to IBM.

Explanation: The ICLI detected an invalid condition. The reason is probably an internal ICLI problem.

System Action: The current function immediately returns an error to the calling layer. Nothing is processed.

User Response: Save all available diagnostics and report the error to IBM.

ICLC5081E Request to establish a TCP/IP connection to
SAPDBHOST *sss* (IP address *sss*, port *nnn*) which is active on z/OS system *sss* was not successful. Check the preceding message for details on the problem and try to solve it.

Explanation: The ICLI client could not establish a TCP/IP connection to the ICLI server running on the database host specified by the described IP address, connection port, and z/OS system. The z/OS system is identical to the result of the uname -n command of z/OS Unix System Services. It is the node name of that machine, which is set by the SYSNAME sysparm (specified at IPL), and usually differentiates z/OS machines running at a single location.

System Action: The ICLI client returns the error to the calling process. SAP will retry the request.

User Response: If all attempts by SAP to establish a connection fail, check for preceding messages which indicate the reason for this message and resolve the problem. Otherwise, report the error to SAP/IBM.

ICLC5082I   **Established a TCP/IP connection to SAPDBHOST** *sss* **(IP address** *sss***, port** *nnn***), which is active on z/OS system** *sss***.**

Explanation: The ICLI client established a TCP/IP connection to the ICLI server, which runs on the reported database host with the reported IP address, connection port, and z/OS system. The z/OS system is identical to the result of the uname -n command of z/OS Unix System Services. It is the node name of that machine, which is set by the SYSNAME sysparm (specified at IPL), and it usually differentiates z/OS machines running at a single location.

System Action: Processing of the request to connect to the database continues.

User Response: None.

ICLC5083E   **Error in file** *sss* **line** *nnn***: ICLI function** *sss* **failed with return code** *nnn***. The ICLI tries to continue processing, but some functionality might be lost.**

Explanation: An internal ICLI function detected an invalid condition. The reason is probably an internal ICLI problem or an invalid parameter received from the calling DBSL.

System Action: The ICLI tries to continue processing, but some functionality might be lost.

User Response: Save all available diagnostics, especially the SAP work files and ICLI trace files, and report the error to SAP/IBM.

ICLC5084E   **Application error in file** *sss* **line** *nnn***: SQLExtendedBind was called with an invalid fBindCol value** *nnn***. Save all available diagnostics and report the error to SAP.**

Explanation: The SAP database layer called the ICLI function SQLExtendedBind and used a wrong value for the parameter fBindCol. The only valid values are SQL_TRUE (1) and SQL_FALSE (0).

System Action: The SQLExtendedBind function returns with this error. Further action depends on the application (DBSL).

User Response: Save all available diagnostics and report the error to SAP.

ICLC5085I   **No performance data available which matches the criteria specified in SAP transaction ST04.**

Explanation: A user requested performance data within SAP transaction ST04 using additional qualification values. There was no data available which satisfies the given qualification.

System Action: Processing continues.

User Response: None.

ICLC5086E   **DB2 performance statistics error. Function db2ps() returned with RC=***nnn***. Processing continues.**

Explanation: A user requested performance data within SAP transaction ST04. This request could not be handled due to a DB2 or ICLI problem.

System Action: Processing continues.

User Response: There might be preceding ICLI client messages giving more specific information. Also check the ICLI server message file for error messages. If you cannot resolve the problem, set the ICLI client trace to 3 and the ICLI server trace to 6, reproduce the problem, save all available diagnostics and report the error to SAP.

ICLC5087E   **Request to establish a TCP/IP connection to SAPDBHOST** *sss* **(IP address** *sss***, port** *nnn***) was not successful. Check the preceding message for details on the problem and try to solve it.**

Explanation: The ICLI client could not establish a TCP/IP connection to the ICLI server running on the database host specified by the described IP address and connection port.

System Action: The ICLI client returns the error to the calling process. SAP will retry the request.

User Response: If all attempts by SAP to establish a connection fail, check for preceding messages which indicate the reason for this message and resolve the problem. Otherwise, report the error to SAP/IBM.

ICLC5088E   **Error in line** *nnn* **and file** *sss***. Function** *sss* **was called with an unsupported value length** *nnn* **for parameter** *sss***. Expected length is** *nnn***. The SQL request cannot be processed.**

Explanation: The ICLI client received a request with unsupported argument values. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request cannot be processed.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5089I   **Function** *sss* **was called with an unknown parameter type (***nnn***) The SQL request is ignored.**

Explanation: The ICLI client received a request with unsupported argument values. The reason is probably an internal SAP DBSL problem.

System Action: The SQL request is ignored by the ICLI.

User Response: Check for other messages which might indicate the reason for this message. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5090E   **The ICLI server reports a problem. ICLI server message:** *sss*

Explanation: During a database connect the ICLI server needs to determine the CCSIDs used in DB2 and fetch the corresponding translation tables. This step was not successful.

System Action: Processing cannot continue without correct translation tables. The DBSL will stop this work process.

User Response: Check the ICLI server message to get an indication of the reason for this problem. If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5092E   **The codepage translation tables are corrupted at index** *nnn***.**

Explanation: During a database connect the ICLI server needs to determine the CCSIDs used in DB2 and fetch the corresponding translation tables. The tables transferred to the ICLI client are not bijective.

System Action: Processing cannot continue without correct translation tables. The DBSL will stop this work process.

User Response: Check also the ICLI server message file. Check if you have a network problem (network corrupts data). If the error cannot be resolved, save all available diagnostics and report the error to SAP/IBM.

ICLC5093E   **Configuration error: This is DB2 version** *c.c.c***. Only DB2 V6, V7, and (in compatibility mode) V8 are supported.**

Explanation: The SAP work process connected to a DB2 subsystem whose DB2 version is not supported using ICLI.

System Action: The ICLI will disconnect from this DB2 subsystem and cannot process any requests until the work process connects to another DB2 subsystem.

User Response: Check if you have configured your

system to use the correct DB2 subsystem. ICLI supports only DB2 V6, V7, and (in compatibility mode) V8. Use DB2 Connect for DB2 V8 in new-function mode or later DB2 versions.

ICLC5094W   **Warning in line** *nnn* **and file** *sss***. Function** *sss***, unsupported arguments received, handle type (***nnn***) handle (***nnn***), attribute:** *nnn* **value:** *nnn***. The request is ignored.**

Explanation: The ICLI client received a request from the SAP DBSL with unsupported arguments.

System Action: The request is ignored.

User Response: Ignore this message if it does not correlate with other error messages.

ICLI instance setup program messages

ICLI3000I   Message catalog *sss* could not be opened.
Default messages will be used instead.

Explanation: The message catalog that is used to retrieve messages could not be opened.

System Action: The command will proceed using default messages.

User Response: Set the NLS environment variables to be consistent with a language feature installed on your system. In particular, the environment variable NLSPATH must point to the directory where the message catalog resides.

ICLI3001E   **The command was invoked with an invalid option (-*sss*).**
The command terminates. Invoke the command with -?
for command line help.

Explanation: The command was invoked with an invalid option.

System Action: The command terminates.

User Response: Determine valid options by invoking the command with -? and retry with valid options.

ICLI3002E   **The command was invoked with an insufficient number of**
parameters. The command terminates.
Invoke the command with -? for help.

Explanation: The command was invoked with too few parameters.

System Action: The command terminates.

User Response: Determine valid parameters by invoking the command with -? and retry with valid parameters.

ICLI3003E   **The command was invoked with too many parameters (*sss*).**
The command terminates.
Invoke the command with -? for help.

Explanation: The command was invoked with too many parameters.

System Action: The command terminates.

User Response: Determine valid parameters by invoking the command with -? and retry with valid parameters.

ICLI3004S   **Error using syscall *sss*:**
args='*sss*',
rc=*sss*,
errno=*sss*.

The command terminates.

Explanation: The command encountered an error.

System Action: The command terminates.

User Response: Report the exact error message to IBM.

ICLI3005I   **Setting up the ICLI server instance:**
Working directory: *sss*
STEPLIB: *sss*
Plan: *sss*
Connection port: *sss*

Explanation: The command displays the parameters with which it was invoked.

System Action: The command proceeds with the parameters displayed.

User Response: None.

ICLI3006I   **The ICLI server start script *sss***
was created successfully.

Explanation: The command informs you that the ICLI server start script has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3007I   **The JCL file**
sss
for the ICLI server's started task
was created successfully.

Explanation: The command informs you that the ICLI server started task JCL file has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3008I   **The new ICLI server instance was set up successfully.**
Change the start script and the JCL file of the
started task accordingly.

Explanation: The command informs you that it has finished successfully.

System Action: The command terminates successfully.

User Response: Adjust the ICLI server start script and the ICLI server started task JCL file to your needs. For details, refer to Chapter 4, "Setting up instances of the ICLI server," on page 65.

ICLI3009E **The directory** *sss*
could not be created, because a file
with the same name already exists.
The command terminates.

Explanation: The command could not create a directory because a file with that name already exists.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the file with the name of the directory.

ICLI3010E **The directory** *sss*
could not be created (errno=*sss***).**
The command terminates.

Explanation: The command could not create a directory. The errno value indicates the reason. For a description of the meaning of errno values, refer *Using REXX and OpenMVS*.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the reason the directory could not be created.

ICLI3011I **The directory** *sss*
was created successfully.

Explanation: The command informs you that a directory has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3012E **The status of directory** *sss*
could not be obtained (errno=*sss***).**
The command terminates.

Explanation: The command could not obtain the status of a directory. The errno value indicates the reason. For a description of the meaning of errno values, refer to *Using REXX and OpenMVS*.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the reason the status of the directory could not be obtained.

ICLI3013E **The softlink** *sss*
could not be created, because an object
other than
a softlink with the same name already
exists.
The command terminates.

Explanation: The command could not create a softlink

because a file or other object with that name that is not a softlink already exists.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the object with the name of the softlink.

ICLI3014E **Softlink** *sss* **on file** *sss*
could not be created (errno=*sss***).**

Explanation: The command could not create a softlink. The errno value indicates the reason. For a description of the meaning of errno values, refer to *Using REXX and OpenMVS*.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the reason the softlink could not be created.

ICLI3015I **Softlink** *sss* **on file** *sss*
was created successfully.

Explanation: The command informs you that a softlink has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3016E **The status of softlink** *sss*
could not be obtained (errno=*sss***).**
The command terminates.

Explanation: The command could not obtain the status of a softlink. The errno value indicates the reason. For a description of the meaning of errno values, refer to *Using REXX and OpenMVS*.

System Action: The command terminates.

User Response: Either invoke the command with a different working directory or remove the reason the status of the softlink could not be obtained.

ICLI3017I **The ICLI server JCL job file**
sss
was created successfully

Explanation: The command informs you that the ICLI server JCL job file has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3018I   **The ICLI server started task environment file**
sss
was created successfully

Explanation: The command informs you that the ICLI server started task environment file has been created successfully.

System Action: The command proceeds with further actions.

User Response: None.

ICLI3099S   **Internal error on line** *sss***:** *sss***.**

Explanation: The command encountered an internal error.

System Action: The command terminates.

User Response: Report the exact error message to IBM.

ICLI server messages

ICLS1001S   System error in file *sss* and line *nnn*:
     The system call
     pthread_key_create() failed during the
     initialization
     of the ICLI server.
     The key for *sss* data could not be

created:
     RC = *nnn*, errno = *nnn* (*xxx*), errno2 =
     xxx
     Message: *sss*.

Explanation: The system call pthread_key_create() failed during the initialization of the ICLI server. 'Message' describes the cause of the failure. One possible cause is memory shortage.

System Action: The ICLI server abends.

User Response: Try to remove the cause of the failure. In case of memory shortage, adjust your REGION size and restart the ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1002S   Error creating mutex semaphore
     attribute R/C = *nnn*
     errno = *nnn* (*xxx*), errno2 = *xxx*
     Message: *sss*.

Explanation: This is an internal ICLI error. System call pthread_mutexattr_init() failed for the reason indicated in 'Message'. One possible cause is memory shortage.

System Action: ICLI server is stopped.

User Response: Try to remove the cause for the failure. In case of memory shortage adjust your REGION size and restart the ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1003E   System error: Memory for an internal
     cleanup information
     structure could not be allocated:
     errno = *nnn* (*xxx*), errno2 = *xxx*
     Message: *sss*.
     The connection request is rejected. This
     ICLI
     thread is stopped.

Explanation: Memory for an internal structure could not be obtained. One possible reason is memory shortage. The connection request that caused this ICLI thread to be created will fail.

System Action: This ICLI thread is stopped.

User Response: Increase your REGION size before starting the ICLI server or switch to another ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1004S   Error: The DB2 module *sss* could not be
     loaded from DB2
     library 'prefix.SDSNLOAD.suffix' in file
     sss line *nnn*.
     Ensure that the DB2 library 'pre
     fix.SDSNLOAD.suffix'

is defined in your STEPLIB.

Explanation: The dynamic LOAD of DB2 module <sss> failed. The module could not be found in the usual z/OS search path for LOAD modules.

System Action: ICLI server is stopped.

User Response: Make sure that DB2 library 'prefix.SDSNLOAD.suffix' is in your STEPLIB concatenation.

ICLS1005S **Error: An invalid port number (*sss*) was specified for**
command line option -PORT.
The valid range for port numbers is
1024 to 65535.

Explanation: The port number specified for command line option -PORT was invalid.

System Action: The ICLI server abends.

User Response: Restart the ICLI server using a valid port number. Port numbers above 5000 are preferable, because it is less likely that they are already assigned to another application. For detailed information on how to select an ICLI connection port, refer to "ICLI server command line options" on page 101.

ICLS1006S **Error: No port number was specified for**
command line option
'-PORT'.
The valid range for port numbers is
1024 to 65535.

Explanation: The command line option -PORT was invoked without a port number.

System Action: The ICLI server is stopped.

User Response: Restart the ICLI server using a valid port number after the -PORT option. The valid range for port numbers is 1024 to 65535, but it is recommended to use port numbers above 5000, because it is less likely that they are already assigned to other applications. For detailed information on how to select an ICLI connection port, refer to "ICLI server command line options" on page 101.

ICLS1007E **Error: The plan name *sss*, which was**
specified for command line
option -PLAN, is too long. It will be
truncated to *nnn*
characters.

Explanation: The plan name that was specified after the -PLAN command line option was too long.

System Action: The plan name is truncated to the number of characters indicated in the message.

User Response: Make sure that you specify a valid plan name with the -PLAN option. If the truncated plan name does not refer to a valid plan name in your

installation, the ICLI thread will issue further messages. You may have to restart the ICLI server with a correct plan name.

ICLS1008S **Error: No plan name was specified for**
command line
option '-PLAN'.

Explanation: The command line option -PLAN was invoked without a plan name.

System Action: The ICLI server is stopped.

User Response: Restart the ICLI server specifying a valid plan name after the -PLAN option, or omit the -PLAN option such that the ICLI server uses the default plan name.

ICLS1009S **Error: The specified command line**
option *sss* is not supported.
Call '*sss* -HELP' for command line help.

Explanation: An unsupported command line option was specified.

System Action: The ICLI server is stopped.

User Response: Restart the ICLI server using valid command line options.

ICLS1010S **Error: No service entry was found for**
service name *sss* and
protocol *sss*.

Explanation: The specified service name <sss> was not found in /etc/services or <TCPIP>.ETC.SERVICES or <ICLIRUN>.ETC.SERVICES.

System Action: The ICLI server is stopped.

User Response: Define a service name with the correct protocol ('tcp') in /etc/services or <TCPIP>.ETC.SERVICES or <ICLIRUN>.ETC.SERVICES and restart the ICLI server. For more information on how to define the service name for an ICLI server, refer to Chapter 4, "Setting up instances of the ICLI server," on page 65

ICLS1011I **The ICLI server is ready to accept**
connection requests
from ICLI clients.
sss* Version: *nnn.nnn.nnn
ICLI server for SAP Version: *sss*
ICLI server protocol version: *nnn*
ICLI server internal version: *nnn*
The ICLI server is listening on TCP port
nnn
and IP address <*sss*>.
ICLI trace level: (*nnn*), message level:
(*sss*).
ICLI message file number: (*nnn* of *nnn*
message files).
List of DB2 SQL codes that will not be

**reported as
ICLI messages: '*sss*'**

Explanation: This informational message is issued at ICLI server startup. It indicates that the ICLI server is ready to accept connection requests from ICLI clients.

System Action: None.

User Response: It is now possible to start SAP applications that access the DB2 database.

ICLS1012I **Signal *nnn* (*sss*) was received. The ICLI server is stopping.
All threads will terminate. Threads that are working
in DB2 will terminate when they return from DB2.**

Explanation: The ICLI server received either a SIGINT or a SIGTERM signal. SIGTERM is sent when the user issues a 'kill <pid>' command or uses the TERM option on the MODIFY system command. SIGINT is sent when the ICLI server is interrupted by a 'kill -2 <pid>' or a CTRL-C from the UNIX System Services command prompt.

System Action: The ICLI server is stopped. All ICLI threads will terminate. Threads that are working in DB2 will terminate when they return from DB2.

User Response: None.

ICLS1013W **Diagnostic information: Signal SIGPIPE was received but will
be ignored.
The ICLI server continues processing, but handles the
return call of the send call.**

Explanation: A TCP send call got a SIGPIPE signal, but the ICLI server will continue processing.

System Action: The ICLI server ignores the signal, but handles the return code of the send call.

User Response: None.

ICLS1014E **A segmentation violation occurred; Signal SIGSEGV was
received. A CEEDUMP with title *sss* is produced.
This thread will terminate. All other threads
continue processing.**

Explanation: A segmentation violation occurred. The CEEDUMP will be written to DD:CEEDUMP. If you started the server from UNIX System Services, then the CEEDUMP is placed in the HFS in one of the following directories in the specified order: 1) the directory found in environment variable _CEE_DMPTARG, if found 2) the current working directory, if this is not the root directory (/), and the directory can be written to 3) the

directory found in environment variable TMPDIR (this environment variable defines the location of a temporary directory if it is not /tmp) 4) the /tmp directory Save CEEDUMP for further problem analysis by SAP/IBM.

System Action: The thread that encountered the problems is terminated. All other threads continue processing.

User Response: Save all available diagnostic information including the CEEDUMP and report the error to SAP/IBM.

ICLS1015S **Error setting mutex semaphore attribute
R/C = *nnn*
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.**

Explanation: This is an internal ICLI error. System call pthread_mutexattr_setkind_np() failed for the reason indicated in 'Message'.

System Action: ICLI server is stopped.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1016S **System error in file *sss* line *nnn*: The ICLI server failed
during a call of function pthread_mutex_init():
RC = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
If the error was caused by memory shortage, adjust
the REGION size of the ICLI server and restart it.**

Explanation: The ICLI server failed during a call of the function pthread_mutex_init(). The cause of the failure is described in 'Message'. One possible cause is memory shortage.

System Action: The ICLI server is stopped.

User Response: Try to remove the cause of the failure. In case of memory shortage, adjust your REGION size and restart the ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1017S **System error: The ICLI server failed during a call of
function pthread_cond_init():
RC = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
If the error was caused by memory shortage, adjust
the REGION size of the ICLI server and restart it.**

Explanation: The ICLI server failed when it tried to call the function pthread_mutex_init(). 'Message'

describes the cause of the failure. One possible cause is memory shortage.

System Action: ICLI server is stopped.

User Response: Try to remove the cause for the failure. In case of memory shortage adjust your REGION size and restart the ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1018S **System error: An error was encountered when the function socket() was called to create a socket of domain** *nnn*, **type** *nnn*, **protocol** *nnn*. **errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss*.

Explanation: The system call socket() returned an error. This error is described by the reason indicated in 'Message'.

System Action: ICLI server is stopped.

User Response: If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1019S **System error: An error was encountered when the function bind() was called to bind a socket** *nnn* **to port** *nnn* **using IP address <***sss***>. RC =** *nnn* **errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss*. **The port may be in use by another application.**

Explanation: The system call bind() returned an error. This error is described by the error message given in 'Message'. One cause for this error is that the port is still used by another application. For details about selecting the ICLI connection port, refer to "The ICLI connection port" on page 99.

System Action: The ICLI server is stopped.

User Response: Choose a port number which is not yet used by another application.

ICLS1020S **System error: An error was encountered when the function listen() was called for socket** *nnn*, **backlog (connection request queue)** *nnn*. **errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss*.

Explanation: The system call listen() returned an error. This error is described by the error message given in 'Message'.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server.

ICLS1021W **System error: The ICLI server could not create a socket while in retry mode. This message indicates that TCP/IP is still down. errno =** *nnn* **(***xxx***), errno2 =** *xxx* **The ICLI server will continue to stay in retry mode and will retry to perform the check at 5 second intervals.**

Explanation: The ICLI server encountered an error when it attempted to call the function socket(). The check if TCP/IP is up again failed.

System Action: The ICLI server waits 5 seconds until it retries the check.

User Response: Restart TCP/IP.

ICLS1022E **System error: The function call accept() could not be executed. Function accept() was interrupted for: socket** *nnn* **RC =** *nnn* **errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss*

Explanation: The system call accept() was interrupted. The reason for the interrupt is described in 'Message'.

System Action: ICLI server resumes its operation.

User Response: Check the z/OS Console for possible TCP/IP errors.

ICLS1023I **Connection request from in_addr** *sss* **has been accepted. The client has successfully connected to this ICLI server. Received socket is** *nnn*. **The new thread (***xxx***) will be using port** *nnn*.

Explanation: If the ICLI server environment variable ICLI_MSGLEVEL is set to I, this message is issued for every client that connects successfully to this ICLI server. The thread number <xxx> is the number with which all further messages for this new thread will be prefixed.

System Action: None.

User Response: None.

ICLS1024E **System error in file** *sss* **line** *nnn*: **The system call pthread_mutex_lock() failed: RC =** *nnn* **errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss*.

Explanation: The system call pthread_mutex_lock()

failed. The explanation for the failure is given in 'Message'.

System Action: ICLI processing continues.

User Response: If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLS1025E **System error: There are no more free ports available:**
RC = *nnn*
This ICLI thread terminates. All other ICLI threads
continue processing.

Explanation: The ICLI server could not get a free PORT for a new ICLI work thread.

System Action: This ICLI thread stops. All other ICLI threads continue processing.

User Response: Determine the cause of the problem by reading the previous message. This can be any of ICLS1026E, ICLS1033E, ICLS1034E, or ICLS1048E.

ICLS1026E **System error: The system call bind() of socket *nnn* to any port**
failed.
RC = *nnn*
errno = *nnn* **(*xxx*), errno2 =** *xxx*
Message: *sss*.

Explanation: The system call bind() returned an error. This error is described by the error message given in 'Message'. If you use the Common INET stack, one cause for this error is that the number of available ports defined by INADDRANYCOUNT, INADDRANYPORT and PORTRANGE is too small to serve all the ICLI threads.

System Action: This ICLI server thread is stopped.

User Response: Solve the problem described in 'Message', e.g., extend the port range specified by the TCP parameters INADDRANYCOUNT, INADDRANYPORT, and PORTRANGE.

ICLS1027E **System error: The system call pthread_attr_init() failed:**
RC = *nnn*, **errno =** *nnn* **(*xxx*), errno2 =** *xxx*
Message: *sss*.
The new thread could not be created.
The connection attempt of the ICLI client failed.
If the error was caused by memory shortage, adjust the REGION size.

Explanation: The system call pthread_attr_init() failed. The reason is described in 'Message'.

System Action: The new thread cannot be created.

The connection attempt by the ICLI client fails.

User Response: Try to remove the cause of the failure, in case of memory shortage adjust your REGION size. Either retry the connection attempt or start another ICLI server and try to connect to this new server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1028E **System error: The system call pthread_attr_setstacksize() failed.**
The stack size for a new thread could not be set to *nnn*:
RC = *nnn*, **errno =** *nnn* **(*xxx*), errno2 =** *xxx*
Message: *sss*.
The new thread could not be created.
The ICLI client's connection attempt failed.
If the error was caused by memory shortage, adjust the REGION size.

Explanation: The system call pthread_attr_setstacksize() failed. The stack size for a new thread could not be set.

System Action: The new thread cannot be created. The connection attempt by the ICLI client fails.

User Response: Try to remove the cause of the failure. In case of memory shortage, adjust your REGION size. Either retry the connection or start another ICLI server and try to connect to this new server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1029E **System error: The system call pthread_create() failed:**
RC = *nnn*, **errno =** *nnn* **(*xxx*), errno2 =** *xxx*
Message: *sss*.
A new thread could not be created.
All previously created threads continue processing.

Explanation: The system call pthread_create() failed. The new thread could not be created.

System Action: The new thread cannot be created. The connection attempt by the ICLI client fails. All previously created threads continue processing.

User Response: Remove the cause of the failure. In case of memory shortage check your REGION size. Either retry to establish the connection or start another ICLI server and try to connect to this new server.

ICLS1030E System error: The system call
pthread_detach() failed:
RC = *nnn*, errno = *nnn* (*xxx*), errno2 =
xxx
Message: *sss*.
The new thread could not be detached.
It executes
properly but may not return all
resources (such
as memory) upon exit.

Explanation: The system call pthread_detach() failed.
The new thread could not be detached. The new thread
works properly but might end up not returning all
resources (such as memory) upon exit.

System Action: Processing continues.

User Response: Try to remove the cause of the failure.
In case of memory shortage adjust your REGION size.
If the error recurs, save all available diagnostic
information and report the error to SAP/IBM.

ICLS1031E System error: System call
pthread_cond_wait() failed:
RC = *nnn*, errno = *nnn* (*xxx*), errno2 =
xxx
Message: *sss*.
The newly created thread may not
execute properly.

Explanation: The system call pthread_cond_wait()
failed. The new thread might not work properly.

System Action: Processing continues.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1032E System error in file *sss* and line *nnn*:
System call pthread_mutex_unlock()
failed:
RC = *nnn*, errno = *nnn* (*xxx*), errno2 =
xxx
Message: *sss*.
The newly created thread may not
execute properly.
Further connection attempts will fail.

Explanation: The system call pthread_mutex_unlock()
failed. The new thread might not work properly.
Further connection attempts will fail because the
semaphore could not be unlocked.

System Action: Processing continues. Further
connection attempts will fail.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1033E System error in file *sss* and line *nnn*:
System call setsockopt() for socket *nnn*
and option *nnn*
failed. RC = *nnn*,
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
The connection attempt failed. All other
ICLI threads
continue processing.
This message is immediately followed
by message ICLS1025,
which provides further information.

Explanation: This is a diagnostic message of the ICLI
server and is followed by message ICLS1025.

System Action: This connection attempt failed. All
other ICLI threads continue processing.

User Response: Determine the cause by inspecting
'Message' of the current message and the directly
following message ICLS1025. If the cause of failure
can't be determined: Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1034E System error: System call getsockname()
for socket *nnn* failed:
RC = *nnn*, errno = *nnn* (*xxx*), errno2 =
xxx
Message: *sss*.
The connection attempt failed. All other
ICLI threads
continue processing.
This message is followed by message
ICLS1025, which
provides further information.

Explanation: This is a diagnostic message of the ICLI
server and is followed by message ICLS1025.

System Action: This connection attempt failed. All
other ICLI threads continue processing.

User Response: Determine the cause by inspecting
'Message' of the current message and the directly
following message ICLS1025. If the cause of failure
cannot be determined: Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1035W TCP_KEEPALIVE option could not be
set for socket *nnn*:
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
The ICLI server will not be able to
exploit the TCP
Keep Alive function for the socket.

Explanation: The ICLI server wants to exploit the TCP
Keep Alive function of the specified socket. This
function is not available. The reason can be a missing
APAR PQ18618 or the usage of a non IBM TCP/IP
stack, e.g. Interlink.

System Action: The ICLI server continues with the system settings.

User Response: If you want to exploit the new function, install APAR PQ18618 and use the IBM TCP/IP stack. Use the environment variable ICLI_TCP_KEEPALIVE to set the Keep Alive value. For more information on ICLI_TCP_KEEPALIVE, see "ICLI server environment variables" on page 159 and the *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition*.

ICLS1036S   System error: System call *sss* **failed:**
            RC = *nnn*, **errno =** *nnn* **(***xxx***), errno2 =**
            xxx
            Message: *sss*.

Explanation: The system call <sss> failed. The error is described in 'Message'.

System Action: ICLI server is stopped.

User Response: Try to remove the cause of the error and restart the ICLI server. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1037S   Error: The log directory (*sss***) of the ICLI**
            server could not
            be accessed.
            The function call stat() to access the
            directory failed:
            RC = *nnn*, **errno =** *nnn* **(***xxx***), errno2 =**
            xxx
            Message: *sss*.
            This message will be redirected to
            DD:STDERR.

Explanation: The directory which was specified with the -LOGDIR command line option could not be accessed. The error is further described in 'Message'. This error message is written to DD:STDERR instead of the message file.

System Action: ICLI server is stopped.

User Response: Remove the cause for the error, for example by setting the access rights for the directory, and restart the ICLI server.

ICLS1038S   An invalid HFS directory (*sss***) was**
            specified for command line
            option -LOGDIR.

Explanation: The directory which was specified with the -LOGDIR command line option is not a valid HFS directory.

System Action: ICLI server is stopped.

User Response: Specify a valid HFS directory with the -LOGDIR command line option.

ICLS1039E   System error: The ICLI server could not
            open its message
            file *sss*.
            C-function open fails with RC = *nnn*
            errno = *nnn* **(***xxx***), errno2 =** *xxx*
            Message: *sss*.
            When this error occurs at startup, the
            ICLI server
            is stopped, otherwise, it continues
            processing and
            writes its messages to stderr.

Explanation: The ICLI message file of the ICLI server could not be opened. The error is further described in 'Message'.

System Action: ICLI server is stopped, when the error occurred at server startup. If it happens later, the ICLI server writes its messages to stderr.

User Response: Remove the cause for the error (for example by setting the access rights for the directory into which the message file is written) and restart the ICLI server.

ICLS1041S   Error: The service name '*sss***', which was**
            specified for command
            line option -SERVICENAME, is too
            long.
            Maximum allowed length is *nnn*.

Explanation: The service name that was specified with the option -SERVICENAME is too long.

System Action: ICLI server is stopped.

User Response: Specify a valid service name and restart the ICLI server. For information on how to select a servicename, refer to "ICLI server command line options" on page 101.

ICLS1042E   Error: The ICLI server could not accept
            the connection request.
            The newly created work thread failed in
            accept
            for socket *nnn* **RC =** *nnn*, **errno =** *nnn*
            (*xxx***), errno2 =** *xxx*
            Message: *sss*.
            See the preceding message for more
            information on the
            cause of the error.

Explanation: The accept processing in the newly created work thread failed. The connection between the ICLI client and server cannot be established. This message is preceded by another message which gives more details about the cause of the error.

System Action: This ICLI thread is stopped. All other threads continue processing.

User Response: Remove the cause of the failure and restart the ICLI server.

ICLS1043S **Error: The port number** *sss***, which was specified for command line option -PORT, is not within the valid range (1024 to 65535). The ICLI server is stopped.**

Explanation: The port number which was specified with the command line option -PORT was not in the valid range of port numbers.

System Action: ICLI server is stopped.

User Response: Specify a valid port number when starting the ICLI server with the -PORT command line option. For information on how to select an ICLI connection port, refer to "ICLI server command line options" on page 101.

ICLS1044S **Error: No service name was specified for the -SERVICENAME command line option. The ICLI server is stopped.**

Explanation: The command line option -SERVICENAME was not followed by a servicename parameter.

System Action: ICLI server is stopped.

User Response: Specify a valid servicename with the -SERVICENAME option. For information on how to select a servicename, refer to "ICLI server command line options" on page 101.

ICLS1045I **Information: The ICLI thread for port** *nnn* **has been terminated.**

Explanation: This message is issued whenever an ICLI work thread ends. The reason for this termination of the thread can either be a DISCONNECT request from the ICLI client or it may have been caused by a previous error.

System Action: The ICLI thread is terminated.

User Response: None.

ICLS1046W **Warning: The ICLI server environment variable ICLI_TCP_KEEPALIVE is not set or its value is out of range. ICLI thread processing continues using the default value** *nnn***.**

Explanation: The value specified for TCP_KEEPALIVE using the environment variable ICLI_TCP_KEEPALIVE is out of range. Values between 0 and 2,147,460 are supported.

System Action: The ICLI thread continues processing using the default value.

User Response: For more information on ICLI_TCP_KEEPALIVE, see "ICLI server environment variables" on page 159 and the *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition*. For more information on TCP_KEEPALIVE socket option semantics, read the description in APAR PQ18618.

ICLS1047S **Error: No directory was specified for command line option -LOGDIR. The ICLI server is stopped.**

Explanation: The command line option -LOGDIR was not followed by a log directory argument.

System Action: ICLI server is stopped.

User Response: Specify a valid log directory after the -LOGDIR option. For information on how to select a log directory, refer to "ICLI server command line options" on page 101.

ICLS1048E **System error: An attempt to create a socket of domain** *nnn* **type** *nnn* **protocol** *nnn* **failed: errno =** *nnn* **(***xxx***), errno2 =** *xxx* **Message:** *sss***. This ICLI server thread is stopped.**

Explanation: Socket could not be created for the reason indicated in 'Message'.

System Action: This ICLI server thread is stopped.

User Response: If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1049W **Warning: ICLI passticket verification is disabled. Processing continues. However, be aware that your environment may not be secure if the network between the ICLI server and clients is not protected by other means.**

Explanation: The ICLI passticket authorization mechanism was disabled by setting the environment variable ICLI_TRUSTED_CONNECTIONS. The ICLI client now does not need to do any passticket authorization when connecting to this ICLI server. On the ICLI client side the ICLI_TRUSTED_CONNECTIONS variable must be set for this purpose. Note: If you want a secure environment in which only authorized clients can access the ICLI server, you should have the environment variable unset!

System Action: Processing continues without passticket verification.

User Response: For details, see "Configuring ICLI PassTicket signon" on page 29 in this manual.

ICLS1050E   System error: No cleanup information was available upon thread exit. The cleanup could not be performed. All other threads continue processing.

Explanation: This error indicates that the cleanup structure was never filled with data we cannot do cleanup work.

System Action: The ICLI thread is stopped. All other threads continue.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1051S   Severe error: Main listening socket was closed in file *sss* and line *nnn*. The ICLI server will not accept any further connection requests. The ICLI control thread is stopped. Further messages will be issued indicating that ICLI work threads are exiting.

Explanation: The ICLI server can no longer accept new connection requests. When the ICLI server is being stopped, this is one of the first actions that are taken to prevent further connections from ICLI Clients. After this is done, you will see messages indicating the exit of ICLI work threads.

System Action: The ICLI control thread is stopped.

User Response: None.

ICLS1052W   Warning: errno *nnn* was received from TCP/IP. This error indicates that TCP/IP has been stopped. All existing connections will be dropped. Check your SYSLOG for TCP/IP-related messages. The ICLI server stays in retry mode until you restart TCP/IP or stop the ICLI server.

Explanation: The ICLI server received a return code that indicates TCP/IP is down. All existing connections are dropped. The ICLI server starts checking every 5 seconds if TCP/IP is up again by trying to create a socket. Check your z/OS SYSLOG for messages regarding TCP/IP. As soon as TCP/IP is up and running again the creation of the socket will succeed and the ICLI server will be able to accept new connections and continues normal operations. During this retry mode, the ICLI server can be stopped normally.

ICLS1053W   Warning: The environment variable *sss* has an invalid value (*sss*). Processing continues with passticket verification enabled.

Explanation: The environment variable <sss> was set to an invalid value.

System Action: Processing continues with passticket verification enabled.

User Response: Check your environment settings and specify valid values for this variable. For details see Appendix C, "ICLI environment variables," on page 157.

ICLS1055E   The signon attempt of the ICLI client was rejected. ICLI passticket verification failed : applidlen = *xxx* applid = *sss* passticket = *sss* useridlen = *xxx* userid = *sss* SAFRC (reg15) = *nnn* RACFRC = *nnn* RACFRS = *nnn* The newly created thread is terminated. All other threads continue processing.

Explanation: The signon attempt of the ICLI client was rejected. The message text contains the <application_name> and <user_ID> parameters from the signon file of the client. The SAFRC, RACFRC, and RACFRS return codes are described in *z/OS Security Server RACROUTE Macro Reference* in the section on the RACROUTE REQUEST=VERIFY" macro call.

System Action: The newly created ICLI thread is stopped. All other threads continue processing.

User Response: Make sure the values specified in the signon file on the client match the values specified in the RACF PTKTDATA class profile. If you do not need the additional security provided by passticket signon, you can disable it by setting the environment variable ICLI_TRUSTED_CONNECTIONS=1 for both ICLI client and server.
Special Case 1:
The values SAFRC=8 and RACFRC=28 are a special case not described in detail in *z/OS Security Server RACROUTE Macro Reference*. In the z/OS system log, you will find the RACF message: ICH70003I YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF RACF PASSWORD ATTEMPTS. This means that too many consecutive unsuccessful signon attempts have been performed and that the user ID has been revoked.
Special Case 1 User Response:
Ask your system administrator to reset the user ID from the revoked state.

Special Case 2:
If the signon attempt fails with application name "NOTHING" and user ID "DEFINED", the environment variable ICLI_TRUSTED_CONNECTIONS was set inconsistently (the ICLI server runs with and the ICLI client without security).
Special Case 2 User Response:
If RACF secured signon is desired, then issue 'unset ICLI_TRUSTED_CONNECTIONS' on the ICLI client. If security should be disabled, issue 'export ICLI_TRUSTED_CONNECTIONS=1' on the ICLI server. For details, see "Configuring ICLI PassTicket signon" on page 29 in this manual.

ICLS1056S   **Error: No DB2 subsystem name was specified for the command line option -CHECK. The ICLI server is stopped.**

Explanation: The command line option -CHECK was not followed by a DB2 subsystem name or DB2 group attachment name.

System Action: The ICLI server is stopped.

User Response: Specify a DB2 subsystem or a DB2 group attachment name after the -CHECK option. For information on how to use -CHECK, refer to "ICLI server command line options" on page 101.

ICLS1057I   **Information: Cleanup processing for the ICLI server main thread is complete. The main thread will exit. Those ICLI work threads that are currently processing in DB2 will exit when they return from DB2.**

Explanation: Cleanup processing for the ICLI server main thread ended, the main thread will exit. Other ICLI threads can still continue to process in DB2 but they will exit when they return from DB2.

System Action: ICLI server main thread is stopped.

User Response: None.

ICLS1058E   **Error: The DB2 subsystem name '*sss*', which was specified for command line option -CLEANIPC, is too long. It is truncated to *nnn* characters.**

Explanation: The DB2 subsystem name that was specified after the -CLEANIPC command line option was too long.

System Action: The DB2 subsystem name is truncated to the number of characters indicated in the message.

User Response: Make sure that you specify a valid DB2 subsystem name with the -CLEANIPC option. If the truncated DB2 subsystem name does not refer to a valid DB2 subsystem in your installation, the

semaphore and shared memory cannot be removed. You might have to restart the ICLI server with a correct DB2 subsystem name.

ICLS1059S   **Error: No DB2 subsystem name was specified for command line option -CLEANIPC.**

Explanation: The command line option -CLEANIPC was not followed by a DB2 subsystem name or DB2 group attachment name.

System Action: ICLI server is stopped. The semaphore and the shared memory are not removed.

User Response: Specify a DB2 subsystem name or a DB2 group attachment name after the -CLEANIPC option.

ICLS1060S   **Severe error: The command line option -CLEANIPC cannot be used in conjunction with other command line options.**

Explanation: The -CLEANIPC command line option is mutually exclusive with all other command line options.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server either specifying the -CLEANIPC command line option without any additional command line options or specifying the other command line options.

ICLS1061W   **Warning: Signal *sss* could not be deleted from the signal set for the ICLI server. The C-function sigdelset() failed: RC = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx* Message: *sss*. Processing continues. Problems are to be expected when the ICLI server is stopped.**

Explanation: Signals cannot be deleted from signal set for ICLI server. Expect problems when stopping the ICLI server!

System Action: Processing continues.

User Response: If the error recurs each time you start an ICLI server, save all available diagnostic information and report the error to SAP/IBM.

ICLS1062S   **Severe error: The ICLI server is not running APF-authorized. The ICLI server is stopped.**

Explanation: The reason for this error is that the ICLI server executable is not APF authorized by the extended attribute 'a'. The extended attribute gets lost

| e.g. if the normal cp command is used to copy the ICLI
| server executable. As a consequence the ICLI server
| does not run APF authorized. APF authorization is
| needed for the ICLI server to function properly.

| **System Action:** ICLI server is stopped.

| **User Response:** Check the ICLI server executable by
| the command ls -E. Check your SMP/E APPLY
| procedure for the ICLI PTF. If you need to copy the
| ICLI server executable, make sure that you set the
| extended attribute 'a' afterwards.

| **ICLS1063I** Information: ICLI shutdown processing
| found
| ICLI thread *xxx* active in DB2 subsystem
| *sss*.
| Its corresponding DB2 thread (*nnn*) will
| be cancelled.
| Message ICLS1125I (Thread cancelled)
| might be
| reported by the ICLI thread as a
| consequence.

| **Explanation:** A abrupt stop of the ICLI server was
| initiated but there was an ICLI work thread which was
| executing in DB2. In order to terminate this ICLI work
| thread, its corresponding DB2 thread needs to be
| cancelled. A subsequent message ICLS1125I reported by
| the work thread is a consequence of this cancel action
| and should not be considered an error.

| **System Action:** ICLI server will stop after all ICLI
| work threads have been terminated,

| **User Response:** None

ICLS1064E Error: The ICLI received an abend
condition *sss*:
Reason: *sss*
More info: *sss*
CEEDUMP title: *sss*.
The ICLI thread is stopped.

Explanation: The condition handler reports that an
abend condition was encountered. The abend may
either be caused by components external to the ICLI,
for example, DB2 and z/OS services, or by an internal
ICLI failure.

System Action: ICLI thread is stopped.

User Response: Examine the condition code and the
reason code and try to find the cause for the condition.
In case of DB2 abends (condition codes 04E and 04F)
check "DB2 Messages and Codes" for an explanation of
the reason code. In any case check SYS1.LOGREC for
any errors reported by z/OS components. If the error is
in an component external to the ICLI please report the
error through the appropriate IBM channels for this
component. If the cause of the error cannot be
determined or if it is an ICLI internal failure, save all
available diagnostic information including the

CEEDUMP and ICLI message files and report the error
to SAP/IBM.

ICLS1065I Information: An active DB2 thread (*nnn*)
was terminated
(ABEND 04E RC E50013) successfully
because of a reconnect.
This DB2 thread belongs to WP *nnn* for
host (ip=*sss*,port=*nnn*),
ICLI thread *xxx*, and connection ID *nnn*.

Explanation: An active DB2 thread was terminated
because of reconnect.

System Action: Because of reconnect the idle thread is
terminated.

User Response: None

ICLS1066E System error: The ICLI server
encountered an error while
it was waiting for console input (STOP
or MODIFY command).
As a result, it may not be possible to
stop the started task
of the ICLI server using the STOP
command.
The error occurred in __console() wait:
RC = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.

Explanation: An internal error occurred while the ICLI
server was waiting for console input (STOP or
MODIFY command). As a consequence stopping the
ICLI server-started task via the STOP command might
not be possible.

System Action: Processing continues.

User Response: Stop the ICLI server using MODIFY
with option TERM or the UNIX System Services kill
command. If the error occurs each time you start an
ICLI server, save all available diagnostic information
and report the error to SAP/IBM.

ICLS1067I Information: A STOP command was
issued from the console,
the ICLI server is closing down.

Explanation: The started task of the ICLI server was
stopped via a console STOP command.

System Action: ICLI server will exit.

User Response: None.

ICLS1068I Information: ICLI_MSGLEVEL was set
from *c* to *c*.
The ICLI server will exploit the new
message level.

Explanation: New message level for the ICLI server
was specified.

System Action: ICLI exploits new message level.

User Response: None.

ICLS1069I   **Information: ICLI_COM_TRACE will be set to** *nnn***.**
If set to 1, the communication trace is switched on
and a communication trace will be available.
Otherwise, it is switched off.

Explanation: The communication trace is switched 0 = off or 1 = on.

System Action: If the communication trace is switched on, there will be a communication trace available.

User Response: None.

ICLS1070I   **Information: ICLI_TRACE_LEVEL will be set from** *nnn* **to** *nnn***.**
The ICLI server will exploit the new trace level.

Explanation: New trace level for the ICLI server was specified.

System Action: ICLI exploits new trace level.

User Response: None.

ICLS1074S   **Error: An invalid value was specified for command line**
option -MFS (message file size).
Valid values are numbers greater than 0 and -1 for the
maximum (which is specified in the BPXPRMxx parmlib).
The ICLI server is stopped.

Explanation: The command line option -MFS, which specifies the size of the message file (in megabytes) was followed by an invalid number.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server and use a valid number for -MFS. For -MFS are valid: Numbers greater 0 and -1 for the maximum (which is specified in the BPXPRMxx parmlib). For information on how to select the message file size, refer to "ICLI server command line options" on page 101.

ICLS1075S   **Error: An invalid value was specified for the command line**
option -MFN (number of message files to be used).
Valid values are numbers greater than 0 and -1 for the
maximum (which is specified in the BPXPRMxx parmlib).
The ICLI server is stopped.

Explanation: The command line option -MFN, which specifies the maximum number of the message files was followed by an invalid number.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server and use a valid number for -MFN. For -MFN are valid: Numbers greater 0 and -1 for the maximum (which is specified in the BPXPRMxx parmlib). For information on how to select the message file number, refer to "ICLI server command line options" on page 101.

ICLS1076S   **Error: An invalid combination of values was specified for the**
command line options -MFS (message file size) and
-MFN (number of message files).
The ICLI server is stopped.

Explanation: The command line option -MFS and -MFN, which specifies the maximum size and the maximum number of the message files have the invalid combination (-MFS = -1 and -MFN not 1). If you use for -MFS -1, which means using the maximum file size, it is not possible to have more than one message file.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server and don't use an invalid combination of -MFS and -MFN. If you set -MFS = -1, you have to set -MFN = 1.

ICLS1078W   **Warning: The attempt to make the ICLI server non-swappable**
failed. errno = *nnn* **(***xxx***), errno2 =** *xxx*
Message: *sss*
The ICLI server address space remains swappable.
This may result in a slight performance degradation.

Explanation: For performance reasons, the ICLI Server tries to make itself non-swappable. If this attempt fails, the ICLI Server will to run swappable and there can be a slight performance degradation.

System Action: Processing continues, but the ICLI Server address space remains swappable.

User Response: If you want to enable the ICLI Server to make its memory non-swappable, give the user ID that is used to start the ICLI Server read permission to the FACILITY class profile BPX.STOR.SWAP. See *z/OS UNIX System Services Planning* for details.

ICLS1079E   **System error: The attempt to work with the ICLI message**
file failed. The ICLI server continues processing
and writes all further messages to stderr.
C-function lseek fails with RC = *nnn***,**

errno = *nnn* (*xxx*), **errno2** = *xxx*
Message: *sss*.

Explanation: System error. It was not possible to set the cursor to the beginning of the ICLI message file. lseek() terminates with a bad return code.

System Action: The ICLI server continues processing. The ICLI server writes all further messages to stderr.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1080E **System error: The attempt to work with the ICLI**
server message file failed.
The ICLI server continues processing and writes
all further messages to stderr.
C-function lseek fails with RC = *nnn*,
errno = *nnn* (*xxx*), **errno2 =** *xxx*
Message: *sss*.

Explanation: It was not possible to change the cursor position in the current message file to overwrite the -----EOF----- string. lseek() terminates with a bad return code.

System Action: The ICLI server continues processing. The ICLI server writes all further messages to stderr.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1081E **Internal error: The size of the message exceeds**
the size of the message file. All further messages
will be written to stderr.
msg_file size = *nnn*, **data_length =** *nnn*,
msg_fd = *nnn*.
Additional info:
errno = *nnn* (*xxx*), **errno2 =** *xxx*
Message: *sss*.

Explanation: Internal error. The message (data_length) was larger than the message file size (msg_file_size).

System Action: The ICLI server continues processing. The ICLI server writes all further messages to stderr.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1082W **System warning: It was not possible to remove the**
-----EOF----- string at the end of the last message
file. The ICLI server will write all further
messages to the new message file.
The C-function lseek fails.
The full message file descriptor is *nnn*

errno = *nnn* (*xxx*), **errno2** = *xxx*
Message: *sss*.

Explanation: It was not possible to remove the -----EOF----- string at the end of the last message file. The problem is to change the cursor position in the last message file. lseek() terminates with a bad return code. Now you can see more than one file with the -----EOF----- string.

System Action: The ICLI server continues processing. The ICLI server writes to the new message file.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1083W **System warning: It was not possible to remove the**
-----EOF----- string at the end of the last message
file. The ICLI server will write all further
messages to the new message file.
The C-function write fails.
The full message file descriptor is *nnn*.
errno = *nnn* (*xxx*), **errno2 =** *xxx*
Message: *sss*.

Explanation: It was not possible to remove the -----EOF----- string at the end of the last message file. The problem is to write to the last message file. write() terminates with a bad return code. Now you can see more than one file with the -----EOF----- string.

System Action: The ICLI server continues processing. The ICLI server writes to the new message file.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1084S **Error: An invalid message file size (*nnn*) was**
specified for the -MFS command line option.
The specified value is greater than that of the
MAXFILESIZE parameter (*nnn*) in the BPXPRMxx parmlib.

Explanation: The message file size, set via -MFS is higher than the MAXFILESIZE parameter, specified in the BPXPRMxx parmlib.

System Action: The ICLI server is terminated.

User Response: Restart the ICLI server using a smaller value for the message file size, or change the MAXFILESIZE parameter in the BPXPRMxx parmlib.

ICLS1085S **Error: An invalid message file number (*nnn*) was**
specified for the -MFN option.
The value is greater than that of the MAXFILEPROC
parameter (*nnn*) in the BPXPRMxx parmlib.

Explanation: The message file number, set via -MFN is higher than the MAXFILEPROC parameter, specified in the BPXPRMxx parmlib.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server using a smaller value for the message file number, or change the MAXFILESIZE parameter in the BPXPRMxx parmlib.

ICLS1086E **System error: File *sss* in file *sss* and line *nnn* could not**
be opened. The request to open the file was initiated
by SAP transaction ST04 while trying to retrieve
ICLI server-related files from the host. The files
could not be retrieved. The C-function fopen fails,
the ICLI server continues processing. Additional info:
errno = *nnn* (*xxx*), errno2 = *xxx*

Explanation: This message is issued when the ICLI server is not able to open the specified file. The initiator of the file open is the SAP transaction ST04 retrieving ICLI server related files from the host.

System Action: The ICLI server continues processing. The SAP user cannot retrieve the requested file.

User Response: Please check errno and errno2 and, if possible, alter file permissions. If you are unsure about the reason for this problem, collect all information and send it to SAP/IBM.

ICLS1087W **System warning: File *sss* could not be closed.**
The request to close the file was initiated by SAP
transaction ST04 while retrieving ICLI server-related
files from the host. The C-function fclose fails,
the ICLI server continues processing. Additional info:
errno = *nnn* (*xxx*), errno2 = *xxx*

Explanation: This message occurs if the ICLI server was not able to close the specified file. The initiator of the file close is the SAP transaction st04 retrieving ICLI server related files from the host.

System Action: The ICLI server continues processing.

User Response: Check errno and errno2 for reason and correct the error. If you are unsure about the reason for this problem, collect all information and send it to SAP/IBM.

ICLS1088W **Internal warning: The ICLI server received an unknown**
request type. The download request was initiated
by SAP transaction ST04 while retrieving information
(files or a directory) from the host. Received request type: *nnn*.

Explanation: This message occurs if the ICLI server received an unknown request type. The request type determines what kind of information (directory or file) the client wants to retrieve from the host.

System Action: The ICLI server continues processing.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1089E **System error: The ICLI server could not open the directory for**
downloading files. The download request was initiated
by SAP transaction ST04. The C-function opendir fails,
the ICLI server continues processing. Errno: *nnn*

Explanation: This message occurs if the ICLI server cannot open the directory for the file download request. (Initiated via SAP transaction st04.)

System Action: The ICLI server continues processing.

User Response: If this error occurs repeatedly, save all available diagnostic information and report the error to SAP/IBM.

ICLS1090S **Error: An invalid message file name identifier (*sss*) was**
specified for command line option -MFNAME.
The file name is too long. Maximum allowed length
is 64. The ICLI server is stopped.

Explanation: The message file identifier that was specified after the -MFNAME command line option was too long.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server, specifying a valid file name identifier or omit the parameter -MFNAME.

ICLS1091S   **Error: No file name was specified for command line option -MFNAME. The ICLI server is stopped.**

Explanation: The file name is missing after the -MFNAME command line option.

System Action: ICLI server is stopped.

User Response: Restart the ICLI server, specifying a valid file name identifier or omit the parameter -MFNAME.

ICLS1092I   **Information: The ICLI server received a MODIFY command from the console with message *sss*.**

Explanation: ICLI server received via console a MODIFY command.

System Action: The system action depends on the received message. If there are unknown commands, message ICLS1093I will follow.

User Response: None.

ICLS1093W   **Warning: The ICLI server received a MODIFY command with an unknown or invalid command in message *sss*.**

Explanation: ICLI server received a MODIFY command from the console. The message contained an incorrect or unknown command.

System Action: ICLI server will continue.

User Response: Check the message you entered with the console command MODIFY.

ICLS1094W   **Warning: The message of the MODIFY command received from the console contained an unknown option (*sss*).**

Explanation: ICLI server received via console a MODIFY command with an incorrect or unknown option.

System Action: ICLI server will continue.

User Response: Check and correct your message given with console command MODIFY.

ICLS1095W   **Warning: The MODIFY command received from the console contained an unknown parameter (*sss*) for option=*sss*.**

Explanation: ICLI server received via console a MODIFY command with an incorrect or unknown parameter for the specified option.

System Action: ICLI server will continue.

User Response: Check and correct your message given with console command MODIFY.

ICLS1096I   **The ICLI server is stopping. The server will try to stop smoothly for *nnn* seconds.**

Explanation: Stop can need the given time plus the time which is needed to stop (rollback) all actions.

System Action: ICLI server will try x seconds to stop smoothly.

User Response: Wait the given time.

ICLS1097I   **The smooth stop of the ICLI server completed successfully. All work processes (threads) were stopped smoothly.**

Explanation: The ICLI server was stopped smoothly.

System Action: ICLI server is stopped.

User Response: None.

ICLS1098I   **The time limit defined for a smooth stop of the ICLI server has expired. Some work processes (threads) are still active. Further ICLI server actions depend on the stop mode chosen: The ICLI server will either stop immediately or continues processing until all work processes (threads) have stopped.**

Explanation: The ICLI server unable to stop smoothly within the time period defined for this stop mode.

System Action: If MODE(NOFORCE) or ICLI_STOP_MODE=2 was defined: The ICLI server continues until all work processes finish smooth. If MODE(FORCE) or ICLI_STOP_MODE=1 for the stop: The ICLI server and all remaining active work processes stop.

User Response: None.

ICLS1099E   **Internal error: The preparations for a later stop smooth could not be made. The ICLI server continues processing, but problems may have to be expected when a smooth stop of the ICLI server is initiated.**

Explanation: An internal error was detected during the preparation for a stop smooth.

System Action: ICLI server continues processing.

There can be problems in stopping the ICLI server smoothly.

User Response: None. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1100W **System warning: Signals to interrupt the ICLI's main thread**
could not be blocked.
The C-function sigprocmask() failed:
RC = *nnn*, **errno =** *nnn* (*xxx*), **errno2 =** *xxx*
Message: *sss*.

Explanation: Signals cannot be blocked to interrupt the main thread.

System Action: Processing continues processing.

User Response: If the error occurs each time you start an ICLI server, save all available diagnostic information and report the error to SAP/IBM.

ICLS1101W **System warning: File status information**
for file *sss*: *sss*
could not be read.
The C-function stat fails:
(errno = *nnn* (*xxx*)).

Explanation: Received an error code when reading status information of the file. Possible reasons: - The process does not have search permission on some component of the pathname prefix. - A loop exists in symbolic links encountered during resolution of the pathname argument.

User Response: Set the access permissions for the ICLI directory in USS correctly or fix the symbolic link loop.

| **ICLS1102W** **Error: The ICLI server received no or an**
| **invalid thread ID,**
| **thread id =** *xxx*
| **for file** *sss*, **line** *nnn*
| **Processing continues. No threads will be**
| **canceled.**

| **Explanation:** Received an invalid thread id.

| **System Action:** Processing continues. No thread will
| be canceled.

| **User Response:** Get the right thread ID for the cancel
| request.

ICLS1103E **System error: The ICLI server could not**
obtain memory,
possibly due to memory shortage.
Processing
continues, but no health buffer can be
displayed.
C-function malloc for *sss* **failed for file**
sss, **line** *nnn*

Explanation: Received an error code when calling c-function malloc to get memory.

System Action: For the moment, processing continues. No health buffer can be shown.

User Response: Check your system for a memory problem.

ICLS1105I **Registration with Automatic Restart**
Manager (ARM)
for element name <*sss*> **was successful:**
return code = *nnn*, **reason code =** *nnn*
(HEX: *xxx*)
You can now exploit ARM mechanisms
to restart the ICLI
server's started task.

Explanation: The ICLI server registered successfully with Automatic Restart Manager with the element name reported in the message. You can now employ ARM mechanisms to restart the ICLI started task. Refer to the *z/OS MVS Programming: Sysplex Services Guide* for details. Refer to the *z/OS MVS Programming: Sysplex Services Reference* for a detailed explanation of the return and reason codes of the IXCARM macro.

System Action: Processing continues.

User Response: None.

ICLS1106W **Registration with Automatic Restart**
Manager (ARM)
for element name <*sss*> **failed:**
return code = *nnn*, **reason code =** *nnn*
(HEX: *xxx*)
You will not be able to exploit ARM
mechanisms to
restart the ICLI server.

Explanation: The ICLI server failed to register with Automatic Restart Manager with the element name reported in the message using macro IXCARM. You will not be able to employ ARM mechanisms to restart the ICLI. Refer to the *z/OS MVS Programming: Sysplex Services Reference* for a detailed explanation of the return and reason codes of the IXCARM macro.

System Action: Processing continues.

User Response: If you want to exploit ARM to restart the ICLI, remove the cause for failure of the IXCARM macro and restart the ICLI server.

ICLS1107W **READYing the element with Automatic**
Restart Manager (ARM)
failed for element name <*sss*>.
return code = *nnn*, **reason code =** *nnn*
(HEX: *xxx*)
You will not be able to exploit ARM
mechanisms to
restart the ICLI server.

Explanation: The ICLI server failed to set the element

to READY with Automatic Restart Manager using macro IXCARM. You will not be able to employ ARM mechanisms to restart the ICLI. Refer to the *z/OS MVS Programming: Sysplex Services Reference* for a detailed explanation of the return and reason codes of the IXCARM macro.

System Action: Processing continues.

User Response: If you want to exploit ARM to restart the ICLI, remove the cause for failure of the IXCARM macro and restart the ICLI server.

**ICLS1108W   Deregistering with Automatic Restart Manager (ARM)
failed with return code = *nnn*, reason code = *nnn* (HEX: *xxx*)
You will not be able to exploit ARM mechanisms to restart
the ICLI server.**

Explanation: The ICLI server failed to deregister with Automatic Restart Manager using macro IXCARM. You will not be able to employ ARM mechanisms to restart the ICLI. Refer to the *z/OS MVS Programming: Sysplex Services Reference* for a detailed explanation of the return and reason codes of the IXCARM macro.

System Action: Processing continues.

User Response: If you want to exploit ARM to restart the ICLI, remove the cause for failure of the IXCARM macro and restart the ICLI server.

**ICLS1109W   An attempt to set IBM socket option SO_EioIfNewTP
failed in line *nnn* and file *sss*.
Ignore this warning if you are running the
Integrated AF_INET PFS. If you are running
the Common AF_INET PFS and have defined
several TCP/IP stacks, the ICLI server will not be
informed when a TCP/IP stack is restarted and cannot
react accordingly, i.e., it cannot accept connections
via the restarted stack.
Error code: *nnn* / '*sss*'.
errno2 returned: *xxx*.**

Explanation: The ICLI server could not set IBM socket option SO_EioIfNewTP. Under a Common AF_INET PFS the ICLI server needs to set this socket option in order to get informed when a TCP/IP stack is (re)started after the ICLI server was started. If you are using Integrated Sockets AF_INET PFS then you can ignore the warning message since it has no effect on the operation of the ICLI server. For details on the Integrated and Common AF_INET PFS, see *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th*

Edition (chapter "TCP/IP support", section "Defining the BPXPRMxx parmlib member").

**ICLS1111W   Environment variable (*sss*) has an invalid value ('*sss*').
The default message level ('*sss*') will be used.**

Explanation: The value that was specified for the ICLI message level environment variable is invalid.

System Action: Processing continues with the default message level setting.

User Response: Check your environment settings and specify valid values for this variable. For details, see Appendix C, "ICLI environment variables," on page 157.

**ICLS1115E   System error: The requested thread could not be canceled
properly. The ICLI server and all other threads
continue processing. The system call 'shutdown'
failed with rc =*nnn* and errno=*nnn* (*xxx*), errno2 = *xxx*.
For thread id =*xxx*, socket = *nnn* in file *sss*, line *nnn***

Explanation: The ICLI server was not able to cancel the requested thread properly.

System Action: Processing of the ICLI server and all other threads continues.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

**ICLS1121W   Environment variable ICLI_COM_TRACE has an invalid value (*sss*).
No communications trace will be generated.**

Explanation: The value that was specified for the ICLI_COM_TRACE environment variable is invalid.

System Action: Processing continues as if the environment variable were unset. No communications trace will be generated.

User Response: Check your environment settings and specify a valid value for this variable. For details, see Appendix C, "ICLI environment variables," on page 157.

**ICLS1123S   Severe error: Environment variable *sss* has an invalid
value ('*sss*'). This value cannot be interpreted as
a list of DB2 SQL codes.**

Explanation: The value that was specified for the environment variable is invalid.

System Action: ICLI server terminates.

User Response: Check your environment settings and specify a valid value for this variable. For details, see Appendix C, "ICLI environment variables," on page 157.

ICLS1124S **Severe error: Environment variable ICLI_BIND_IPADDR has an invalid value (<*sss*>). The value does not represent a valid IP address.**

Explanation: The value that was specified for the environment variable is invalid, because it does not represent a valid host address expressed in standard dotted-decimal notation (for example: "10.0.0.1").

System Action: The ICLI server terminates.

User Response: Check your environment settings and specify a valid value for this variable. For details, see Appendix C, "ICLI environment variables," on page 157.

| **ICLS1125I** **This thread was cancelled.**

| **Explanation:** The thread was cancelled in DB2 using command -CANCEL or stopping DB2 with the FORCE option. If there is a preceding ICLS1063I message for this thread in the ICLI message log, the -CANCEL was issued by the ICLI to cancel the old work thread after it was restarted, or during an ICLI server shutdown.

| **System Action:** The ICLI server work thread terminates.

| **User Response:** None.

ICLS1190S **Severe system error: A semaphore that is used for ICLI thread management during setup could not be initialized. The system call pthread_mutex_init() failed: R/C = *nnn*, errno = *nnn* (*xxx*), errno2 = *xxx* Message: *sss*.**

Explanation: This is a system error. A semaphore could not be initialized, which is used for ICLI thread management during setup. The reason for the failure is given in 'Message'.

System Action: The ICLI server terminates.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1191W **System error: A semaphore that is used for ICLI thread management could not be destroyed during an attempt to terminate the ICLI server. The resource will be freed by the operating system when the server process terminates. The system call pthread_mutex_destroy() failed: R/C = *nnn*, errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss*.**

Explanation: This is a system error. A semaphore could not be destroyed during termination of ICLI server, which is used for ICLI thread management. The reason for the failure is given in 'Message'.

System Action: ICLI server cannot free this resource properly. It will be freed by the Operating System if the server process terminates.

User Response: If the ICLI server did not terminate, cancel the ICLI server. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1192E **System Error: The attempt to add a new element to the ICLI thread queue failed, because not enough memory could be obtained. The system call calloc(1, *nnn*) failed: R/C = *ppp*, errno = *nnn* (*xxx*) Local socket : fd=*nnn* *sss*(*nnn*) = *xxx*(*xxx*) Client socket: *sss*(*nnn*) = *xxx*(*xxx*) This ICLI server thread will be stopped. Existing ICLI server threads continue processing. Further problems related to memory shortage are to be expected.**

Explanation: Because of memory problems, a new element could not be added to the ICLI thread queue. As a consequence, this ICLI server thread will be closed. You should expect further problems related to memory shortage.

System Action: This ICLI server thread will be stopped. Processing of existing ICLI ICLI server threads continues.

User Response: Increase the memory available to the ICLI server address space, for instance by increasing the region size.

ICLS1193W **The ICLI server thread that accepts connection requests from ICLI clients could not be canceled during an attempt to terminate the ICLI server.**

In line *nnn* and file *sss* system call
pthread_cancel() R/C = *nnn*,
errno = *nnn* (*xxx*), errno2 = *xxx*.
Message: *sss*

Explanation: This is a system error. The thread which accepts connection requests from ICLI clients cannot be stopped during termination of ICLI server processing. The reason for the failure is given in 'Message'.

System Action: The ICLI server cannot free the thread resources properly. They will be freed by the Operating System if the server process terminates.

User Response: If the ICLI server did not terminate, stop the ICLI server. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1220W **System error: Tag information for the ICLI server**
thread could not be set. The tag cannot be displayed
using console command DISPLAY OMVS,PID=nnnnn.
In line *nnn* and file *sss* system call
pthread_tag_np() R/C = *nnn*.
New tag string is:
<*sss*>
Old tag string is:
<*sss*>
errno = *nnn* (*xxx*), errno2 = *xxx*,
Message: *sss*

Explanation: This is a system error. The system call pthread_tag_np() failed, which is used to set the tag info for the ICLI server thread. Further explanation is given in 'Message'.

System Action: Processing continues. The tag is not available for display with the DISPLAY OMVS,PID=nnnnn console command.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1225W **System error: Signals could not be blocked to interrupt**
the ICLI closeall() function.
System call sigprocmask() failed: R/C = *nnn*
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.

Explanation: This is a system error. Signals cannot be blocked to interrupt the ICLI closeall() function.

System Action: Processing continues.

User Response: If the error recurs each time you stop an ICLI server, save all available diagnostic information and report the error to SAP/IBM.

ICLS1226W **System error: The local socket could not be closed when**
the ICLI server was being stopped (closeall()).
Depending on the cause of the failure, the ICLI server
may not terminate properly.
System call 'shutdown' (*nnn*) failed.
R/C = *nnn*, errno = *nnn* (*xxx*), errno2 = *xxx*,
Message: *sss*
Local socket : *sss*(*nnn*) = *xxx*(*xxx*)
Client socket: *sss*(*nnn*) = *xxx*(*xxx*)

Explanation: This is a system error. The local socket could not be closed during termination processing of the ICLI server. The reason for the failure is given in 'Message'.

System Action: The ICLI server may not terminate properly depending on the reason for the failure.

User Response: If the ICLI server did not terminate, cancel the ICLI server. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1231E **System error: The communications trace file *sss* in the**
log directory could not be opened. ICLI server
processing continues, but communications tracing
is disabled. System call open() failed:
R/C = *nnn*, errno = *nnn* (*xxx*)
errno2 = *xxx*

Explanation: The communications trace file <sss> could not be opened in the log directory. Check the errno and errno2 values for the reason.

System Action: ICLI server continues but communications tracing is disabled.

User Response: Resolve the problem by increasing the file system size or changing to another log directory.

ICLS1232E **System error: The attempt to write a trace entry**
to the communications trace file failed. Check whether there is enough space left in the
file system. ICLI server processing continues,
but communications tracing is currently disabled.

Explanation: The ICLI server tried unsuccessfully to write a trace entry into the communications trace file. Check if there is enough space left on the file system.

System Action: ICLI server continues but communications tracing is disabled.

User Response: Resolve the problem by increasing the

file system size or changing to another log directory and restart the ICLI server.

ICLS1300I The following ICLI client has successfully connected
to the DB2 system:
ICLI client hostname: *sss*
ICLI client ip: *sss*
ICLI client for SAP Version: *sss*
ICLI client protocol version: *nnn*
ICLI client internal version: *nnn*
ICLI client time: *sss*
ICLI client port = *nnn*, process ID = *nnn*,
and connection ID = *nnn*.
Workprocess class = *nnn* (*sss*)
SAPSYSTEM = *sss* SAP Workprocess
Number = *nnn*
connected to database attach name <*sss*>
Plan <*sss*>.

Explanation: An ICLI client connected successfully to the DB2 subsystem. The 'Workprocess ID' and 'workprocess class' allow you to correlate client process IDs with ICLI server threads.

System Action: Processing continues.

User Response: None.

ICLS1301E Interface error in line *nnn* and file *sss*:
The statement ID *nnn* is too large. It exceeds the
currently permitted maximum size for the statement
cache (*nnn*). The SQL statement sent by the ICLI
client could not be processed.

Explanation: This is an internal DBSL or ICLI interface error. The ICLI client tried to use statement IDs that exceeded the maximum size currently permitted.

System Action: The SQL statement sent by the ICLI client could not be processed.

User Response: Save the message file and any further information, for example, trace files. Report the error to SAP/IBM.

ICLS1302E Interface error: The PREPARE statement string sent by the
client is too long (*nnn*). It exceeds the currently
permitted maximum statement length (*nnn*).
The SQL statement sent by the client could not be
processed.

Explanation: This is an internal error in the DBSL or ICLI interface. The ICLI client tried to send a statement for PREPARE whose statement length exceeds the current maximum.

System Action: The SQL statement sent by the ICLI client could not be processed.

User Response: Save the message file and any further information, for example trace files. Report the error to SAP/IBM.

ICLS1303E System error: The attempt to obtain time via system call
time() failed.
R/C = *nnn*, errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
The passticket verification will not work properly
and the ICLI client will not be able to connect to
the ICLI server.

Explanation: System error. The time() system function returned with an error. The reason for the failure is given in 'Message'.

System Action: The SAF passticket verification will not work properly and the ICLI client will be unable to connect. Circumvention: Set the environment variable ICLI_TRUSTED_CONNECTIONS to 1 on both the client and server sides.

User Response: Save the message file and any further information, for example trace files. Report the error to SAP/IBM.

ICLS1304E Error: ICLI client connecting from *sss* caused a
version mismatch:
ICLI server SAP Version: *nnn* (*sss*)
ICLI server protocol version: *nnn*
ICLI server internal version: *nnn*
ICLI client SAP Version: *nnn* (*sss*)
ICLI client protocol version: *nnn*
ICLI client internal version: *nnn*
The ICLI thread is stopped. The ICLI client cannot
connect to this ICLI server.

Explanation: An ICLI client tried to connect with a backlevel ICLI version. The SAP Version must always be identical between ICLI client and server. The protocol version must also match between ICLI client and server. The only difference that can exist is in the internal version.

System Action: The ICLI thread is stopped. The ICLI client is not allowed to access this ICLI server.

User Response: Ensure that the correct ICLI client version is installed on the client machine.

**ICLS1305E   Error: The ICLI client could not connect to the
DB2 subsystem, because the DB2 subsystem name defined
in the profile or environment variable on the client
side is too long. Specified length is *nnn*,
maximum allowed length is *nnn*.**

Explanation: The DB2 subsystem name which was specified on the client side via profile or environment variable is too long. The currently allowed maximum is 4 characters.

System Action: This ICLI thread cannot connect to the DB2 subsystem.

User Response: Correct the specification of the DB2 subsystem name on the client side.

**ICLS1306E   Internal error: The ICLI thread could
not connect to the
DB2 subsystem, because the DB2 subsystem name defined
in the profile or environment variable on the client
side is too short. Specified length is *nnn*, minimum
allowed length is 1 character.**

Explanation: This is an internal protocol error. The length of the DB2 subsystem name which was specified on the client side via profile or environment variable is too short. The minimum length is 1 character.

System Action: ICLI thread cannot connect to the DB2 subsystem.

User Response: Save the message file and any further information, for example trace files. Report the error to SAP/IBM.

**ICLS1317E   Interface error: The string of the Execute
Immediate
statement sent by the client is too long.
Length of
the statement is *nnn* characters.
DB2 SQL statements must not exceed
nnn characters.
The SQL statement could not be
executed. SQL code
-101 will be passed back to the ICLI
client.**

Explanation: This is an internal DBSL or ICLI interface error. The ICLI client sent a statement for EXECUTE IMMEDIATE whose statement length exceeds the current DB2 maximum statement length.

System Action: The SQL statement sent by the ICLI client could not be processed. An SQL code of -101 is passed back to the ICLI client. Processing continues.

User Response: Save the message file and any further

information, for example trace files. Report the error to SAP/IBM.

**ICLS1350W   The ICLI server was unable to connect
to the Workload
Manager (WLM). The connect to WLM failed with
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*
The ICLI server continues processing without WLM
enablement.**

Explanation: The ICLI server was unable to connect to the WLM workload manager. A possible reason is that the user starting the ICLI server is not authorized to access the RACF BPX.WLMSERVER profile.

System Action: The ICLI server continues without attaching to WLM.

User Response: Please make sure that you have defined the BPX.WLMSERVER profile within RACF and that the user ID used to start the ICLI server has sufficient access to this profile. See also Chapter 2, "z/OS security, group IDs and user IDs," on page 23 for a detailed description. If the reason for the failed WLM connect attempt is not given by insufficient user rights, please discuss it with your WLM administrator and report the error to SAP/IBM.

**ICLS1351E   System error: A Workload Manager
(WLM) server classify
handle could not be created:
errno = *nnn* (*xxx*), errno2 = *xxx*.
Message: *sss*
This thread runs, but it is not
WLM-enabled.**

Explanation: The ICLI server couldn't create a server classify handle.

System Action: This thread runs not WLM enabled. Processing continues.

User Response: Check whether enough storage is available to the ICLI server. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

**ICLS1352E   System error: Server classify area field
sss could not be set.
errno = *nnn* (*xxx*), errno2 = *xxx*.
Message: *sss*
This thread runs, but it is not
WLM-enabled.**

Explanation: The ICLI server couldn't set a field for a server classify handle.

System Action: This thread runs not WLM enabled. Processing continues.

User Response: check that the ICLI server has enough

storage left. If the problem persists save all available diagnostic information and report the error to SAP/IBM.

ICLS1353E System error: An enclave for work process type *sss* could not be created. errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss* This thread runs, but it is not WLM-enabled.

Explanation: The creation of the enclave specified in the error message failed.

System Action: Processing continues.

User Response: Examine the return and reason code and discuss it with your system WLM administrator. If the problem persists, save all available diagnostic information and report the error to SAP/IBM.

ICLS1355W The deletion of enclave '*sss*' failed with errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss* The ICLI server terminates without deleting the enclave.

Explanation: The deletion of the enclave specified in the error message failed. The ICLI server terminates without deleting this enclave.

System Action: Processing continues, ICLI server terminates. The ICLI server could not free this resource properly. It will be freed by the Operating System after server process termination.

User Response: Examine the return and reason code and discuss it with your system / WLM administrator.

ICLS1356W An attempt to disconnect from the Workload Manager (WLM) failed. errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss* The ICLI server terminates.

Explanation: Before the ICLI terminates, it disconnects from WLM. This disconnect has failed with the return and reason codes shown in the messages.

System Action: Processing continues, ICLI server terminates.

User Response: Examine the return and reason codes and discuss the problem with your system / WLM administrator.

ICLS1359W The thread could not leave enclave *sss*. errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss* Processing continues. The thread still

resides in the enclave.

Explanation: This thread couldn't leave the specified enclave.

System Action: Processing continues. This thread still resides in the enclave.

User Response: Examine the return and reason code and discuss it with your system / WLM administrator.

ICLS1360W The thread could not join enclave *sss*: errno = *nnn* (*xxx*), errno2 = *xxx*. Message: *sss* This thread will not run WLM-enabled. It will be scheduled according to the address space settings.

Explanation: This thread couldn't join the specified enclave.

System Action: Processing continues. This thread will be scheduled according to the address space settings.

User Response: Examine the return and reason code and discuss it with your system / WLM administrator. Furthermore save all available diagnostic information and report the warning to SAP/IBM.

ICLS1361W The user ID for starting the ICLI server could not be obtained from the environment. The user ID ICLIRUN is used instead.

Explanation: The user ID which started the ICLI server cannot be obtained from the environment. The user ID is necessary to classify WLM enclaves.

System Action: The default user ID 'ICLIRUN' is used instead.

User Response: Inform SAP/IBM. As a circumvention, enable the user ID to RACF and change your WLM classification rules accordingly.

ICLS1362I The ICLI server has successfully connected to the Workload Manager (WLM). All ICLI work threads are running WLM-controlled.

Explanation: The ICLI server has successfully connected to the WLM workload manager.

System Action: None.

User Response: Make sure that all the WLM definitions mentioned in Chapter 2, "z/OS security, group IDs and user IDs," on page 23 and "Workload Management (WLM)" on page 36 have been made.

ICLS1363I   An attempt to connect to the Workload Manager (WLM) failed because of a previous error.

Explanation: The ICLI server was unable to connect to the WLM workload manager. Check the preceding error.

System Action: The ICLI server continues without attaching to WLM.

User Response: Please resolve the problem reported error in the previous error.

ICLS1364I   The thread successfully joined the WLM enclave with following attributes: DB2 attach name (PK) = *sss*, Host name (SPM) = *sss*, SAP system number (PR) = *sss*, Work classification type (TN) = *sss*, User ID (UI) = *sss*.

Explanation: This thread successfully joined the specified enclave. For each attribute of the enclave the corresponding WLM work qualifier is shown in parentheses.

System Action: Processing continues. This thread will be scheduled according to the classification rules of the actual WLM policy matched by the attributes of this enclave.

User Response: None.

ICLS1400W   An SQL error occurred. This is the corresponding DB2 SQL error message:
sss
sss
sss
sss
sss
sss
sss
sss
sss
sss
The SQL error is reported to the SAP application, which should handle it appropriately.

Explanation: An SQL error occurred. The corresponding DB2 error message is shown. The SQL error is reported to the SAP application, which should handle it appropriately.

System Action: Processing continues.

User Response: In most cases, you can safely ignore this message, since the SAP application logic should handle these errors. If you see unexpected symptoms in your SAP application, then this message might help in diagnosing an error. In this case, save the ICLI message

log file and use it to resolve the SAP application error. In order to see the sequence of the SQL statements that lead to the error, you might want to re-run the SAP application logic while having set the ICLI server environment variable ICLI_TRACE_LEVEL to 2 or 3. See "ICLI server trace" on page 119.

ICLS1401S   Severe system error: The ICLI work thread may not execute properly. The ICLI main thread will wait forever for the work thread to signal the condition and can no longer accept connection requests. Processing of active ICLI server work threads continues. Signaling condition R/C = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx* Message: *sss*.

Explanation: This is a severe system error. The ICLI work thread might not work properly. The ICLI main thread will wait forever for the work thread to signal the condition. It can no longer accept connection requests.

System Action: Processing of already active ICLI server work threads continues. The ICLI server can no longer accept new connection requests.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1402E   System error: Accept processing returned with an error: accept() for socket *nnn* R/C = *nnn* errno = *nnn* (*xxx*) errno2 = *xxx* Message: *sss*. This ICLI thread will exit. Check the message file for preceding messages from this thread.

Explanation: This is a system error. Accept processing returned with an error. Check the message file for preceding messages from this thread.

System Action: This ICLI thread will exit. All other threads continue.

User Response: Check the message file for preceding messages from this thread, which may give a hint of the reason causing the error and try to solve the problem. Save all available diagnostic information and report the error to IBM/SAP.

ICLS1403E   System error in line *nnn* and file *sss*. Receive failed. System call recv() for socket *nnn* R/C = *nnn* errno = *nnn* (*xxx*), errno2 = *xxx*,

Message: *sss*.
This ICLI thread will exit.

Explanation: The recv() system call returned with a negative return code. See 'Message' for further explanations of the error.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: If errno/errno2 indicate a network problem which you or your network group cannot explain, then save all available diagnostic information and report the error to IBM/SAP.

ICLS1404W   The TCP connection was closed (line *nnn* **and file** *sss***).
System call recv() for socket** *nnn* **R/C =** *nnn*
errno = *nnn* **(***xxx***), errno2 =** *xxx*
This ICLI thread will exit.

Explanation: This message indicates that the TCP connection was closed. If 'errno' and/or 'errno2' are not 0, then check the explanation of the reported values in *z/OS UNIX System Services Messages and Codes*.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: If errno and errno2 are not null, it may be necessary to check your network components.

ICLS1405E   Internal error in line *nnn* **and file** *sss***:
Less than 4 bytes
were received with one recv() system
call.
recv() for socket** *nnn* **R/C =** *nnn*
This ICLI thread will exit.

Explanation: This is an internal ICLI error. Less than 4 bytes were received with one recv() system call.

System Action: This ICLI thread is terminated. All other threads continue to be processed.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1406E   Internal error in line *nnn* **and file** *sss***:
The ICLI client sent
more bytes than the ICLI server can
process.
Number of bytes to be received:** *nnn*
Maximum buffer size: *nnn*
This ICLI thread will exit.

Explanation: An internal error occurred in the ICLI protocol. The ICLI client sent more bytes than the ICLI server can process. This error message indicates an internal protocol error between ICLI client and server.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1407E   Internal error in line *nnn* **and file** *sss***:
Too many bytes
were received.
Number of bytes received: (***nnn***)
Number of bytes expected: (***nnn***).
This ICLI thread will exit.**

Explanation: An internal error in the ICLI protocol occurred. The ICLI client sent more bytes than the ICLI server expected. This error message indicates a protocol error between ICLI client and server.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

**ICLS1408E   Internal error in the ICLI protocol: The
ICLI client sent
an unknown request (***nnn***).
This ICLI thread exits.**

Explanation: An internal error in the ICLI protocol occurred. The ICLI client sent a request which is unknown.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1409E   Internal error in line *nnn* **and file** *sss***:
An SQL statement sent by the client
contained too
many DB2 parameter markers. The
statement could not
be executed. Number of DB2 parameter
markers
received:** *nnn*
Allowed maximum: *nnn*

Explanation: This is an internal error. An ICLI client tried to execute an SQL statement which contained more parameter markers than the allowed maximum.

System Action: The SQL statement is not executed. Processing continues.

User Response: Determine the cause of the error by inspecting the SAP application which sent the SQL statement. Reproduce the error situation with SQL tracing switched on, so you can see which statement caused the error situation. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1410E System error: The attempt to send data to the ICLI
client failed. This ICLI thread will exit.
send() for socket *nnn* R/C = *nnn*
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.

Explanation: A system error was encountered when the ICLI server tried to send data to the ICLI client.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: If errno/errno2 indicate a network problem which you or your network group cannot explain, then check the z/OS System Log for possible error messages and also check any messages that precede this message in the ICLI message file. Save all available diagnostic information and report the error to IBM/SAP.

ICLS1411W An error was encountered when the DB2 module DSNTIAR
was called to display error messages.
DSNTIAR warning in line *nnn* and file *sss*:
SQL Error text is longer than message buffer.
SQL Error message will be truncated.

Explanation: An error was encountered when calling DB2 module DSNTIAR to display SQL error messages. The error text that DSNTIAR returned was longer than the message buffer that was supplied by the ICLI server.

System Action: Processing continues. SQL error message is truncated.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1412E System error in line *nnn* and file *sss*:
The attempt to
call the DB2 module DSNTIAR to
display SQL error
messages failed.
An internal DSNTIAR error occurred,
R/C = (*nnn*).

Explanation: An error was encountered when calling DB2 module DSNTIAR to display SQL error messages. For further explanations, see *DB2 for OS/390 and z/OS Messages and Codes*.

System Action: Processing continues.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1413E Internal error: The ICLI client will not be able to
connect to the ICLI server. The ICLI server received
an invalid eyecatcher while performing a version check (*sss*).
The error may be caused by a corrupted data buffer.

Explanation: This is an internal ICLI error. The ICLI server received an unexpected eyecatcher. The reason may be a corrupted data buffer.

System Action: Processing continues, but the ICLI client will not be able to connect to the ICLI server.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1414W Connection has been shut down during termination
processing in line *nnn* and file *sss*.
System call recv() for socket *nnn* R/C = *nnn*
This ICLI thread will exit.

Explanation: This message indicates that the TCP connection was closed during termination processing of the ICLI server.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: None.

ICLS1415E System Error in line *nnn* and file *sss*:
Connection has been closed.
System call recv() for socket *nnn* R/C = *nnn*
errno = *nnn* (*xxx*), errno2 = *xxx*
Message: *sss*.
This ICLI thread will exit.

Explanation: This message indicates that the TCP connection was closed. Check the explanation of the reported values in *z/OS UNIX System Services Messages and Codes*.

System Action: This ICLI thread is stopped. All other threads continue.

User Response: If errno/errno2 indicate a network problem which you or your network group cannot explain, then save all available diagnostic information and report the error to IBM/SAP.

ICLS1416E Error: Open plan <*sss*> failed in DB2
subsystem <*sss*>.
The RRSAF function
CREATE_THREAD returned with
SQLCODE = *nnn*.
This thread will exit.

Explanation: During connect to DB2 the plan cannot

| be opened in the DB2 subsystem. A verbose DB2
| message ICLS1400W precedes this message if the
| ICLI_MSGLEVEL environment variable is set to 'W'.

| **System Action:** This ICLI thread cannot connect to
| DB2 and therefore is stopped. No other thread can
| connect to DB2 either.

| **User Response:** Check whether plan and DB2
| subsystem are appropriate. For more information see
| the preceding message ICLS1400W if available.

ICLS1418E   **Error: An SAP application encountered
an SQL error
(SQLCODE *nnn*). The error is reported
to the SAP
application, which should handle it
appropriately.**

Explanation: An SAP application program
encountered an SQL error. The SQL error is reported
back to the SAP application which should handle it
appropriately. The verbose DB2 error message
ICLS1400W is only written to the message file if the
ICLI_MSGLEVEL environment variable is set to 'W'.

System Action: Processing continues.

User Response: In most cases you can safely ignore
this message, since the SAP application logic should
handle these errors. If you see unexpected symptoms in
your SAP application, then this message might help in
diagnosing an error. In this case save the ICLI message
log file and use it to resolve the SAP application error.
In order to see the sequence of SQL statements which
have lead to the error you might want to re-run the
SAP application logic while having set the ICLI server
environment variable ICLI_TRACE_LEVEL to 2 or 3.
See "ICLI server trace" on page 119.

ICLS1420I   **The ICLI client sent a DISCONNECT
request to the ICLI work
thread. This thread will exit.**

Explanation: An ICLI client sent a DISCONNECT
request to the ICLI work thread. This will cause the
ICLI work thread to end normally.

System Action: The ICLI thread will end, but all other
threads continue.

User Response: None

ICLS1421E   **System error in line *nnn* and file *sss*:
The attempt to call
the Language Environment routine *sss*
failed with
return code *nnn*. This thread will exit.
Condition
and abend handling for the ICLI server
will not be
available.**

Explanation: This is a system error. Calling a

Language Environment routine failed. Condition and
abend handling of the ICLI server will be unavailable.

System Action: The ICLI thread will end, but all other
threads continue.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1422I   **An SQL error occurred. The following
SQL error message
was issued:**
sss
sss
sss
sss
sss
sss
sss
sss
sss
sss
**The SQL error is reported to the SAP
application,
which should handle it appropriately.**

Explanation: An SQL error occurred. The
corresponding DB2 error message is shown. The SQL
error is reported back to the SAP application which
should handle it appropriately.

System Action: Processing continues.

User Response: In most cases you can safely ignore
this message, since the SAP application logic should
handle these errors. If you see unexpected symptoms in
your SAP application, then this message might help in
diagnosing an error. In this case save the ICLI message
log file and use it to resolve the SAP application error.
In order to see the sequence of SQL statements which
lead to the error you might want to re-run the SAP
application logic while having set the ICLI server
environment variable ICLI_TRACE_LEVEL to 2 or 3.
See "ICLI server trace" on page 119.

ICLS1423E   **Error: The name of the DB2 stored
procedure (*nnn*)
is too long. It either exceeds the allowed
maximum (*nnn*)
or is shorter than 1. These are the first
19 characters
of the procedure name: *sss*.
The ICLI thread continues processing.**

Explanation: The ICLI client passed a DB2 stored
procedure name which is longer than the allowed DB2
maximum for such names.

System Action: The ICLI thread continues processing.

User Response: Check the usage of DB2 stored
procedure names by the SAP application. If do not find
any problem there, the error may point to an ICLI
interface problem. Save all available diagnostic

information report the error to SAP/IBM.

**ICLS1424W   The ICLI client closed its TCP/IP
connection while the
ICLI server was performing a version
check.
Connection closed in line** *nnn* **and file**
*sss***.
recv() for socket** *nnn* **R/C =** *nnn*
errno = *nnn* **(***xxx***), errno2 =** *xxx*
**The ICLI server thread will continue to
listen for new
connection requests.**

Explanation: This message indicates that the ICLI
client has closed its end of the TCP connection while
the ICLI server was checking the ICLI client version.
This might for example occur if the SAP application is
terminated by a signal.

System Action: The ICLI server listening thread will
continue to wait for new connection requests.

User Response: None.

ICLS1425E   Error in line *nnn* **and file** *sss***: The ICLI
client
passed a wrong Large Object Type
(LOB) value to the
requested operation** *sss***.
The specified type is** *nnn***, expected type
is DB2_BLOB_LOCATOR (961) or
DB2_CLOB_LOCATOR (965).
This thread exits.**

Explanation: The ICLI client passed a wrong Large
Object Type (LOB) value to the requested operation.

System Action: The ICLI thread stops. The error is
reported to the SAP application.

User Response: Check the usage of Binary LOB
(BLOB) or Character LOB (CLOB) manipulation in the
SAP application. If you do not find any problem, save
all available diagnostic information and report the error
to SAP/IBM.

**ICLS1426E   System error: An attempt to allocate
memory (size =** *nnn***)
for a receive buffer failed:
errno =** *nnn* **(***xxx***), errno2 =** *xxx***.
Message:** *sss***.
This thread will exit. To resolve the
problem, adjust
the REGION size of ICLI server.**

Explanation: This is a system error. Memory allocation
for the receive buffer for this thread could not be
obtained.

System Action: The ICLI thread stops.

User Response: Increase your REGION size before
starting the ICLI server. If the error recurs, save all

---

available diagnostic information and report the error to
SAP/IBM.

ICLS1427W   The stored procedure *sss* **provided more
than** *nnn*
**result sets, but ICLI server will only
forward the
first** *nnn* **result sets. The DB2 statement
ASSOCIATE
returned SQLCODE =** *nnn***. The ICLI
thread continues
processing, but some of the result sets
cannot be read.**

Explanation: A stored procedure was executed which
opened more cursors than supported by ICLI server.

System Action: The ICLI thread continues but some of
the result sets cannot be read.

User Response: Check the DB2 stored procedure
called by the SAP application. It should not open more
cursors than the number mentioned in the message. If
this is the case, then the error can point to an ICLI
interface problem. Save all available diagnostic
information report the error to SAP/IBM.

ICLS1428E   Error in line *nnn* **and file** *sss***: The
BLOB/CLOB length
value passed by the ICLI client exceeds
the limit
defined for the ICLI server. The length
value for
the requested operation (***sss***) is too high.
Specified length is** *nnn***, maximum
allowed length is** *nnn***.
The ICLI thread continues processing.**

Explanation: The ICLI client passed a length value for
Binary LOB (BLOB) or Character LOB (CLOB)
manipulation which is longer than the maximum
allowed value defined by the ICLI server.

System Action: The ICLI thread continues processing.
The error is reported to the SAP application.

User Response: Check the usage of Binary LOB
(BLOB) or Character LOB (CLOB) manipulation in the
SAP application. If you do not find any problem, save
all available diagnostic information and report the error
to SAP/IBM.

ICLS1430E   System error in line *nnn* **and file** *sss***:
The ICLI server
encountered an error when it tried to
send data
to the ICLI client during a version
check.
System call sendto() for socket** *nnn* **R/C
=** *nnn*
errno = *nnn* **(***xxx***), errno2 =** *xxx*
Message: *sss***.**

The ICLI server will wait for new connection requests.

Explanation: A system error was encountered when the ICLI server tried to send data to the ICLI client during version checking.

System Action: The ICLI server thread will continue to listen for connection requests.

User Response: If errno/errno2 indicate a network problem which you or your network group cannot explain, then save all available diagnostic information and report the error to IBM/SAP.

ICLS1502E **Error: DB2 subsystem *sss* is not active.**
The ICLI server cannot connect to the DB2 subsystem.
The corresponding ICLI server thread terminates.

Explanation: An ICLI server thread tried to connect to a DB2 subsystem. The DB2 subsystem which was passed to the ICLI server was not started.

System Action: This ICLI thread is stopped. All other threads continue processing.

User Response: Start the DB2 subsystem and then restart the SAP application.

ICLS1503E **Error: The DB2 subsystem *sss* could not be found.**
The ICLI server cannot connect to the DB2 subsystem.
The corresponding ICLI server thread terminates.

Explanation: The DB2 subsystem name that was passed to the ICLI server could not be found. Possible reasons are misspelled DB2 subsystem names in the SAP profiles or the environment variables on the client side.

System Action: ICLI thread is stopped.

User Response: Specify a valid DB2 subsystem in either SAP profiles or the environment on the client side and restart the SAP application.

ICLS1510W **Application error: The application program tried to invoke RRSAF IDENTIFY while it was already connected to the DB2 subsystem *sss*.**
Reason code = 0x*xxx*.

Explanation: The application program invoked RRSAF IDENTIFY while it was already connected to DB2. This situation can occur with SAP applications that have a 'retry mechanism'.

System Action: Processing continues.

User Response: None.

ICLS1512E **System error: The IDENTIFY request to connect to the DB2 subsystem '*sss*' was rejected, because DB2 is running in restricted access mode.**
Reason code: 0x*xxx*.
The ICLI server cannot connect to the DB2 subsystem.
The corresponding ICLI server thread terminates.

Explanation: An IDENTIFY request for connection to DB2 was rejected. DB2 had been started in restricted access mode. Only authorization IDs authorized to perform maintenance functions are permitted access to DB2.

System Action: The connection request fails; the error is reported to the ICLI client.

User Response: Find out why the subsystem was started in restricted access mode. Retry when the subsystem is restarted in general access mode.

ICLS1513E **System error: The application program issued an RRSAF IDENTIFY function request, but RRSAF is not available.**
The IDENTIFY request will not be processed.
The error is reported to the ICLI client.
Reason code: 0x*xxx*.

Explanation: The application program issued an RRSAF IDENTIFY function request, but RRSAF is not available.

System Action: The IDENTIFY request is not processed; the error is reported to the ICLI Client.

User Response: Retry after RRSAF has been started.

ICLS1514E **Internal error: DB2 thread termination was requested, but the thread has not reached a point of consistency. The function request is not processed.**
The error is reported to the ICLI client.
Reason code: 0x*xxx*.

Explanation: The application program issued an RRSAF TERMINATE THREAD or TERMINATE IDENTIFY function request, but the application has issued an SQL request since the last invocation of COMMIT or ROLLBACK and therefore is not at a point of consistency. This is an internal error.

System Action: The function request is not processed; the error is reported to the ICLI client.

User Response: None. Save all available diagnostic information and report the error to SAP/IBM.

ICLS1515E   **System error: An error was encountered in either DB2 or *sss* RRSAF. The application abends. In many cases, DB2 will abend as well. The error is reported to the ICLI client. Request '*sss*' returns *nnn* with reason code 0x*xxx*.**

Explanation: A system error was detected in either DB2 or z/OS RRSAF.

System Action: The application is abended. This error may, in many cases, abend the DB2 subsystem; the error is reported to the ICLI Client.

User Response: Collect the following diagnostic items listed in IBM manual *DB2 for OS/390 and z/OS Messages and Codes* for reason code <nnn> and report the error to SAP/IBM.

ICLS1516E   **A failure occurred during processing of request '*sss*'. The DB2 subsystem abends. The error is reported to the ICLI client. Return code: *nnn* Reason code = 0x*xxx*.**

Explanation: The DB2 subsystem was abended after a failure occurred during processing of request <sss>. The recovery routine DSN3SSES attempted to clean up processing and pass the error to the caller, but was unable to do so.

System Action: The DB2 subsystem is terminated; the error is reported to the ICLI client.

User Response: The DB2 subsystem should be restarted. Collect the following diagnostic items listed in IBM manual *DB2 for OS/390 and z/OS Messages and Codes* for reason code <xxx> and report the error to SAP/IBM.

ICLS1517E   **System error in the RRSAF connection: Call of request '*sss*' failed with return code *nnn* and reason code 0x*xxx*. RRSAF function return code: *nnn* The error is reported to the ICLI client.**

Explanation: An error occurred in request <sss>. See *DB2 for OS/390 and z/OS Messages and Codes* for a detailed description of the return code and reason code values.

System Action: The error is reported to the ICLI client.

User Response: Examine the DB2 return and reason codes and try to resolve the problem. If the error cannot be resolved, save all available diagnostic information and report the error to SAP/IBM.

ICLS1518E   **System error: The DB2 function DSNRLI TRANSLATE encountered an error (RC: *nnn*). Diagnostic information: DSNRLI TRANSLATE: return code: *nnn* reason code: 0x*xxx* File *sss*, line *nnn***

Explanation: The DB2 function DSNRLI TRANSLATE encountered an error. See *DB2 for OS/390 and z/OS Messages and Codes* for a detailed description of the return code and reason code values. The ICLI server message file contains for preceding DB2 errors.

System Action: Processing continues.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1519E   **A TERMINATE THREAD command was issued, but no thread was active (reason code 0x*xxx*). Processing continues. The error is reported to the ICLI client.**

Explanation: The RRSAF command TERMINATE THREAD was issued when there was no active thread.

System Action: Processing continues, the error is reported to the ICLI client.

User Response: The reason for the problem is described in previous error messages, try to resolve this problem. If you cannot resolve it, save all available diagnostic information and report the error to SAP/IBM.

ICLS1530E   **An invalid value (*sss*) was specified for environment variable ICLI_WRITE_TO_SYSLOG. Valid values are 0 and 1, if the environment variable is set. Processing continues, but the ICLI server will not write to the system log.**

Explanation: You specified an invalid value for environment variable ICLI_WRITE_TO_SYSLOG.

System Action: Processing continues. ICLI server didn't write to system log.

User Response: Correct the value of ICLI_WRITE_TO_SYSLOG to 1 or 0 (or remove this environment variable).

ICLS1601E   **System error in line *nnn* and file *sss*: pthread_setspecific() returned with RC *nnn*. Processing continues, but some functionality**

(e.g., tracing capability) may not be
available.
errno = *nnn* (0x*xxx*), errno2 = 0x*xxx*
Message: *sss*.

Explanation: The C standard library function
pthread_setspecific() returned an error. The ICLI server
uses this function to set thread specific information, e.g.
trace flag and strings that help in monitoring the ICLI
or analyzing problem situations.

System Action: Processing continues, but some
functionality (e.g. tracing capability) may be lost.

User Response: If the error message <sss> is not
sufficient to resolve the problem, save all available
diagnostic information and report the error to
SAP/IBM.

**ICLS1602E   System error in line *nnn* and file *sss*:
Memory for trace buffers could not be
allocated,
because insufficient storage was
available.
Processing continues, but tracing is
disabled for
this thread.**

Explanation: Memory for trace buffers for this thread
could not be obtained.

System Action: Processing continues, tracing is
disabled for this thread.

User Response: Resolve your memory shortage.

**ICLS1603E   System error: Trace file *sss* could not be
opened:
Tracing will be disabled for this thread.
errno = *nnn* (0x*xxx*)
errno2 = 0x*xxx*
Message: *sss*.**

Explanation: The trace file for this thread could not be
opened. A further explanation of the error is given in
'Message'. Most common reasons are: directory is not
accessible because of permission settings or file system
full condition.

System Action: Processing continues, tracing is
disabled for this thread.

User Response: If you need the traces, remove the
cause of the error (for example by setting the
permission or ensuring that the file system is large
enough), and restart the ICLI server.

**ICLS1604E   System error in line *nnn* and file *sss*:
pthread_getspecific() returned with
RC=*nnn*.
errno = *nnn* (0x*xxx*), errno2 = 0x*xxx*
Message: *sss*.
Processing continues, but some
functionality**

(e.g., tracing capability) may not be
available.

Explanation: The C standard library function
pthread_getspecific() returned an error. The ICLI server
uses this function to manage thread specific
information, e.g. trace flags.

System Action: Processing continues, but some
functionality (e.g. tracing capability) may be lost.

User Response: If the description in 'Message' is not
sufficient to resolve the problem, save all available
diagnostic information and report the error to
SAP/IBM.

**ICLS1605W   An invalid value (*nnn*) was specified for
environment
variable ICLI_TRACE_LEVEL.
Valid values are *nnn* through *nnn*.
The ICLI trace level is set to the default
value (*nnn*).**

Explanation: The environment variable
ICLI_TRACE_LEVEL was set to an invalid value before
starting the ICLI server. The trace level is reset to the
default trace level.

System Action: Processing continues with trace level
set to the default.

User Response: Make sure that you specify valid
values for the environment variable
ICLI_TRACE_LEVEL before you start the ICLI server. If
you do not set this variable, a default will be used.

**ICLS1606E   Error in line *nnn* and file *sss*: Tracing is
not possible,
because the trace file for this thread has
already been closed.**

Explanation: The trace file for this thread has been
closed. Most common reasons are: During ABEND
handling the trace file was closed.

System Action: Processing continues, tracing is
disabled for this thread.

User Response: Look for preceding error messages
and correct the cause of that error (for example the
cause of an ABEND).

**ICLS1607E   An attempt to write to the trace file
failed.
The file system may be full. Tracing is
now disabled.
Diagnostic information:
vfprintf() errno = *nnn*
Message: *sss***

Explanation: Writing to the trace file failed. Most
probable reason is a filesystem full condition.

System Action: Processing continues, tracing is disabled for this thread.

User Response: If you need the trace information, remove the cause of the error (for example, provide more space in the file system) and restart the ICLI server. If you do not need the trace information, check the ICLI_TRACE_LEVEL environment variable, set it to 0 and remove unnecessary files before you restart the ICLI server.

ICLS1608E **An attempt to write to the ICLI server message file failed.**
The file system may be full. All further messages
will be written to STDERR.
errno = *nnn* **(0x***xxx***), errno2 = 0x***xxx*
Message: *sss***.**

Explanation: Writing to the ICLI server message file failed. Most probable reason is a filesystem full condition. This message is written to STDERR instead of the ICLI server message file. Without further messages being written to the ICLI server message file, the error determination will be very difficult.

System Action: Processing continues, message logging is disabled.

User Response: Remove the cause for the error (provide more space in the filesystem).

ICLS1609E **System error: Memory for the DB2 Connection Information**
could not obtained for this thread. This thread exits.

Explanation: This is an internal ICLI error. Memory for the DB2 Connection Information cannot be obtained for this thread.

System Action: This thread exits.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1610E **Error in pthread_once() routine R/C =**
nnn
errno = *nnn* **(***xxx***), errno2 =** *xxx*
Message: *sss***.**

Explanation: This is an internal ICLI error. A ICLI server encountered an internal error when processing work to be done once and only once during the life time of the ICLI server process together with its threads.

System Action: Processing continues, but some functionality (e.g. tracing capability) may be lost.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1701W **System warning: A code page converter could not be opened.**
Conversion from CP *sss* **to CP** *sss* **failed.**
Processing continues, but the SQL trace may be
unreadable. Function iconv_open(),
errno = *nnn* **(0x***xxx***), errno2 = 0x***xxx*
Message: *sss***.**

Explanation: An ICLI trace function tried to open a code page (CP) converter. This is necessary to make ASCII strings readable in an SQL trace. See the description of the SQL trace format for details on where this code page conversion happens. As a result, SQL traces might not be usable. The error message ICLS1702S might appear later in the ICLI server message file.

System Action: Processing continues. SQL traces might not be usable.

User Response: If you cannot solve the problem using errno and the message, save all available diagnostic information and report the error to SAP/IBM.

ICLS1702W **System error: Code page conversion from CP** *sss* **to CP** *sss* **failed.**
Processing continues, but the SQL trace may be unreadable.
Function iconv(), errno = *nnn* **(0x***xxx***), errno2 = 0x***xxx*
Message: *sss***.**

Explanation: A ICLI trace function tried to do a code page (CP) conversion. This is necessary to make ASCII strings readable in an SQL trace. See the description of the SQL trace format for details on where this code page conversion happens. As a consequence, SQL traces might not be usable.

System Action: Processing continues. SQL traces might not be usable.

User Response: If you cannot solve the problem using errno and the message, save all available diagnostic information and report the error to SAP/IBM.

ICLS1705I **Diagnostic information: An ICLI server dump buffer is**
produced, processing continues.
sss sss

Explanation: This message is generated as an aid in diagnosing errors. The buffer printout follows with 32 bytes displayed in each row. The starting address of each row of 32 bytes is printed first. Following are the 32 bytes in hex and after that the EBCDIC character representation of the 32 bytes. Unprintable characters are displayed as dots. The reason for writing out this message can be a preceding error, an ICLI health check or cancel thread, so no further action might be necessary.

System Action: Processing continues.

User Response: Look for an eventually preceding error message and its user response.

ICLS1800E **System error: A semaphore could not be created.**
Processing continues, but the ICLI Performance monitor
functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
semget() RC = *nnn***, errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss***.**

Explanation: The system call semget() failed for the reason indicated in the 'Message'.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1801E **System error: A semaphore could not be initialed.**
Processing continues, but the ICLI Performance monitor
functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
semctl() RC = *nnn***, errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss***.**

Explanation: The system call semctl() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1802E **System error: An attempt to call a semaphore failed in**
line *nnn* **and file** *sss***.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
semget() RC = *nnn***, errno =** *nnn* **(0x***xxx***),**

errno2 = 0x*xxx*
Message: *sss***.**

Explanation: The system call semget() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1803E **System error: An attempt to call a semaphore failed.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
semop() RC = *nnn***, errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss***.**

Explanation: The system call semop() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1804E **System error: A semaphore could not be released.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
semop() RC = *nnn***, errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss***.**

Explanation: The system call semop() failed for the reason indicated in the message.

System Action: Processing continues, but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details will not work any more.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1805E **System error: A semaphore could not be removed.**
A small amount of system memory was not freed,
although it is no longer needed.
Diagnostic information:
semctl() RC = *nnn*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.

Explanation: The system call semctl() failed for the reason indicated in the message.

System Action: Processing continues. A small amount of system memory is not freed, although it is not needed any more.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1806E **System error: Shared memory could not be created.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
shmget() RC = *nnn*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.

Explanation: The system call shmget() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1807E **System error: Shared memory could not be accessed.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
shmget() RC = *nnn*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.

Explanation: The system call shmget() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic

information and report the error to SAP/IBM.

ICLS1808E **System error: Shared memory could not be attached.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that collect
thread-related information are not available.
Diagnostic information:
shmat() RC = *ppp*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.

Explanation: The system call shmat() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1809E **System error: Shared memory could not be detached.**
Processing continues, but the ICLI Performance
monitor functions (SAP transaction ST04) that
collect thread-related information are not available.
Diagnostic information:
shmdt(), RC = *nnn*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.

Explanation: The system call shmdt() failed for the reason indicated in the message.

System Action: Processing continues but the ICLI Performance monitor functions (SAP transaction ST04) concerning thread details will not work any more.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1810E **System error: Shared memory could not be removed.**
Some system memory has not been freed, although
it is no longer needed.
Diagnostic information:
shmctl() RC = *nnn*, **errno =** *nnn* **(0x***xxx***),**
errno2 = 0x*xxx*
Message: *sss*.
Processing continues.

Explanation: The system call shmctl() failed for the reason indicated in the message.

System Action: Processing continues. Some system

memory is not freed, although it is not needed any more.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1811E **The maximum number of users who can concurrently access**
SAP transaction ST04 has been reached.
Allowed maximum = *nnn*
User number = *nnn*

Explanation: This is an SAP internal error. The maximum number of users allowed to use the ICLI performance monitor concerning thread details (SAP transaction ST04) has been reached.

System Action: Processing continues. Performance monitor functions concerning thread details for this user are not available.

User Response: Save all available diagnostic information and report the error to SAP/IBM.

ICLS1812W **The amount of thread-activity-related data returned by**
DB2 exceeds the amount of shared memory that is currently
available. Processing continues, but the lock information
displayed via SAP transaction ST04 may be incomplete.

Explanation: An SAP user requested some thread details using SAP transaction ST04. The data returned by DB2 exceeds the amount of allocated shared memory.

System Action: Processing continues, but the data concerning lock information may be incomplete.

User Response: None.

ICLS1820E **System error in line** *nnn* **and file** *sss***:**
***nnn* bytes of memory could not be allocated.**
Processing continues. The ICLI server returns an error
to the calling process.
Diagnostic information:
errno = *nnn* **(0x***xxx***), errno2 = 0x***xxx*
Message: *sss***.**

Explanation: Memory for an internal data structure could not be obtained. One possible reason is memory shortage.

System Action: Processing continues. The ICLI server returns an error to the calling process.

User Response: Try to resolve the problem indicated by errno and the message, for example, increase your REGION size before starting the ICLI server.

ICLS1821E **DB2 error in line** *nnn* **and file** *sss***:**
The DB2 trace could not be started.
Processing continues.
Return code = 0x*xxx***, reason code = 0x***xxx*
DB2 message(s): *sss*

Explanation: This is a DB2 error. The reason code is described in *DB2 for OS/390 and z/OS Messages and Codes*.

System Action: Processing continues.

User Response: Try to remove the cause of the failure. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1822E **DB2 error: A DB2 authorization failure occurred,**
because a required privilege has not been granted.
The request was rejected by DB2.
Processing continues.
Return code = 0x*xxx***, reason code = 0x***xxx*
DB2 message(s): *sss*

Explanation: This is a DB2 error. The request was rejected by DB2, since at least one of the privileges TRACE, MONITOR1, and MONITOR2 has not been granted. The reason code is described in *DB2 for OS/390 and z/OS Messages and Codes*.

System Action: Processing continues.

User Response: Remove the cause of the failure. To grant the privileges refer to the SAP manual *SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server* in the section "Performance Monitoring". If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1823E **DB2 error in line** *nnn* **and file** *sss***:**
An unexpected DB2 return code was received.
Return code = 0x*xxx***, reason code = 0x***xxx*
DB2 message(s): *sss*

Explanation: This is a DB2 error. The reason code and the DB2 message, if available, are described in *DB2 for OS/390 and z/OS Messages and Codes*.

System Action: Processing continues.

User Response: Try to correct the cause of the failure, if possible. If the error recurs, save all available diagnostic information and report the error to SAP/IBM.

ICLS1824E DB2 error in line *nnn* and file *sss*:
The DB2 Instrumentation Facility
delivered an
incomplete IFI record.
Processing continues, but some DB2
performance data
cannot be processed.
Diagnostic information:
ICLI function = *nnn*
Additional information: 0x*xxx* 0x*xxx*
0x*xxx* 0x*xxx*

Explanation: This is a DB2 error. The DB2
Instrumentation Facility delivered an incomplete IFI
record.

System Action: Processing continues, but some DB2
performance data cannot be processed.

User Response: If you can reproduce the problem, set
ICLI_TRACE_LEVEL=6, reproduce it and reset the trace
level back to 0. Save all available diagnostic
information, including the ICLI server trace files, and
report the error to SAP/IBM.

ICLS1825I DB2 error: The given ACE is no longer
active.
Processing continues.
return code = 0x*xxx*, reason code = 0x*xxx*

Explanation: This is a DB2 error. Data was obtained
for a thread which is no longer active. The reason code
is described in *DB2 for OS/390 and z/OS Messages and
Codes*.

System Action: Processing continues.

User Response: None.

ICLS1826E Internal error in line *nnn* and file *sss*.
Processing continues, but some
functionality may
not be available.
Diagnostic information:
Internal reason code = *nnn*, function =
nnn
Additional information: 0x*xxx* 0x*xxx*
0x*xxx* 0x*xxx*

Explanation: This is an internal ICLI server error
which should never happen.

System Action: Processing continues but some
functionality may be lost.

User Response: If you can reproduce the problem, set
ICLI_TRACE_LEVEL=6, reproduce it and reset the trace
level back to 0. Save all available diagnostic
information and report the error to SAP/IBM.

ICLS1827E DB2 IFI error: The DB2 instrumentation
facility returned an
invalid IFCID *nnn*. The returned record
is incomplete:
Number of required sections: *nnn*
Number of returned sections: *nnn*
Processing continues, but the queried
data cannot
be displayed.

Explanation: This is a DB2 error. The DB2
instrumentation facility returned a record with missing
sections.

System Action: Processing continues but the queried
data cannot be displayed.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.
Additional diagnostic data can be gathered setting
ICLI_TRACE_LEVEL=6.

ICLS1828E DB2 IFI error: The DB2 instrumentation
facility returned
a record with an invalid product section
(invalid
IFCID: *nnn*). Length of product section
standard
header (QWHS): *nnn* .Processing
continues, but the
queried data cannot be displayed.

Explanation: This is a DB2 error. The DB2
instrumentation facility returned a record with an
invalid product section.

System Action: Processing continues, but the queried
data cannot be displayed.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.
Additional diagnostic data can be gathered setting
ICLI_TRACE_LEVEL=6.

ICLS1829E DB2 IFI error: The DB2 instrumentation
facility
returned a record with an invalid
product section
(invalid IFCID *nnn*). Type of product
section standard
header (QWHS): *nnn* Processing
continues, but the
queried data cannot be displayed.

Explanation: This is a DB2 error. The DB2
instrumentation facility returned a record with an
invalid product section.

System Action: Processing continues but the queried
data cannot be displayed.

User Response: Save all available diagnostic
information and report the error to SAP/IBM.

Additional diagnostic data can be gathered setting ICLI_TRACE_LEVEL=6.

ICLS1830E   A DB2 COMMAND failed:
return code = 0xxxx**, reason code = 0x**xxx
DB2 message(s): sss

Explanation: ICLI tried to execute a DB2 COMMAND without success. The reason code and the DB2 message, if available, are described in *DB2 for OS/390 and z/OS Messages and Codes*.

System Action: Processing continues.

User Response: Remove the cause of the failure, if possible. If the error recurs, set ICLI_TRACE_LEVEL=6, reproduce it and reset the trace level back to 0. Save all available diagnostic information, including the ICLI server trace files, and report the error to SAP/IBM.

ICLS1831E   DB2 error in line nnn **and file** sss**:**
The DB2 Instrumentation Facility
returned an invalid
IFI record. Processing continues, but
some DB2
performance data cannot be processed.
Diagnostic information:
ICLI function = nnn
Additional information: 0xxxx **0x**xxx
0xxxx **0x**xxx

Explanation: This is a DB2 error. The DB2 Instrumentation Facility delivered an invalid IFI record.

System Action: Processing continues but some DB2 performance data cannot be processed.

User Response: If you can reproduce the problem, set ICLI_TRACE_LEVEL=6, reproduce it and reset the trace level back to 0. Save all available diagnostic information, including the ICLI server trace files, and report the error to SAP/IBM.

ICLS1832E   DB2 error: Function get_ccsids()
returned with return code nnn
because of a DB2 problem.
This work thread cannot continue
processing.
DB2 message: sss

Explanation: The ICLI server could not retrieve correct tables to translate between ASCII and EBCDIC.

System Action: An appropriate error is sent to the client. Processing cannot continue without correct translation tables.

User Response: Check the DB2 message. Problems might be insufficient authority to start the DB2 monitor trace or to retrieve IFCID 106. Check also for other error messages preceding this one.

ICLS1833E   DB2 error: could not get translation
table from CCSID nnn **to CCSID** nnn**.**
This work thread cannot continue
processing.

Explanation: The ICLI server tried to fetch a codepage translation table from DB2. This SELECT failed.

System Action: An appropriate error is sent to the client. Processing cannot continue without correct translation tables.

User Response: Check for the SQLCODE of the DB2 error message preceding this one. Check if the corresponding table is available in SYSIBM.SYSSTRINGS and the ICLI server is authorized to read it.

ICLS1834E   Error: translation table from CCSID nnn
to CCSID nnn **has wrong length (**nnn**).**

Explanation: The codepage translation table retrieved from DB2 has a wrong length. Only tables of length 256 are supported.

System Action: An appropriate error is sent to the client. Processing cannot continue without correct translation tables.

User Response: Check if the corresponding table in SYSIBM.SYSSTRINGS is corrupted.

ICLS1835E   Error: translation tables between CCSID
nnn **to CCSID** nnn **are not bijective at**
index nnn**.**

Explanation: The codepage translation tables retrieved from DB2 are not bijective. To run SAP you have to configure DB2 to use ASCII and EBCDIC codepage settings that translate bijective to avoid any data loss or data corruption.

System Action: An appropriate error is sent to the client. Processing cannot continue without correct translation tables.

User Response: Check the codepage settings of DB2 and the corresponding translation tables in SYSIBM.SYSSTRINGS. For easy validation of the tables they are hex dumped into the ICLI server message file (message ICLS1705I).

ICLS1900E   Message ICLSnnnsss **(Message Set** nnn**)**
could not be
retrieved from Message Catalog sss**.**

Explanation: Either the message catalog that is used to retrieve a message text did not contain the message, or the specified message set was not found in the message catalog. The ICLI message catalog has only one message set; this number specifies the current level. Therefore, if the message set is not found there is a mismatch between the APAR level of the ICLI server code and the APAR level of the message catalog.

System Action: The message was not retrieved and cannot be displayed. Processing continues.

User Response: To verify that the correct message catalog exists, you can use the USS command dspcat:

```
dspcat <NLSPATH>/fome620m.cat      or
dspcat <NLSPATH>/fome640m.cat
```

The message set of the ICLS1900E message you have found in the ICLI message file needs to be the same as the output from this command. If it is the same, ICLI could have a problem if the message is missing, for example. If the message set is not the same, an environment problem is likely, such as one of the following situations:

- The apply of the PTF was not 100% successful. After installing a new ICLI PTF, you should find the correct message catalog in /usr/lpp/icli/lib/nls/msg/C and a softlink to this directory in /usr/lib/nls/msg/C/.
- You do not have the right NLS path to the correct message catalog. The shipped message catalogs in English, for example, are activated through:

```
 ENTER  export NLSPATH=/usr/lib/nls/msg/%L/%N
        export LANG=C
```

- (6.20 only) Since the 6.20 ICLI must reside in SYS1.LNKLIB, a missing LLA refresh might be the problem.

ICLS1901S **The attempt to open message catalog** *sss*
failed.
The ICLI server is stopped.
errno = *nnn* (0x*xxx*), errno2 = 0x*xxx*
Message: *sss*.

Explanation: The Message catalog that is used to retrieve ICLI server messages could not be opened.

System Action: The ICLI server is stopped.

User Response: Check the setting of the environment variable NLSPATH. It must point to the directory where the message catalogs reside. The default for this variable is "/usr/lib/nls/msg/%L/%N". Refer to the manual *z/OS UNIX System Services Command Reference* for detailed information.

ICLS1902W **Internal problem in message reporting**
in
line *nnn* and file *sss*:
Severity *nnn* is not supported.
Severity is reset to the default value
(*sss*).

Explanation: The ICLI server issued a message with an invalid severity value. The message severity is reset a default value. If the message is not found, an error ICLS1900E will be reported.

System Action: Processing continues.

User Response: Save all available diagnostic

information and report the error to SAP/IBM.

ICLS1903E **Thread *xxxxxx* encountered an ABEND**
(secondary error)
while terminating the thread because of
a previous error.
This ICLI thread terminates. Other ICLI
threads are not
affected.

Explanation: The ICLI server got an ABEND (secondary error) during signal handling of a (primary) error. This message is written to stderr.

System Action: Current ICLI thread terminates. Other ICLI threads are not affected.

User Response: Check the reason for the primary error. Save all available diagnostic information of the primary and secondary error and report the error to SAP/IBM.

ICLS1904E **Configuration error: This is DB2 version**
***c.c.c*.**
Only DB2 V6, V7, and (in compatibility
mode) V8 are supported.

Explanation: The SAP work process connected to a DB2 subsystem running a DB2 version that is not supported using ICLI.

System Action: The ICLI will disconnect from this DB2 subsystem and cannot process any requests until the work process connects to another DB2 subsystem.

User Response: Check if you have configured your system to use the correct DB2 subsystem. ICLI supports only DB2 versions 6, 7, and (in compatibility mode) 8. Use DB2 Connect for DB2 V8 in new-function mode or later DB2 versions.

List of ICLI exit values

The ICLI can exit with different return codes. These so-called *exit values* are integers ranging from 0 to 255 (only those indicated below are currently in use). Before the ICLI exits, an error message is written. This error message provides more information as to why the exit occurred. The following table lists each possible exit value and the corresponding error message(s) which are written.

Table 29. ICLI exit values and corresponding error messages

Exit value		Error message
(decimal)	(hexadecimal)	
1	1	ICLS1000
2	2	ICLS1001
3	3	ICLS1004
4	4	ICLS1005
5	5	ICLS1006
6	6	ICLS1008
7	7	ICLS1009
8	8	ICLS1010
9	9	ICLS1016
10	A	ICLS1017
11	B	ICLS1018
12	C	ICLS1019
13	D	ICLS1020
14	E	ICLS1036
15	F	ICLS1037
16	10	ICLS1038
17	11	ICLS1039
19	13	ICLS1043
20	14	ICLS1044
21	15	ICLS1047
22	16	ICLS1056
23	17	ICLS1059
24	18	ICLS1060
25	19	ICLS1062
26	1A	Not used
27	1B	Not used
28	1C	ICLS1074
29	1D	ICLS1075
30	1E	ICLS1076
31	1F	ICLS1084
32	20	ICLS1085
33	21	ICLS1123
34	22	ICLS1190

Table 29. ICLI exit values and corresponding error messages (continued)

Exit value		Error message
(decimal)	(hexadecimal)	
35	23	ICLS1351
36	24	ICLS1353
37	25	ICLS1421
38	26	ICLS1601
39	27	ICLS1802, ICLS1805, ICLS1806, ICLS1807, ICLS1808, ICLS1809, ICLS1810
40	28	ICLS1820
41	29	ICLS1901
42	2A	In this case, there is no corresponding error message. This exit occurs when you have used the ICLI server command line option *-CHECK*. You will find information on whether the check yielded positive or negative results by looking in the standard error output file STDERR.
43	2B	ICLS1091
44	2C	ICLS1124
45	2D	ICLS1041
46	2E	ICLS1090

List of abbreviations

The abbreviations used in this document are listed below. For more detailed explanations, refer to the glossary.

ABAP/4	Advanced Business Application Programming 4th Generation Language (SAP)
abend	Abnormal end of task
ACE	Auxiliary Control Element
ACS	Automated Class Selection
AFP	Advanced Function Printing (or Presentation)
AI	Accounting information (WLM work qualifier)
AIX	Advanced Interactive Executive (IBM implementation of UNIX)
ALE	Application Linking and Embedding
ANSI	American National Standards Institute
APAR	Authorized Program Analysis Report
APF	Authorized Program Facility
API	Application Program Interface
APPC	Advanced Program-to-Program Communication
ARM	Automatic Restart Management
AS	Application Server
ASCII	American National Standard Code for Information Interchange
ASID	Address Space ID
ASVT	Address Space Vector Table
ATM	Asynchronous Transfer Mode
BAPI	Business Application Program Interface
BC	Basic Component
BLOB	Binary Large Object
BSDS	Bootstrap Dataset
BTC	Batch (SAP process type)
BW	Business Information Warehouse
CAS	Catalog Address Space
CBU	Capacity Backup Upgrade
CCMS	Computing Center Management System
CCSID	Coded Character Set Identifier
CEC	Central Electronics Complex
CEE	Common Execution Environment
CF	Coupling Facility
CI	Correlation ID (WLM work qualifier), Central Instance, Control Interval
CICS	Customer Information Control System
CLI	Command Line Interface
CLOB	Character Large Object
CP	Control program
CPC	Central Processor Complex
CPU	Central Processing Unit
CRM	Customer Relationship Managment
CSM	Communications Storage Manager
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DASD	Direct Access Storage Device
DB	Database
DB2I	DB2 Interactive
DB2	Database 2 (an IBM relational database management system)
DBA	Database Administrator

	DBD	Data Base Descriptor
	DBIF	Database Interface (SAP component)
	DBMS	Database Management System
	DBRM	Data Base Request Module
	DBSL	Database Service Layer (functional interface within DBIF)
	DCE	Distributed Computing Environment
	DDF	Distributed Data Facility
	DFS	Distributed File Service
	DFSMS	Data Facility Storage Management Subsystem
	DIA	Dialog (SAP process type)
	DLL	Dynamic Linked Library
	DRDA	Distributed Relational Database Architecture
	DSN	Data Set Name
	DSP	Dispatcher (SAP process type)
	EBCDIC	Extended Binary Coded Decimal Interchange Code
	EDM	Environment Descriptor Modules
	EJB	Enterprise Java Beans
	ENQ	Enqueue (SAP process type)
	ERP	Enterprise Resource Planning
	ESCON	Enterprise Systems Connection
	FDDI	Fiber Distributed Data Interface
	FTP	File Transfer Protocol
	GB	Gigabytes
	GbE	Gigabit Ethernet
	GEN	Generic (SAP process type)
	GID	Group ID
	GMT	Greenwich Mean Time
	GRS	Global Resource Serialization
	GSA	General Services Administration (U.S.)
	GUI	Graphical User Interface
	GWY	Gateway (SAP process type)
	HFS	Hierarchical File System
	HLQ	High-Level Qualifier
	I/O	Input/Output
	IBM	International Business Machines Corporation
	ICF	Integrated Catalog Facility
	ICLI	Integrated Call Level Interface
	ICM	Internet Communications Manager (SAP process type)
	ICMP	Internet Control Message Protocol
	ID	Identifier
	IEEE	Institute of Electrical and Electronics Engineers (USA)
	IFCID	Instrumentation Facility Component ID
	IFI	Instrumentation Facility Interface
	IMS	Information Management System
	INET	Integrated Network
	IP	Internet Protocol
	IPL	Initial Program Load
	IPP	Internet Printing Protocol
	IRLM	Internal Resource Lock Manager
	ISO	International Standards Organization
	ISPF	Interactive System Productivity Facility
	ITSO	International Technical Support Organization
	J2EE	Java 2 Enterprise Edition
	JCL	Job Control Language
	JCS	Job Control Statement

JDK	Java Development Kit
JES	Job Entry Subsystem
JES2	Job Entry Subsystem 2
KB	Kilobytes (1024 bytes)
LAN	Local Area Network
LLA	Library Lookaside
LOB	Large Object
LOMS	Logical Output Management System
LPAR	Logical Partition
LPD	Line Printer Daemon
LUW	Logical Unit of Work
LX	Linkage Index
MB	Megabytes
MCOD	Multiple Components in One Database
MSG	Message server (SAP process type)
MTU	Maximum Transmission Unit
MVS	Multiple Virtual Storage (component of z/OS)
NFS	Network File System
NIC	Network Interface Card
NIS	Network Information System
NLS	National Language Support
NTT	Network Turnaround Time
OLAP	Online analytical processing
OLTP	Online transaction processing
OMS	Output management system
OMVS	OpenEdition MVS
OS/390	Operating System/390
OSA	Open Systems Architecture
OSAD	Open Systems Adapter Diagnostic Device
OSS	Online Service System
OTF	Output Text Format
PC	Process name (WLM work qualifier)
PCL	Printer Control Language
PDS	Partitioned Data Set
PFS	Physical File System
PID	Process ID
PK	Package Name
PR/SM	Processor Resource / Systems Manager
PSF	Print Services Facility
PTF	Program Temporary Fix
R/3	SAP R/3 System
RACF	Resource Access Control Facility
RC	Return Code
RDBMS	Relational Database Management System
REXX	Restructured Extended Executor Language
RFC	Remote Function Call
RFCOSCOL	RFC OS Collector
RISC	Reduced Instruction Set Computer
RMF	Resource Management Facility
RNL	Resource Name List
ROMS	Real Output Management System
RRS	Recoverable Resource Services, or Resource Recovery Services
RRSAF	Recoverable Resource Management Services Attachment Facility
RS/6000	IBM RISC System/6000
S/390	System/390

SAF	System (or Security) Authorization Facility
SAP	Systems, Applications, Products in Data Processing (software vendor)
SAPOSCOL	SAP OS Collector
SAPSID	SAP system ID
SCS	SAP central services
SDK	Software Development Kit
SLES	SUSE Linux Enterprise Server
SMB	Session Message Block
SMF	System Management Facility
SMS	Storage Management Subsystem
SNA	Systems Network Architecture
SPM	Subsystem Parameter (WLM work qualifier)
SPO	Spool (SAP process type)
SPUFI	SQL Processor Using File Input
SQL	Structured Query Language
STC	Started Tasks for Communication
SVC	Service
SWA	Scheduler Work Area
SYSADM	System Administration Authority
Sysplex	Systems Complex
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TFS	Temporary File System
TNG	Transaction name group
TP	Transport Tool (SAP)
TSO	Time Sharing Option
UACC	Universal Access Authority
UDB	Universal Database
UDP	User Datagram Protocol
UID	User ID
UNIX	An operating system developed at Bell Laboratories
UP2	Update (SAP process type)
UPD	Update (SAP process type)
USS	UNIX System Services
UTC	Universal Time Coordinated
VIPA	Virtual IP Address
VM	Virtual Machine
VMCF	Virtual Machine Communication Facility
VSAM	Virtual Storage Access Method
VTAM	Virtual Telecommunications Access Method
VTOC	Volume Table of Contents
VVDS	VSAM Volume Data Set
WLM	Workload Manager, Workload Management
WP	Work Process
WWW	World Wide Web
XCF	Cross-System Coupling Facility
XPLink	Extra Performance Linkage
zFS	z/OS File System
z/OS	zSeries Operating System
z/VM	zSeries Virtual Machine

Glossary

This glossary defines terms used in this publication.

abnormal end of task (abend). Termination of a task, a job, or a subsystem because of an error condition that cannot be resolved during execution by recovery facilities.

Advanced Interactive Executive (AIX). IBM's licensed version of the UNIX operating system. The RISC System/6000 system, among others, runs on the AIX operating system.

American National Standard Code for Information Interchange (ASCII). The code developed by ANSI for information interchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters.

application plan. The control structure produced during the bind process and used by DB2 to process SQL statements encountered during statement execution.

Asynchronous Transfer Mode (ATM). A transfer mode in which the information is organized into cells. It is asynchronous in the sense that the recurrence of cells containing information from an individual user is not necessarily periodic.

authorization ID. A string that can be verified for connection to DB2 and to which a set of privileges are allowed. It can represent an individual, an organizational group, or a function, but DB2 does not determine this representation.

Authorized Program Analysis Report (APAR). A report of a problem caused by a suspected defect in a current unaltered release of a program. The correction is called an APAR fix.

Authorized Program Facility (APF). A z/OS facility that permits identification of programs authorized to use restricted functions.

automatic bind. (more correctly, automatic rebind). A process by which SQL statements are bound automatically (without a user issuing a **BIND** command) when an application process begins execution and the bound application plan or package it requires is not valid.

bind. The process by which the output from the DB2 precompiler is converted to a usable control structure called a package or an application plan. During the process, access paths to the data are selected and some authorization checking is performed. See also 'automatic bind,' 'dynamic bind,' and 'static bind.'

client. In commercial, organizational, and technical terms, a self-contained unit in an SAP system with separate master records and its own set of tables.

compatibility mode.

1. (WLM) A mode of processing (no longer supported as of z/OS V1R3) in which the IEAIPSxx and IEAICSxx parmlib members determine system resource management. See also *goal mode*.

2. (DB2 V8) The first mode reached when migrating from DB2 V7. In this mode, new functions introduced in V8 cannot be used.

Cross-System Coupling Facility (XCF). The hardware element that provides high-speed caching, list processing, and locking functions in a Sysplex.

daemon. A task, process, or thread that intermittently awakens to perform some chores and then goes back to sleep.

data sharing. The ability of two or more DB2 subsystems to directly access and change a single set of data.

data sharing member. A DB2 subsystem assigned by XCF services to a data sharing group.

data sharing group. A collection of one or more DB2 subsystems that directly access and change the same data while maintaining data integrity.

database. A collection of tables, or a collection of tablespaces and index spaces.

Database Administrator (DBA). An individual responsible for the design, development, operation, safeguarding, maintenance, and use of a database.

database attach name. The name the ICLI server uses to attach to the DB2 subsystem.

database host. A machine on which the SAP database is stored and which contains the support necessary to access that database from an instance.

Database Interface (DBIF). In SAP, a program that resides on the application server and is responsible for interacting with the database.

Database Request Module (DBRM). A data set member created by the DB2 precompiler that contains information about SQL statements. DBRMs are used in the bind process.

database server. A term that is used for both database host and database service.

database service. A service that stores and retrieves business data in an SAP system.

DB2 Connect. The DB2 component providing client access to a remote database within the Distributed Relational Database Architecture (DRDA).

DB2 Interactive (DB2I). The DB2 facility that provides for the execution of SQL statements, DB2 (operator) commands, programmer commands, and utility invocation.

default. An alternative value, attribute, or option that is assumed when none has been specified.

Direct Access Storage Device (DASD). A device in which the access time is effectively independent of the location of the data.

Distributed Relational Database Architecture (DRDA). A connection protocol for distributed relational database processing that is used by IBM's relational database products. DRDA includes protocols for communication between an application and a remote relational database management system, and for communication between relational database management systems.

dynamic bind. A process by which SQL statements are bound as they are entered.

Enterprise Systems Connection Architecture (ESCON). An architecture for an I/O interface that provides an optical-fiber communication link between channels and control units.

Ethernet. A 10- or 100-megabit baseband local area network that allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD).

Extra Performance Linkage (XPLink). A high-performance linkage option in z/OS that enables performance improvements for a large class of call-intensive C and C++ applications. XPLink achieves this by changing register conventions and the layout of the stack, allowing for faster detection of stack overflow, faster saving of registers, faster allocation of local storage for called functions, opportunities for improved register allocation, and increased use of registers for function arguments and return values.

Fast Ethernet. Fast Ethernet is an Ethernet networking standard capable of data transmission rates as high as 100 Mbps. Fast Ethernet networking requires a network interface card (NIC) capable of transmitting data at 100 Mbps. Fast Ethernet can use copper twisted pair wires, coaxial cable, and optical fiber cable as its medium of transmission.

FDDI adapter. A hardware adapter attached to an AIX, Solaris, or Windows or a gateway system that connects the AIX, Solaris, or Windows or a gateway system via a fiber to an FDDI LAN.

fiber. The transmission medium for the serial I/O interface.

Fiber Distributed Data Interface (FDDI). A high-performance, general-purpose, multi-station network designed for efficient operation with a peak data transfer rate of 100 Mbps. It uses Token Ring architecture with optical fiber as the transmission medium over distances of several kilometers.

File Transfer Protocol (FTP). The Internet protocol (and program) used to transfer files between hosts. It is an application layer protocol in TCP/IP that uses TELNET and TCP protocols to transfer bulk-data files between machines or hosts.

Extended Binary Coded Decimal Interchange Code (EBCDIC). A set of 256 characters, each represented by 8 bits.

gateway. Intelligent interface that connects dissimilar networks by converting one protocol to another. For example, a gateway converts the protocol for a tokenring network to the protocol for SNA. The special computers responsible for converting the different protocols, transfer speeds, codes, and so on are also usually considered gateways.

goal mode. A mode of Workload Management processing where the active service policy determines system resource management. See also *compatibility mode*.

group name. The MVS XCF identifier for a data sharing group.

hexadecimal. (1) Pertaining to a selection, choice, or condition that has 16 possible different values or states. (2) Pertaining to a fixed-radix numeration system, with radix of 16. (3) Pertaining to a system of numbers to the base 16; hexadecimal digits range from 0 through 9 and A through F, where A represents 10 and F represents 15.

Hierarchical File System (HFS). A file system in which information is organized in a tree-like structure of directories. Each directory can contain files or other directories.

home address. Defines a single virtual IP address that is used by all RS/6000 systems to access a z/OS system, independent of the number of RS/6000 gateways connected to a given z/OS. This

implementation differs from the standard IP model that defines an IP address per physical adapter.

incremental bind. A process by which SQL statements are bound during the execution of an application process, because they could not be bound during the bind process and VALIDATE(RUN) was specified.

Initial Program Load (IPL). The process that loads the system programs from the auxiliary storage, checks the system hardware, and prepares the system for user operations.

instance. An administrative unit that groups together components of an SAP system that provide one or more services. These services are started and stopped at the same time. All components belonging to an instance are specified as parameters in a common instance profile. A central SAP system consists of a single instance that includes all the necessary SAP services.

Integrated Call Level Interface (ICLI). A component used by the SAP DBIF interface. It consists of client and server components and allows AIX, Solaris, or Windows application servers to access a z/OS or OS/390 database server remotely across a network. The DBIF uses only a subset of data base functions and the ICLI delivers exactly that subset.

Internal Resource Lock Manager (IRLM). A subsystem used by DB2 to control communication and database locking.

Internet. A worldwide network of TCP/IP-based networks.

job. Continuous chain of programs, controlled one after the other in time by particular control commands.

Job Control Language (JCL). A programming language used to code job control statements.

Job Control Statement (JCS). A statement in a job that is used in identifying the job or describing its requirements to the operating system.

Job Entry Subsystem (JES). In OS/VS2 MVS, a system facility for spooling, job queuing, and managing the scheduler work area.

link. The transmission medium for the serial I/O interface. A link is a point-to-point pair of conductors (optical fibers) that physically interconnects a control unit and a channel, a channel and a dynamic switch, a control unit and a dynamic switch, or, in some cases, a dynamic switch and another dynamic switch. The two conductors of a link provide a simultaneous two-way communication path. One conductor is for transmitting information and the other is for receiving information. A link is attached to a channel or control unit by means of the link interface of that channel or control unit and to a dynamic switch by means of a dynamic-switch port.

Local Area Network (LAN). A data network located on the user's premises in which serial transmission is used for direct data communication among data stations.

Logically Partitioned (LPAR) mode. A central processor complex (CPC) power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including central processors, central storage, expanded storage, and channel paths) among logical partitions. Contrast with basic mode.

Multiple Components in One Database. An SAP term that describes topologies in which more than one SAP system share one 'database'. In DB2 terminology, the SAP term 'database' is equivalent to a DB2 subsystem or a DB2 data sharing group. General information on MCOD is available at `http://service.sap.com/mcod`.

OSA-2 Feature. A hardware adapter attached to an OS/390 system, that connects the OS/390 system via a fiber to an FDDI LAN.

OSA-Express Feature. A hardware adapter attached to a zSeries or S/390 system that combines the functions of an I/O channel with the functions of a network port to provide direct connectivity between zSeries applications and their clients on the attached network.

PassTicket. An alternative to the RACF password that permits workstations and client machines to communicate with the host. It allows a user to gain access to the host system without sending the RACF password across the network.

password. In computer security, a string of characters known to the computer system and a user, who must specify it to gain full or limited access to a system and to the data stored within it. In RACF, the password is used to verify the identity of the user.

plan name. The name of an application plan.

proactive redirection. In DB2 data sharing topologies, the need can arise to redirect the work processes of an SAP application server to a different DB2 member of the data sharing group. Optimally, this operation should not be noticed by end users. Therefore, the SAP application server allows the SAP administrator to proactively redirect the work processes to a different DB2 member and thus avoid an error situation.

profile. Summary of system parameters with defined values. The parameters define, for example, the size of buffer areas, the maximum number of system users, and so on. The system parameters can be grouped together in a profile. When activating the system, a certain profile can be called up.

Program Temporary Fix (PTF).   A temporary solution or by-pass of a problem diagnosed by IBM System Support as the result of a defect in a current unaltered release of the program.

Reduced Instruction Set Computer (RISC).   A computer that uses a small, simplified set of frequently used instructions for rapid execution.

Relational Database Management System (RDBMS).   A relational database manager that operates consistently across supported IBM systems.

Resource Access Control Facility (RACF).   An IBM-licensed product that provides for access control by identifying and verifying users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

router.   An intelligent network component that holds information about the configuration of a network and controls data flows accordingly.

SAP system.   An SAP database and a collection of SAP instances (application servers) that provide services to the users. The collection of instances comprises one central instance and, optionally, one or more secondary instances. Each system has a system identifier, referred to as the SAPSID.

schema.   A logical grouping for user-defined functions, distinct types, triggers, and stored procedures. When an object of one of these types is created, it is assigned to one schema, which is determined by the name of the object. For example, the following statement creates a distinct type *T* in schema *C*:

```
CREATE DISTINCT TYPE C.T ...
```

SQL Processor Using File Input (SPUFI).   A facility of the TSO attachment subcomponent that enables the DB2I user to execute SQL statements without embedding them in an application program.

static bind.   A process by which SQL statements are bound after they have been precompiled. All static SQL statements are prepared for execution at the same time. Contrast with dynamic bind.

Storage Management Subsystem (SMS).   A component of MVS/DFP that is used to automate and centralize the management of storage by providing the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

Structured Query Language (SQL).   A standardized language for defining and manipulating data in a relational database.

subsystem.   A distinct instance of an RDBMS.

superuser.   In OpenEdition MVS, a system user who operates without restrictions. A superuser has the special rights and privileges needed to perform administrative tasks.

sysplex failover.   Sysplex failover support is the capability of SAP on DB2 to redirect application servers to a standby database server in case the primary database server becomes inaccessible.

System Authorization Facility (SAF).   A z/OS component that provides a central point of control for security decisions. It either processes requests directly or works with RACF or another security product to process them.

System Modification Program Extended (SMP/E).   A licensed program used to install software and software changes on z/OS systems.

Systems Complex (Sysplex).   The set of one or more OS/390 systems that is given a cross system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A Sysplex consists of one or more OS/390 systems.

Systems Network Architecture (SNA).   A widely used communications framework developed by IBM to define network functions and establish standards for enabling its different models of computers to exchange and process data. SNA is essentially a design philosophy that separates network communications into five layers.

table.   A named data object consisting of a specific number of columns and some number of unordered rows. Synonymous with base table or temporary table.

Time-Sharing Option (TSO).   An option of z/OS that provides conversational time-sharing from remote terminals.

Transmission Control Protocol/Internet Protocol (TCP/IP).   A software protocol developed for communications between computers.

Universal Access Authority (UACC).   The default access authority that applies to a resource if the user or group is not specifically permitted access to the resource. The Universal Access Authority can be any of the access authorities.

UNIX System Services.   The set of functions provided by the Shell and Utilities, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the z/OS and OS/390 operating systems that allow users to write and run application programs that conform to UNIX standards.

User Datagram Protocol (UDP).   A packet-level protocol built directly on the Internet protocol layer. UDP is used for application-to-application communication between host systems.

Virtual IP Address (VIPA). A generic term referring to an internet address on an host that is not associated with a physical adapter.

Virtual Machine (VM). A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system.

Virtual Storage Access Method (VSAM). (1) An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry sequence), or by relative-record number. (2) Term used for storing data on direct-access volumes.

Virtual Telecommunications Access Method (VTAM). A set of IBM programs that control communication between terminals and application programs running under DOS/VS, OS/VS1, and OS/VS2 MVS.

Workload Manager (WLM). The workload management services enable z/OS to cooperate with subsystem work managers to achieve installation-defined goals for work to distribute work across a sysplex, to manage servers and to provide meaningful feedback on how well workload management has achieved those goals. They also allow programs to create an interface to define a service definition. To change from resource-based performance management to goal-oriented workload management, many transaction managers, data managers, and performance monitors and reporters need to take advantage of the services z/OS workload management provides.

Work Process (WP). A job in the SAP system that actually does the work. Each work process is assigned a primary role by the dispatcher, which controls, to a certain degree, what type of work is to be performed by that work process. The number of work processes and the types that can exist for an instance are controlled by the instance profile and within the SAP system by the Central Computer Management System.

Bibliography

IBM documents

In the IBM Collection Kits listed in Table 30 you will find most of the documents that provide information related to the topics covered in this edition of the Planning Guide.

Table 31 through Table 33 on page 243 presents lists of IBM documents you will most likely have to consult in addition to this document. In most cases, the titles and form numbers given apply to the minimum release of the software required to run the SAP software described in this document (see also the release matrixes in Table 2 on page 7). Always use the latest edition of a manual that applies to the software release running on your system.

Table 34 on page 243 lists the IBM Redbooks that contain further interesting information.

In Table 35 on page 244 you will find the IBM order numbers and SAP material numbers for earlier editions of this document and for the current edition of the IBM manual *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition*.

Table 30. List of IBM Collection Kits

IBM Online Documents	Collection Kit Number
z/OS V1Rx Collection	SK3T-4269
z/OS Software Products Collection	SK3T-4270
IBM eServer zSeries Redbooks Collection	SK3T-7876
DB2 UDB Server for OS/390 and z/OS V7 Licensed Library Collection	LK3T-6999
DB2 UDB Server for OS/390 V6 Online and PDF Library Collection	SK3T-3518
Transaction Processing and Data Collection	SK2T-0730

Table 31. IBM DB2 documents

IBM DB2 documents	Order Number
DB2 for OS/390 and z/OS Administration Guide	SC26-9931
DB2 for OS/390 and z/OS Application Programming and SQL Guide	SC26-9933
DB2 for OS/390 Call Level Interface Guide and Reference	SC26-8959
DB2 for OS/390 and z/OS Command Reference	SC26-9934
DB2 for OS/390 and Z/OS Data Sharing Quick Reference	SX26-3846
DB2 for OS/390 and z/OS Data Sharing: Planning and Administration	SC26-9935
DB2 for OS/390 and z/OS Diagnosis Guide and Reference	LY37-3740
DB2 for OS/390 and z/OS Diagnostic Quick Reference	LY37-3741
DB2 for OS/390 and z/OS Installation Guide	GC26-9936
DB2 for OS/390 and z/OS Licensed Program Specifications	GC26-9938
DB2 for OS/390 and z/OS Messages and Codes	GC26-9940
DB2 for OS/390 and z/OS Reference for Remote DRDA Requesters and Servers	SC26-9942

Table 31. IBM DB2 documents (continued)

IBM DB2 documents	Order Number
DB2 for OS/390 and z/OS Reference Summary	SX26-3847
DB2 for OS/390 and z/OS Release Planning Guide	SC26-9943
DB2 for OS/390 and z/OS SQL Reference	SC26-9944
DB2 for OS/390 and z/OS Utility Guide and Reference	SC26-9945
DB2 for OS/390 and z/OS What's New?	GC26-9946

Table 32. IBM z/OS documents

IBM z/OS documents	Order Number
z/OS Communications Server: CSM Guide	SC31-8808
z/OS Communications Server: IP Configuration Guide	SC31-8775
z/OS Communications Server IP Configuration Reference	SC31-8776
z/OS Communications Server: IP Migration	GC31-8773
z/OS Communications Server: IP User's Guide and Commands	SC31-8780
z/OS DFSMS: Implementing System-Managed Storage	SC26-7407
z/OS DFSMS: Managing Catalogs	SC26-7409
z/OS DFSMSdss Storage Administration Guide	SC35-0423
z/OS DFSMSdss Storage Administration Reference	SC35-0424
z/OS Distributed File Service SMB Administration Guide and Reference	SC24–5918
z/OS Infoprint Server Customization	S544-5744
z/OS Infoprint Server User's Guide	S544-5746
z/OS Language Environment Customization	SA22-7564
z/OS Language Environment Debugging Guide and Run-Time Messages	GA22-7560
z/OS Language Environment Programming Guide	SA22-7561
z/OS Language Environment Programming Reference	SA22-7562
z/OS MVS Initialization and Tuning Reference	SA22-7592
z/OS MVS JCL Reference	SA22-7597
z/OS MVS Planning: APPC Management	SA22-7599
z/OS MVS Planning: Global Resource Serialization	SA22-7600
z/OS MVS Planning: Workload Management	SA22-7602
z/OS MVS Programming: Resource Recovery	SA22-7616
z/OS MVS Programming: Sysplex Services Guide	SA22-7617
z/OS MVS Programming: Sysplex Services Reference	SA22-7618
z/OS MVS Programming: Workload Management Services	SA22–7619
z/OS MVS Setting Up a Sysplex	SA22-7625
z/OS MVS System Commands	SA22-7627
z/OS Network File System Customization and Operation	SC26-7417
z/OS Network File System User's Guide	SC26-7419
z/OS Resource Measurement Facility (RMF) Performance Management Guide	SC33-7992
z/OS SecureWay Security Server RACF Security Administrator's Guide	SA22-7683
z/OS SecureWay Security Server External Security Interface (RACROUTE) Macro Reference	SA22-7692

Table 32. IBM z/OS documents (continued)

IBM z/OS documents	Order Number
z/OS UNIX System Services Messages and Codes	SA22-7807
z/OS UNIX System Services Command Reference	SA22-7802
z/OS UNIX System Services Planning	GA22-7800
z/OS UNIX System Services User's Guide	SA22-7801

Table 33. Other IBM documents

Other IBM documents	Order Number
Linux for zSeries and S/390 Device Drivers and Installation Commands (Linux kernel 2.4)	LNUX-1313
OSA-Express Customer's Guide and Reference (zSeries)	SA22-7935
OSA-Express Customer's Guide and Reference (S/390)	SA22-7403
Porting Applications to the OpenEdition MVS Platform	GG24-4473
PSF for OS/390: Introduction	G544-5625
PSF for OS/390: Customization	S544-5622
PSF for OS/390: User's Guide	S544-5630
SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition (see note [1])	SC33-7965-03
S/390 ESCON Channel PCI Adapter: User's Guide and Service Information.	SC23-4232
Using REXX and z/OS UNIX System Services	SA22-7806
High Availability for SAP on zSeries Using Autonomic Computing Technologies (see note [1])	SC33–8206
[1] Always refer to both the *Connectivity Guide* and *High Availability for SAP on zSeries Using Autonomic Computing Technologies* (SC33-8206). The latter contains the latest connectivity information.	

Table 34. IBM Redbooks and Redpapers covering related topics

Redbooks/Redpapers (published by the IBM International Technical Support Organization)	Order Number
SAP on DB2 for z/OS and OS/390: DB2 System Cloning	SG24-6287
Open Source Software for z/OS and OS/390 UNIX	SG24-5944
AFP Printing for SAP Using R/3 and R/2	SG24-4629
Implementing SAP R/3 in an OS/390 Environment Using AIX Application Servers	SG24-4945
SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations (see note [1])	SG24-5690
XPLink: OS/390 Extra Performance Linkage	SG24-5991
SAP on DB2 for OS/390 and z/OS: Multiple Components in One Database (MCOD)	SG24-6914
zSeries HiperSockets	SG24-6816
SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation (see note [1])	SG24-6836
SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing (see note [1])	SG24-6950
Distributed Functions of DB2 for z/OS and OS/390	SG24-6952
DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There	SG24-7088
mySAP Business Suite Managed by IBM Tivoli System Automation for Linux (Redpaper) (see note [1])	REDP-3717
[1] Information in these publications has been incorporated and updated in the IBM publication *High Availability for SAP on zSeries Using Autonomic Computing Technologies*, SC33–8206.	

Table 35. IBM and SAP numbers for the Connectivity Guide and for Planning Guide editions for earlier SAP releases

IBM Documents	IBM Order Number	SAP Material Number
SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide, 4th Edition	SC33-7965-03	51012938
SAP R/3 on DB2 for OS/390: Planning Guide; SAP R/3 Release 3.1I	SC33-7961-01	51005031
SAP R/3 on DB2 for OS/390 Planning Guide; SAP R/3 Release 4.0B Support Release 1	SC33-7962-03	51006445
SAP R/3 on DB2 for OS/390 Planning Guide; SAP R/3 Release 4.5B	SC33-7964-01	51005858
SAP R/3 on DB2 for OS/390 Planning Guide; SAP R/3 Release 4.6B	SC33-7966-01	51006841
SAP R/3 on DB2 for OS/390: Planning Guide; SAP R/3 Release 4.6D	SC33-7966-03	51009937
SAP on DB2 UDB for OS/390 and z/OS: Planning Guide; SAP Web Application Server 6.10	SC33-7959-00	51012939
SAP on DB2 UDB for OS/390 and z/OS: Planning Guide; SAP Web Application Server 6.20	SC33-7959-01	(Not assigned)

SAP documents

SAP documents can be ordered through SAPNet —Web Frontend (formerly: OSS) under: XX-SER-SWFL-SHIP.

Table 36. SAP documents

SAP Documents
BC SAP High Availability
(SAP online documentation is available in the SAP Library or at `http://service.sap.com/ha`)
SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide: SAP Web Application Server
SAP Web Application Server Heterogeneous System Copy
SAP Web Application Server Homogeneous System Copy
SAP Web Application Server Installation on UNIX: IBM DB2 UDB for OS/390 and z/OS
SAP Web Application Server Installation on Windows: IBM DB2 UDB for OS/390 and z/OS
SAP Software on UNIX: OS Dependencies
Planning Guide: z/OS Configuration for SAP on DB2 UDB for z/OS (starting with SAP Web Application Server 6.40)
SAP on IBM DB2 UDB for OS/390 and z/OS: Best Practice for Installing or Migrating to DB2 V8

SAP Notes

This section lists selected SAP Notes that are referenced in this publication.

Table 37. SAP Notes

SAP Note	Title
81737	DB2/390: APAR List
138906	DB2/390: Planning Info (Release, Connectivity Matrix)
160484	DB2/390: Database multi-connect with EXEC SQL
183311	Automated PTF Check
353529	DB2/390: Switching from daylight to standard time

Table 37. SAP Notes (continued)

SAP Note	Title
407325	Released operating systems SAP kernel 6.x DB2/390
660528	Requirements to run SAP NetWeaver on zSeries
728743	zSeries: Release of DB2 V8 for SAP Components
731937	DB2–z/OS: Customizing DB2 Connect for Web AS 6.20 and DB2 V8

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user. IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany
Attention: Information Request

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this publication to Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites. The materials at these Web sites are not part of the licensed materials for SAP on DB2 UDB for OS/390 and z/OS. Use of these materials is at your own risk.

Trademarks and service marks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

Advanced Function Presentation
Advanced Function Printing
AFP
AIX
CICS
Database 2
DB2

| DB2 Connect
| DB2 Universal Database
| DFS
| DFSMSdss
| Distributed Relational Database Architecture
| DRDA
| Enterprise Systems Connection Architecture
| ESCON
| eServer
| HiperSockets
| IBM
| IMS
| Infoprint
| IP PrintWay
| Language Environment
| Micro Channel
| MVS
| MVS/DFP
| OpenEdition
| OS/390
| Parallel Sysplex
| Print Services Facility
| PrintWay
| Processor Resource / Systems Manager
| PR/SM
| RACF
| RAMAC
| Redbooks
| RISC System/6000
| RMF
| RS/6000
| S/390
| SecureWay
| System/390
| Tivoli
| VTAM
| z/OS
| zSeries
| z/VM

| Linux is a trademark of Linus Torvalds in the United States, other countries, or
| both.

| Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation
| in the United States, other countries, or both.

UNIX is a registered trademark of the Open Group in the United States and other
countries.

Microsoft, Windows, Windows NT, Windows 2000, Windows 2003, and the
Windows logo are trademarks of Microsoft Corporation in the United States, other
countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the
United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

A

abbreviations   231
abnormal end of task (abend)   160
Address Space ID   45
Advance Function Printing (AFP)   141
American National Standard Code for
   Information Interchange (ASCII)   124
application name field   112
application servers
   changing the database server   115
   on AIX
      user ID <sapsid>adm   57
   on UNIX   8, 17, 23, 24, 26
   on UNIX or Linux   31
   on Windows   8, 17, 26, 31, 141
   on z/OS   17, 26, 29, 141
      ASCII mode   49
      DB2 authorization   134
      DB2 binding   137
      group ID (GID)   25, 26
      heterogeneous configuration   17
      hostname command   49
      Sysplex configuration   19
      TCP/IP setup   47
      user ID <sapsid>adm   23, 26
Asynchronous Transfer Mode (ATM)   9
authorizations
   superuser authorization   25
Authorized Program Analysis Report
   (APAR)
   fixes
      OW35419   45
      OW48709   45
Authorized Program Facility (APF)
   APF modules   125
Automatic Restart Management
   (ARM)   115, 159

B

bibliography
   IBM documents   241
   SAP documents   244

C

C-shell
   csh   63
   logon profiles   63
   tcsh   63
caches
   Catalog Data Space Cache   71
   Dynamic Statement Cache   122, 129,
      160
Catalog Address Space (CAS) and Global
   Resource Serialization (GRS) settings
   CATMAX   71
   Exclusion RNL   71
   Reserve Conversion RNL   71
   RESMIL   71

Catalog Address Space (CAS) and Global
   Resource Serialization (GRS) settings
   (continued)
   STRNO   71
   SYSVTOC   71
   SYSZYYDS   71
CD-ROMs   61
checklist guiding through this
   document   153
Code page conversion
   ICLI   98
Coded Character Set Identifier
   (CCSID)   124
Computing Center Management System
   (CCMS)   29
Computng Center Management System
   (CCMS)   77
configuration structure
   adding ICLI server instances   104
   components   3
   ICLI server instances   6
   OSA-2   10
   sample configuration structures   11
   SAP application servers   5
connectivity
   hardware   10
   recommendations   11
csh   63
Customer Information Control System
   (CICS) monitoring facility traces   72

D

daemons
   administration of   28
   defining security for   28
   Infoprint Server SAP Callback
      daemon   142, 143, 145
   LPDs (Line Printer Daemons)   144
      Infoprint Server LPD   142, 143,
         145, 146
      lpd   141
      saplpd   141
   setting up   29
data sets   29, 104
   DB2 data sets   25
   full data set director   35
   JCL data sets   28
   log stream data sets   34, 35
   logger couple data sets   34
   MVS data sets   49, 100, 104
   partitioned data sets   68
   Sysplex couple data sets   34
database attach name   167
database connection test   125
Database Interface (DBIF)   98, 129
Database Request Module (DBRM)   125,
   136
database servers
   changing   115
   failure of   6

database servers *(continued)*
   stopping   113
DB2
   proper code levels   71
   topologies   4
   WLM considerations   35
   z/OS setup requirements   33
DB2 authorizations   134
   for the ICLI server   134
DB2 binding
   for the ICLI server   135
   for the SAP application server on
      z/OS   137
DB2 buffer pools   18
DB2 Connect
   planning transition to   147
DB2 data sharing   4, 6, 14, 19
   groups   6
   members   115
DB2 group ID <SCHEMA>   24, 29, 125
   in MCOD environments   25
DB2 load library   135, 136
DB2 logs
   archiving media   33
   archiving space   33
   space requirements   33
DB2 privileges   24, 135
DB2 subsystem
   definition of   4
   name   111, 135, 136
DB2 system parameters
   with required settings
      CTHREAD   129
      IDBACK   129
DB2 threads
   identifying for SAP work
      processes   111
DB2 utilities   29
   RUNSTATS   29
diagnostic information to collect
   basic   166
   in case of communication
      problems   166
Direct Access Storage Device (DASD)   33

E

enclaves
   BATCH   87
   DIALOG   87
   ENQUEUE   86
   GENERIC   87
   SPOOL   87
   UNKNOWN   86
   UPDATE   87
Enhanced ASCII   50
environment variables
   _BPX_SHAREAS   47
   ICLI client   157
      ICLI_COM_TRACE   121
      ICLI_TRACE   120

251

U

Universal Access Authority (UACC) 28,
  72
UNIX sockets
  increasing 45
UNIX System Services parameters
  checking 44
    for the ICLI server 41
    for the SAP application server 41
user IDs
  creation of 23
  for FTP access 29
  with DB2 SYSADM authority 29, 135

V

VIPA
  dynamic 11, 101
virtual storage
  checking with memlimits 42
Virtual Storage Access Method
  (VSAM) 71
    ICF catalog size 33
VTOC size 33
VVDS size 33

W

Workload Management (WLM)
  checking setup 72
  classification of ICLI server
    workload 83
  classification of SAP application server
    workload on z/OS 87
  classification of SAP workload 82
  classification rules 88
  considerations for DB2 35
  minimum configuration for SAP
    workload 36
  preparing SAP application servers 37
  sample service definition 89
  scenarios 38
  service definition 84
  setup 36
  system settings 36

X

XPLink 40

Z

z/OS
  configuration 33
  configuring ICLI server 40
  configuring SAP application
    server 40
  Infoprint Server 142
  proper code levels 71
  setup for DB2 33
z/OS UNIX System Services
  group ID (GID) 24, 25, 26, 28
  process ID 103, 107, 109, 110, 117,
    119, 130

z/OS UNIX System Services *(continued)*
  shell scripts
    iclistart 66
  shells 49, 66, 70, 104, 105, 107, 109,
    110, 116
    C-shell (csh) 26, 63
    tcsh 63
  user ID (UID) 24, 25, 26, 28

Readers' Comments — We'd Like to Hear from You

z/OS
SAP on DB2 UDB for OS/390 and z/OS: Planning Guide
2nd Edition
SAP Web Application Server 6.20

Publication No. SC33-7959-02

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?   ☐ Yes   ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

_____   _____
Name   Address

_____
Company or Organization

_____
Phone No.

IBM ®

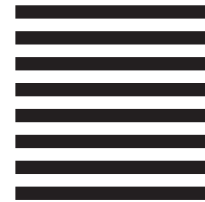Fold and Tape   **Please do not staple**   Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL   PERMIT NO. 40   ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schönaicher Strasse 220
D-71032 Böblingen
Federal Republic of Germany
 71032-0000

Fold and Tape   **Please do not staple**   Fold and Tape

IBM ®

Printed in USA

Spine information:

IBM

z/OS

SAP on DB2: Planning Guide 2nd Ed. (SAP Web AS 6.20)

SC33-7959-02