

**MAGRO**

**TCS**  
**R5**

**Client & Server**

**User Guide**

**TCS and TCS-Tools are products of  
MAGRO M&W Grossbauer, Wals-Salzburg / Austria**

UNIX is a registered trademark of AT&T.  
UNISYS is a registered trademark of Unisys Corporation.  
Micro Focus is a registered trademark of Micro Focus.  
ORACLE is a registered trademark of Oracle Corporation.  
INFORMIX is a registered trademark of Informix Software, Inc.

# TCS-R5 Client & Server

## Table of Contents

<b>1</b>	<b>PRODUCT DESCRIPTION.....</b>	<b>1-1</b>
1.1	ONE TCS-SERVER - MANY CLIENTS.....	1-1
1.2	EXTERNAL SERVER PROGRAMS.....	1-1
1.3	TCS-DIALOG-INTERFACE.....	1-1
1.4	TCS MANAGER, SERVER AND MESSAGE SERVER.....	1-2
1.5	gTCS - THE TCS GUI APPLICATION-FRAMEWORK.....	1-2
1.6	INTEGRATION OF WINDOWSPROGRAMS.....	1-3
1.7	TCSMON.....	1-3
1.8	SOFTWARE COMPONENTS.....	1-3
1.8.1	Screen conversion.....	1-3
<b>2</b>	<b>REQUIREMENTS.....</b>	<b>2-1</b>
<b>3</b>	<b>CLIENT/SERVER ARCHITECTURE.....</b>	<b>3-1</b>
3.1	SERVER CONFIGURATION.....	3-2
<b>4</b>	<b>TCS DIALOG INTERFACE.....</b>	<b>4-1</b>
4.1	DIALOG GENERAL .....	4-1
4.2	INITIAL DIALOG (PROLOG).....	4-2
4.3	LOGIN DIALOG.....	4-3
4.4	CLIENT DIALOG WITH SERVER.....	4-4
4.4.1	Direct-Dialog-Mode.....	4-5
4.4.2	IMS-Mode.....	4-5
4.5	DIALOG TERMINATION .....	4-6
4.6	SAMPLE DIALOG.....	4-7
4.6.1	Sample Initial Dialog (Prolog).....	4-7
4.6.2	Sample Direct-Dialog-Mode .....	4-7
4.6.3	Sample Session termination.....	4-7
<b>5</b>	<b>COMMON COMMANDS.....</b>	<b>5-1</b>
5.1	START AND STOP SERVER.....	5-1
5.2	TCSNETR5 COMMANDS.....	5-1
5.3	CLIENT COMMANDS.....	5-1
5.4	MANUAL SERVER SHUTDOWN.....	5-2
5.4.1	Clean Shutdown.....	5-2
5.4.2	Force Shutdown.....	5-2
5.4.3	Shutdown single tcsServ-Prozess:.....	5-2
<b>6</b>	<b>SAMPLE UCDINOUT SERVERPROGRAMS.....</b>	<b>6-1</b>



## 1 Product Description

The TCS-R5 Client-Server enables the existing TCS-Application to run over a local Network (LAN) or the Internet

**without changing programs**

The use of client-server technology opens numerous possibilities for improved and optimized utilisation and scaling of the existing IT-application.

It provides the integration of graphical clients (GUI-Client) based on the existing TCS-Application.

Further development and extensions of the existing TCS-Application can be done using new tools and programming-languages, accessing the TCS-R5 Server using the TCS-Dialog-Interface.

### 1.1 One TCS-Server - Many Clients

The TCS-Rel-5 Client-Server System is configurable and allows

**many clients to be managed by one TCS-Dialog-Server,**

thus reducing the necessary server licenses for MF-COBOL and RDBMS-Database.

This is useful e.g. for workstations with more than one TCS-session open.

These sessions can be handled by one TCS-Dialog-Server, thus reducing the required server licenses **without** decreasing the performance.

### 1.2 External Server Programs

The TCS-Dialog-Server architecture allows the integration of non-TCS-style Cobolprograms into the online dialog.

A TCS program may call a non-TCS-program, and at the end return back to the TCS program. This allows the seamless integration of new technology and new programs outside the TCS-IMS program-structure.

Non-Cobol programs can also be integrated, provided that they are called from a cobol-program and control is returning to the cobol program.

### 1.3 TCS-Dialog-Interface

The TCS-Dialog-Interface is the protokoll for communication between client and server, providing the interface between the TCS-Dialog-Server and the TCS-GUI-Clients. It describes the methods, structures and commands necessary to communicate with the TCS-Dialog-Server.

The definition of the TCS-Dialog-Interface enables you to use **client programs in any programming-language** (Tcl/Tk, Java, VisualBasic, Perl, etc.), which are able to communicate via TCP/IP.

## 1.4 TCS Manager, Server and MessageServer

On the server side there are three components providing the efficient execution of the application:

tcsMngr - the managerprocess, delivering the client to the appropriate TCS-Server.  
tcsServ - the TCS server, executes the programs and communicates with the client.  
tcsSMon - the message server, used to send messages to clients.

The TCS-R5 Server provides the full functionality as the previous version TCS-R4.

**Concurrent use:** The TCS-R5 Server can be used concurrently with TCS-R4.

**Application Email:** an interface to the server's e-mail system allows programs to send email.

**Automatic Email:** The TCS Server sends automatic messages to specified addresses on program errors. Server errors are mailed automatically to MAGRO to provide optimized support.

This feature is configurable, among many others.

Popup Messages: a program interface enables graphical communication with the client during the program-flow, e.g. to show the output of a script run by the TCS program.

ProgressBar: long running programs can send a progressbar.

Other features can be implemented, which is a part of the service MAGRO provides.

## 1.5 gTCS - The TCS GUI Application-Framework

The TCS GUI Application-Framework, called **gTCS**, is the backbone of the GUI client frontend. This interface provides the user an interactive interface to the system. A major advantage of a GUI is the ease, intuitive and ergonomical use of applications.

gTCS consists of a **Server Selection Dialog**, a **Login Dialog** and the **gTCS Window** with all elements required to operate the TCS application.

The TCS ClientFramework is fully configurable, e.g. color of terminal background, fontsize, fontcolor, block cursor, etc.

Textcomponents (buttontext, labeltext, menutext, etc.) are held in a textfile (tcsMSG.txt), providing the possibility to translate and/or change the existing wording of the gTCS Application Framework.

By default all the text is in English.

Currently a German textset is available.

**Various graphical elements** for the screens are available:

ComboBox, Tooltip, Scrolling Text, Buttons, Checkbox, Frames and Titleframes, Pictures and other graphical elements (e.g. gif, png, jpeg), Colors, and so on.

**Dynamic ComboBox:** this allows you to retrieve data from the server and fill the combobox. This requires **no** change of the the program.

**Dynamic ToolTip:** data are retrieved from the server and are shown as ToolTip help. Useful on screens with many fields to supply additional information on demand.

The TCS-GUI-Framework (Tcl/Tk) runs on MS-Windows (98, 98SE, NT, 2000, XP), on X-Window (Linux/Unix) and on McIntosh.

## 1.6 Integration Of Windowsprograms

Using the gTCS Application Framework, a seamless integration of other Windows programs like Adobe Reader, webbrowser, email programs etc. is provided.

This allows the integration of existing GUI-applications (e.g. OracleForms, Java-applications) into the gTCS Application Framework.

## 1.7 tcsMon

The TCS Monitorprogram tcsMon, running on MS-Windows, X-Window and MAC-OS, provides a tool to observe the TCS servers and send messages to the clients.

## 1.8 Software Components

By using Open Source standard software components Tcl/Tk, the software costs for the GUI-Frontend are minial or there are even no costs at all. By using the TCS-GUI, the **terminal-emulation** on each workstation is **obsolete**.

### 1.8.1 Screen Conversion

Several conversion-tools are available to convert SFS-Screens into the appropriate GUI-component. Non SFS-screens can also be implemented, but will require some re-engineering or implementaion of an appropriate interface.



## TCS-R5 Client & Server

### 2 Requirements

TCS-R5 Client Server runs concurrently with TCS-R4, thus a valid installation of TCS-R4 is required.

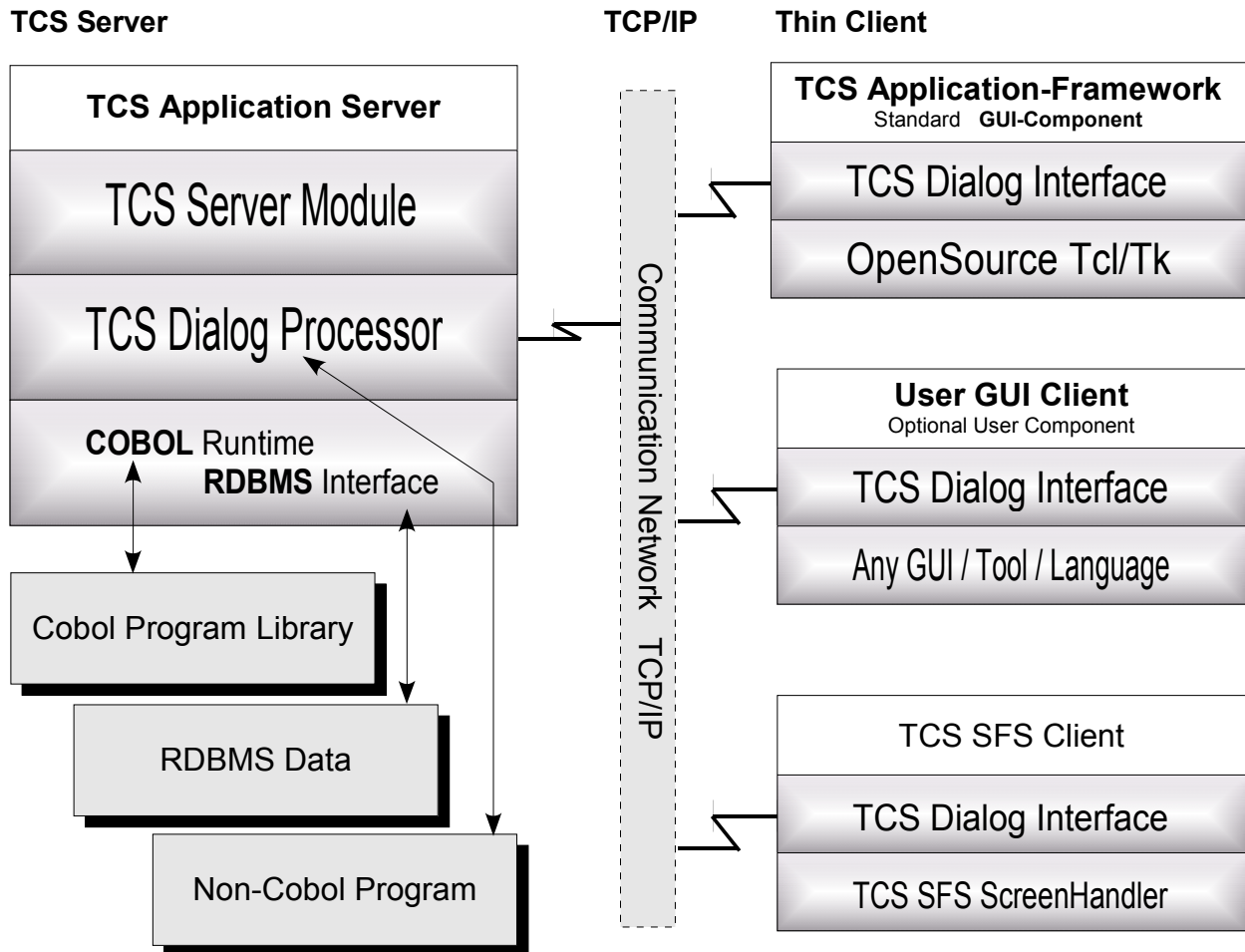
The TCS-R5 servers are sharing the TCS Shared-Memory (known as TCSDDI) with TCS-R4. Furthermore, all the requirements for TCS-R4 also apply for the TCS-R5 Client/Server software.



### 3 Client/Server Architecture

The TCS-R5 Client/Server is designed according to praxis-proven, stable programming techniques.

The base for the design is the server model **'Preforked Server, descriptor passing'**.



### 3.1 Server Configuration

The file 'TCS.cfg', residing in the tcsNetR5-homedirectory, contains all the necessary configuration paramaters:

[Manager]		Manager section
Port	= 5090	Service port
MaxServ	= 128	maximum number of servers
CCOMEXEC	= 1	start CCOMEXEC MENU-Servers (0-n)
[Server name]		<b>Server section</b> + optional servicename Multiple server-sections with servicename can describe different TCS services. If no servicename is specified, this service is referred to as <b>&lt;default&gt;</b> service.
Servers	= 6	number of server to load initially
Autoload	= yes	load server if none available (yes no)
MaxLogSize	= 10	rotate logfile at size (in MB)
Timeout	= 0	terminate after n sec w/o client reaction
PTimeout	= 10	terminate after n sec if no COBOL return
SIC	= yes	enable ScreenInfoCache (yes no)
[Client]		Client section
Clients	= 3	number of clients per server (max=16)
Group	= Off	grouping: Off   IP   GID   Name
	= IP	grouping client by IP-Number
	= GID	grouping by ClientGroup (gTCS -group n)
	= Name	grouping by PC-Name ( <a href="#">login@computer</a> )
[Environment]		<b>Environment section.</b> Sub-section of server-section. By default, TCS-Server uses the environmet variables set in .profile of tcsNetR5. Individual environment variables can be set in this section.
TCSNET_HOME	= autoset	set by tcsMngr
TCS_SMI	= \$TCS_SMI	Production DB's TCSDDI
TCS_LOCK	= path	directory TCS-lockfile

## TCS-R5 Client & Server

TCS\_USER = filename Path and filename of USER.TAB  
TCS\_APP = path application directory (was OS3)  
TCS\_TRM = name 4 char name if no user entry in USER.TAB  
TCS\_DIR = path[:path:path] TCS lodaklibarries  
TCS\_USRLOG = path[:path:path] /username: directory for err-files  
TCS\_TRACE = C Y|N|C; ending with 1 = full trace for SEND

# --- Automatic EMail: P=email-address receiving program errors

# --- optional :S=email-address receiving system errors

# --- optional :N=no mail on this errors

TCS\_MAIL = tcs@magro-soft.com:S=wg@magro-soft.com:N=601,602

TCS\_GUI = path GUI's to upload to client

ORACLE\_SID = SID

USERID = user[/password]

INFORMIXSERVER = servername

TERM = xterm-color required for animator

COBTERMINFO = pathname required for animator

COBSW = +10 **\*must\* enable real cobcancel**

COBPATH = path[:path:...]

COBANIMPATH = path[:path:...]

COBCPY = path[:path:...]

# --- environment for TCSDMS

TCS9\_DBK = 4

TCS9\_ORAFIL = 1



## 4 TCS Dialog Interface

This is the protocol used by Clients to access TCS-COBOL-Programs, handled by TCS-R5-Server.

The Client opens the TCS-R5 Server Port, default is 5090.

### 4.1 Dialog General

**Every message** must begin with a 2-byte length field, format: *short network byte order*.

In this document this is indicated with the notation **LL**.

To convert to network byte order (send message), the function `htons` can be used.

To convert from network byte order (recv message), the function `ntohs` can be used.

These functions are part of all sockets API (e.g. `wsock32.dll` or `<netinet/in.h>`)

Legend:

**SCB** Screen Control Block

**CMA** Client Message Area (IMS IMA)

**SMA** Server Message Area (IMS OMA)

**LL** 2 bytes (short) text length in network byte order

☒ control character; hexadecimal 0x18 or octal 030 or decimal 24

## 4.2 Initial Dialog (Prolog)

establish TCP/IP communication with TCS-R5 Server

- Send message:

**LL**☒@TCS:Service V=3 G= ID=x@y TC=T1

Format:   **LL**                   2 bytes (short) text length in network byte order  
           ☒                     hexadecimal value 18 preceding the string @TCS  
           TCS                   Service  
           :Service              optional; name of service as defined in TCS.cfg.  
                                   If omitted the **default** service is connected.  
           V=3                   version number  
           G=                    optional group information  
           ID=x@y                PC username and computername  
           TC=T1                optional initial transaction code

- Read message:

**LL**☒@OK tcsMngr V3

Check the control sequence ☒@OK

- Read message:

**LL**☒@ID PID ProcessName IP:Port 2004-11-18 10:15:56 SS n Service

Format:   **LL**                   2 bytes (short) text length in network byte order  
           ☒@ID                 message prefix  
           PID                   process id of tcsServ  
           ProcessName         full process name of TCS Serverprocess  
           IP:Port              Server IP- and Portnumber  
           date  
           time  
           SS                    Session String: unique string to identify/verify session  
           n                     TCS\_SMI: TCS shared memory identifier  
           Service              optional name of service.

### 4.3 Login Dialog

- Read message:

**LL**☒@Log:

This introduces the server login dialog.

- Send message:

**LL**serveruserid☒serverpassword☒no

userid and password must end with hexadecimal 0x18.

**Note:** Password transmission within the *local* Network is unencrypted, the implementation of a higher security level is scheduled.

- Read message:

**LL**☒@OK:TCS:n(\*) TRM1 -1 path

Check the control sequence ☒@OK

Format:	<b>LL</b>	2 bytes (short) text length in network byte order
	☒@OK:	message prefix
	TRM1	TCS_TRM
	:n(*)	menufile-mtime(transaction-code) ex. USER.TAB
	-1	session limit (used by gTCS Client)
	path	client's homedirectory on server

In case of error the message contains

**LL**☒@:Invalid Username/Password serverid (1:3)

In this case disconnect and restart from begin.

- Send message:

**LL**☒@InitGUI None|All

'All' get and set UserMenu.

Confirms to the TCS-R5-Server that the GUI Client is ready.

'None' indicates no local GUI-component update, this is used by the *gTCS* client only.

- Read message:

**LL**TCS-INIT

This message has a length of 60 bytes. The first 8 characters are fix, the remainder used by the *gTCS* client only.

- Read next message

**LL**tty1.....SID/UID

This is used by the *gTCS* client only.

After this, the **Initial Dialog** (Prolog) between the Client and TCS-R5-Server is complete.

## 4.4 Client Dialog with Server

A message always consist of 2 parts:

sending SCB and CMA      **ScreenControlBlock** and **ClientMessageArea**  
 receiving SCB and SMA      **ScreenControlBlock** and **ServerMessageArea**

Legend:

UCdInOut      **UserClientdirectInputOutput**  
**SCB**          **Screen Control Block**  
**CMA**          **Client Message Area**                      (IMS IMA)  
**SMA**          **Server Message Area**                      (IMS OMA)

**LL**              2 bytes (short) text length in network byte order  
 ☒                control character; hexadecimal 0x18 or octal 030 or dezimal 24

- Send message part-1 (SCB):

**LL**UCdInOut..transaction-code                      DirectDialogMode

or

**LL**UCdInOut..☒@IMS                                      IMS mode

This is a 60 character message with trailing spaces.

- Send Message part-2 (CMA):

**LL**Message-for-the-Server-COBOL-program  
 up to 32000 characters message for the TCS-COBOL-program.  
 This data is available in the InputMessageArea (IMA) of the TCS COBOL program.

- Read Message part-1 (SCB):

SCB-m2get, Byte 12 (11 bytes offset of SCB) is set to "1" if one more message is waiting.  
 This is an error message in most cases.

- Read Message part-2 (SMA):

Up to 32000 characters built in the OutputMessageArea (OMA) of the TCS COBOL program  
 executed on TCS-R5-Server.

- ▶ The message from the server program begins after a **16 byte header**.

SCB-m2get, Byte 12 (11 bytes offset of SCB) is set to "1":

- Read Message part-1 (SCB).
- Read Message part-2 (SMA):
- ▶ This message has a **10 byte header** (ascii escape seq.) and may be ignored.

## TCS-R5 Client & Server

### 4.4.1 Direct-Dialog-Mode

<i>Client</i>	<i>TCSTCD.TAB</i>	<i>Server</i>
send SCB 'LLUcdInOut.. <i>code1</i> ' (transaction-code) send CMA 'LLmessage for program1'  receive SCB receive SMA 'LLmessage from program1'	<i>code1</i> =program1	Start program1 Process message from client.  Send message to client.
Transactioncode up to 16 characters.		

### 4.4.2 IMS-Mode

	<i>Client</i>	<i>TCSTCD.TAB</i>	<i>Server</i>
1) First time	send SCB 'LLUcdInOut.. <input type="checkbox"/> @IMS' send CMA 'LL <i>code1</i> ' (transaction-code)	<i>code1</i> =program1	Start program1
2) Next times	send SCB 'LLUcdInOut.. <input type="checkbox"/> @IMS' send CMA 'LLmessage for program1'  receive SCB receive SMA 'LLmessage from program1'  This is repeated at step 2) until the the server-COBOL-program sets the TERMINATION-INDICATOR to 'N' . This is usually controlled by the client by sending a request to terminate with 'N' .		process message from client.  send message for client.
A server COBOL program terminating (CALL 'RETURN'; EXIT PROGRAM.) with TERMINATION-INDICATOR set to 'N' tells the server to expect a transaction-code the next time. The communication flow will start again at step 1)			
Transactioncode up to 5 characters.			

## 4.5 Dialog Termination

- Send message part-1:

**LL**UCdInOut

- Send message part-2:

**LL**☒@stop

This terminates the Client-Server communication in a clean way.

## 4.6 Sample Dialog

### 4.6.1 Sample Initial Dialog (Prolog)

```

send  LL☒@TCS V=3 G= ID=wg@magro1.magro TC=
recv  LL☒@OK tcsMngr V3
recv  LL☒@ID 26191 tcsServ-0/0 192.168.0.1:2956 2005-02-11 16:24:19 40B3B370 9
recv  LL☒@Log:
send  LLtcsNetR5☒passwd1234☒no
recv  LL☒@OK:TCS:0() trm1 -1 /opt/tcsNetR5
send  LL☒@InitGUI None
recv  LLTCS-INIT..I0.....tcsServ .....
recv  LLtrm1.....O80:TEST_DB

```

Prolog complete

### 4.6.2 Sample Direct-Dialog-Mode

```

send  LLUcdInOut..test1
send  LLMessage for COBOL-Program1
recv  LLUcdInOut.1B0..0.....PROG001..1974-04-10 09:34:08.
recv  LLMessage from COBOL-Program1

send  LLUcdInOut..test-programm-n2
send  LLMessage for Program2
recv  LLUcdInOut.1B0..0.....PROG002..1974-04-10 09:34:08.
recv  LLMessage from cobol-program2

```

Serverprogram dialog finished

### 4.6.3 Sample Session Termination

```

send  LLUCdInOut
send  LL☒@Stop

```



## TCS-R5 Client & Server

### 5 Common Commands

#### 5.1 Start And Stop Server

This commands are user to control TCS-R4 and TCS-R5.

Start TCS-R4 with TCS-R5 Client/Server	tcs_start.sh
Shutdown TCS-R4 and TCS-R5 Client/Server	tcs_shut.sh
Stop TCS-R4 and TCS-R5 Client/Server	tcs_stop.sh
Activate new Screenformat	ClearCache.sh ScreenName

#### 5.2 tcsNetR5 Commands

This commands must be run with tcsNetR5 UserID (login)

Start TCS-R5 Client/Server	startMngr.sh [-tn] -tn Tracemode; n=1-9
Shutdown TCS-R5 ClientServer	stopMngr.sh

#### 5.3 Client Commands

This commands can be run from the X-Window or the MS-Windows desktop.

TCS Graphic Client	gTCS
TCS Client and Server Monitor	tcsMon
TCS ScreenTest Programm	gSFT
TCS Screen Viewer	gSFV
TCS Screen Generator	GSFG

## 5.4 Manual Server Shutdown

You must be login as tcsNetR5. Otherwise no permissions are granted to shutdown TCS server.

Note: on non-Linux, the appropriate shell-script for `skill` is in `/usr/bin`

### 5.4.1 Clean Shutdown

```
skill -TERM tcsMngr  
skill -TERM tcsSMon
```

check if all processes have terminated:  
`ps -e | egrep "tcsMngr|tcsSMon|tcsServ"`  
This must return without any hits.

### 5.4.2 Force Shutdown

```
skill -KILL tcsServ  
skill -KILL tcsMngr  
skill -KILL tcsSMon
```

### 5.4.3 Shutdown Single Server:

```
kill -QUIT tcsServ-Processnumber
```

## 6 Sample UCdInOut Serverprograms

DISPLAY-output will be output to tcsServ-nn.log

```

*****
* Sample Serverprogram for non gTCS clients
*****
$SET CALLFH(EXTFH)
IDENTIFICATION DIVISION.
PROGRAM-ID. TEST001.
AUTHOR. MAGRO/WG.
DATE-WRITTEN. 2004-12-30.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. Linux-UNIX.
OBJECT-COMPUTER. Linux-UNIX.
SPECIAL-NAMES.
    DECIMAL-POINT IS COMMA.
*****
DATA DIVISION.
WORKING-STORAGE SECTION.
01  UCDINOUT                PIC X(8) VALUE "UCdInOut".
*****

**** session's protected and private areas
LINKAGE SECTION.
01  PROGRAM-CONTROL.        COPY PIB74 IN COPY.

**** Fields received from client
01  INP-AREA.                COPY IMA74 IN COPY.
02  INP-FIELDS.
03  INP-F001                PIC X(100).

**** WORK-FIELDS: for use between entry and exit program
01  WORK-FIELDS.
**** Fields to be sent to client
02  OUT-FIELDS.
03  OUT-F001                PIC X(60).
03  OUT-NL1                 PIC X(1).
03  OUT-F002                PIC X(60).
03  OUT-NL2                 PIC X(1).
03  OUT-F003                PIC X(60).
03  OUT-NL3                 PIC X(1).
03  OUT-LONG                PIC X(31000).

**** Area for client message transmission
01  OUT-AREA.                COPY OMA74 IN COPY.

**** SESSION-FIELDS: for use from begin until end of session
01  SESSION-FIELDS.
02  FILLER                  PIC X(60).
*****

```

## TCS-R5 Client & Server

```
*****
****      Note: do no change the order of the parameters
PROCEDURE DIVISION USING PROGRAM-CONTROL INP-AREA
                        WORK-FIELDS   OUT-AREA
                        SESSION-FIELDS.

PROC-DIV SECTION.
ANF01.
**** process message received from client
      MOVE "WORK-FIELDS some value..." TO WORK-FIELDS.
      MOVE "SESSION-FIELDS some value..." TO SESSION-FIELDS.

**** prepare message to be sent to client
      MOVE "Message from TEST001.CBL" TO OUT-F001.
      MOVE LENGTH OF OUT-FIELDS TO TEXT-LENGTH OF OUT-AREA.

      CALL 'BUILD' USING OUT-AREA UCDINOUT OUT-FIELDS.

**** DISPLAY can be used to debug the program.
**** It will be output into tcsServ-n.log file
END01.
**** send message to client
      CALL "RETURN" EXIT PROGRAM.
*****
****                      E N D                      ****
*****
```

## TCS-R5 Client & Server

### Index

<b>A</b>	Application EMail.....	1-2	<b>O</b>	One TCS-Server - many Clients.....	1-1
	Application-Framework.....	1-2	<b>P</b>	Port.....	3-2
	Autoload.....	3-2		Product Description.....	1-1
	Automatic EMail.....	1-2		Prolog.....	4-2
	Automatic EMail.....	3-3		PTimeout.....	3-2
<b>C</b>			<b>R</b>		
	CCOMEEXEC.....	3-2		Requirements.....	2-1
	Clean Shutdown.....	5-2	<b>S</b>		
	Client.....	3-1		Sample Dialog.....	4-7
	Client Commands.....	5-1		Sample Direct-Dialog-Mode.....	4-7
	Client Dialog.....	4-4		Sample Initial Dialog.....	4-7
	Client section.....	3-2		Sample Session termination.....	4-7
	Client/Server Architecture.....	3-1		Sample UCdInOut Serverprograms.....	6-1
	Clients.....	3-2		SCB.....	4-1
	CMA.....	4-1		Screen conversion.....	1-3
	Common Commands.....	5-1		Server Configuration.....	3-2
	Concurrent use.....	1-2		Server name.....	3-2
<b>D</b>				Server section.....	3-2
	default.....	3-2		Servers.....	3-2
	default.....	3-2		service.....	3-2
	Dialog General.....	4-1		service.....	3-2
	Dialog Termination.....	4-6		Shutdown single Server.....	5-2
	Direct-Dialog.....	4-5		SIC.....	3-2
	Direct-Dialog-Mode.....	4-5		SMA.....	4-1
<b>E</b>				Software Components.....	1-3
	Environment section.....	3-2		Start and Stop Server.....	5-1
	External Server Programs.....	1-1	<b>T</b>		
<b>F</b>				TCP/IP.....	3-1
	Force Shutdown.....	5-2		TCS Dialog Interface.....	4-1
<b>G</b>				TCS Manager.....	1-2
	Group.....	3-2		TCS Server.....	3-1
	gTCS.....	1-2		TCS_GUI.....	3-3
	gTCS.....	1-2		TCS_MAIL.....	3-3
<b>I</b>				TCS-Dialog-Interface.....	1-1
	IMS-Mode.....	4-5		tcsMgr.....	1-2
	Initial Dialog.....	4-2		tcsMon.....	1-3
	Integration of Windowsprograms.....	1-3		tcsNetR5 Commands.....	5-1
<b>L</b>				tcsServ.....	1-2
	LL.....	4-1		tcsSMon.....	1-2
	Login Dialog.....	4-3		Thin Client.....	3-1
<b>M</b>				Timeout.....	3-2
	Manager.....	3-2	<b>U</b>		
	Manual Server Shutdown.....	5-2		UCdInOut.....	4-4
	MaxLogSize.....	3-2	<b>X</b>		
	MaxServ.....	3-2		X-Window.....	1-3
	MessageServer.....	1-2			